



AeroBest 2023

II ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems

Proceedings

*19–21 July 2023
Lisboa, Portugal*

André C. Marta & Afzal Suleman (editors)

Published by:

IDMEC

Instituto Superior Técnico

Universidade de Lisboa

Portugal

<https://aerobest2023.idmec.tecnico.ulisboa.pt/>

ISBN: 978-989-53599-4-3

Editors:

André C. Marta

Afzal Suleman

Preface

This Book of Proceedings contains the full manuscripts on the works presented at AeroBest 2023, the second edition of the ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems. This document is a testimony of the latest developments in the field, covering a wide range of MDO computational methods, tools and processes and applications spanning civil aviation and space systems.


Following the successful debut in 2021, which was held online due to the COVID-19 pandemic, this second edition gathered in Lisbon almost 60 participants, who assisted to 38 presentations, including four keynote lectures by prominent speakers from academia and industry, who shared the state-of-the-art on the computational methods, tools and processes and experimental facilities in the field of study, and outlined the technological development roadmap for the future of sustainable aviation and space technology.

The participants from academia, government laboratories and industry highlighted the growing awareness and acceptance that 21st-century aerospace design problems must simultaneously take into account multiple physical disciplines using a seamless coupling and a friendly user-interface, while mitigating the inevitable demanding computational resources.

Being a thematic conference, relatively small but highly focused and specialized, resulted in a dynamic environment that fostered interesting and productive exchanges of ideas and discussions. A final word of appreciation to all participants, in particular to those whose work is documented in these proceedings, that made the conference possible.



André C. Marta
(Conference Chair)



Afzal Suleman
(Conference Chair)

Scientific Committee

Juan Alonso, Stanford University, USA

Pedro Camanho, Universidade do Porto, Portugal

Giuseppe Catalanotti, Universidade de Évora, Portugal

Jonathan Cooper, University of Bristol, United Kingdom

Pedro V. Gamboa, Universidade da Beira Interior, Portugal

Graeme Kennedy, Georgia Tech, USA

Alicia Kim, University of California San Diego, USA

André C. Marta, Instituto Superior Técnico, Portugal

Joaquim R. R. A. Martins, University of Michigan, USA

Joseph Morlier, ISAE-SUPAERO / ICA, France

Melike Nikbay, Istanbul Technical University, Turkey

Jacques F. Périaux, CIMNE / UPC Barcelona, Spain

Sergio Ricci, Politecnico di Milano, Italy

Hélder C. Rodrigues, Instituto Superior Técnico, Portugal

Carlos A. M. Soares, Instituto Superior Técnico, Portugal

Craig Steeves, University of Toronto, Canada

Afzal Suleman, University of Victoria, Canada

Sponsors



Mechanical Engineering Institute (*Instituto de Engenharia Mecânica*)



Portuguese Society for Theoretical, Applied and Computational Mechanics
(*Associação Portuguesa de Mecânica Teórica, Aplicada e Computacional*)



European Community on Computational Methods in Applied Sciences

Secretariat



IDMEC

Instituto Superior Técnico

Av. Rovisco Pais, 1

1049-001 Lisboa, Portugal



(+351) 218 417 351



aerobest2023@idmec.tecnico.ulisboa.pt



<https://aerobest2023.idmec.tecnico.ulisboa.pt/>

Keynote Lecturers



Charbel Farhat is the Vivian Church Hoff Professor of Aircraft Structures in the School of Engineering and the inaugural James and Anna Marie Spilker Chair of the Department of Aeronautics and Astronautics. He is also Professor of Mechanical Engineering, Professor in the Institute for Computational and Mathematical Engineering, and Director of the Stanford-King Abdulaziz City of Science and Technology Center of Excellence for Aeronautics and Astronautics. He currently serves on the Space Technology Industry-Government-University Roundtable. From 2007 to 2018, he served as the Director of the Army High Performance Computing Research Center at Stanford University; and from 2015 to 2019, he served on the US Air Force Scientific Advisory Board. He is a Member of the National Academy of Engineering, a Member of the Royal Academy of Engineering (UK), a Member of the Lebanese Academy of Sciences, a recipient of a Docteur Honoris Causa from Ecole Normale Supérieure Paris-Saclay, a recipient of a Docteur Honoris Causa from Ecole Centrale de Nantes, a recipient of a Docteur Honoris Causa from Ecole Nationale Supérieure d'Arts et Métiers, a designated ISI Highly Cited Author in Engineering by the Institute for Science Information (ISI) Web of Knowledge, and a Fellow of six professional societies: SIAM, ASME, IACM, WIF, US-ACM and AIAA. He was knighted by the Prime Minister of France in the Order of Academic Palms and awarded the Medal of Chevalier dans l'Ordre des Palmes Académiques. He is the recipient of several other professional and academic distinctions. Professor Farhat and his research group at Stanford University develop mathematical models, advanced computational algorithms, and high-performance software for the design and analysis of complex systems in aerospace, automotive, marine, mechanical, and naval engineering. They contribute major advances to Simulation-Based Engineering Science. Current engineering foci in research are on the nonlinear aeroelasticity and flight dynamics of Micro Aerial Vehicles (MAVs) with flexible flapping wings and N+3 aircraft with high aspect ratio wings, layout optimization and additive manufacturing of wing structures, supersonic inflatable aerodynamic decelerators for Mars landing, and model predictive control. Current theoretical and computational emphases in research are on high-performance, multi-scale modeling for the high-fidelity analysis of multi-physics problems, high-order embedded boundary methods, uncertainty quantification, probabilistic machine learning, efficient model-order reduction for time-critical applications such as design and active control, and digital twins. Professor Farhat is Editor-in-Chief of the International Journal for Numerical Methods in Engineering and Editor of the International Journal for Numerical Methods in Fluids. He also serves on the editorial boards of ten other international scientific journals. He has been an AGARD lecturer on aeroelasticity and computational mechanics at several distinguished European institutions, and a plenary speaker at numerous international scientific meetings. He is the author of over 650 refereed publications.



António graduated with a Master's degree in Aerospace Engineering from Lisbon Technical University (IST) and Aerospace Vehicle Design from Cranfield University. He joined Airbus in 1998 as an Aerodynamics Engineer at Airbus UK in Bristol, working on the preliminary wing design for the A380 and A400M programmes. He then moved to the Marketing department, based in Toulouse, France, working on sales campaigns to airlines in Asia, Europe, Latin America and Middle East. In 2005 he was appointed Marketing Director for the Middle East sales unit, managing all the marketing activities for this region. Subsequently, he joined the Strategy department working on future aircraft developments, bringing several developments to light such as the A320neo. Currently he is the Vice-President of Single Aisle Marketing, responsible for marketing the A220 and A320 aircraft Families.



Sergio Ricci is a Full Professor at the Department of Aerospace Science and Technology, Politecnico di Milano. He has been Principal Investigator for the more than 12 European funded research projects (EU-FP7, EU-FP6, EU-FP5) of Dipartimento di Ingegneria Aerospaziale, with emphasis to projects GLAMOUR – Gust Load Alleviation techniques assessment on wind tunnel Model of advanced Regional aircraft (2014-2016), NOVEMOR – NOVel AIR VEHICLE configurations: from fluttering wings to MORphing flight (2011-2014), SARISTU – Smart Intelligent Aircraft Structures (2011-2015), AWAHL – Advanced Wing And High-Lift Design (2011-2012), FFAST – Future Fast Aeroelastic Simulation Technologies (2010-2013), SMORPH – Smart Aircraft Morphing Technologies (2007-2009), SimSAC – Simulating Aircraft Stability And Control Characteristics for Use in Conceptual Design (2008-2011), 3AS – Active Aeroelastic Aircraft Structures (2005-2007). Prof. Ricci is the Founder and Editorial Manager of open access International Journal of Aeroelasticity and Structural Dynamics (www.asdjournal.org). He also serves as reviewer for many international scientific journals, such as Aeronautical Journal, Aerospace Science and Technology, AIAA Journal, Journal of Aircraft, Journal of Intelligent Material Systems and Structures, Mechanism and Machine Theory, Chinese Journal of Aeronautics, Computer-Aided Civil and Infrastructure Engineering, and Statistics and Computing. He is a Member of the International Forum of Structural Dynamics and Aeroelasticity (IFASD) Committee, a Member of the International Council of Aeronautical Sciences (ICAS) Committee, a Member of the ACARE Italy Committee, a Member of Aeroelastic Prediction Workshop (AeP) and a Member of RTO AVT 173 panel on Virtual Prototyping Using Advanced MDO. He is also a member of three professional societies: American Institute of Aeronautics and Astronautics (AIAA), Royal Aeronautical Society (RAeS) and Associazione Italiana di Aeronautica e Astronautica (AIDAA). He has been part of the Scientific committees of several international conferences, such as the International Forum of Structural Dynamics and Aeroelasticity (IFASD), the International Council of Aeronautical Sciences (ICAS) and ECCOMAS Thematic Conferences. He has given many invited lectures in Europe and USA on morphing aircraft and conceptual design tools for fast structural sizing and aeroelastic analysis and optimization. Prof. Ricci leads the AeroStructures Design Lab (ASDL), specializing in the development of multi-fidelity analysis and design methods to enable fast and efficient generation of aero-structural models for new, environmentally friendly aerospace systems. The availability of a structural model since the beginning of design loop, i.e. at the conceptual design level, allows the designer to immediately evaluate the potential impacts from aeroelasticity on the definition of global aircraft design parameters, as well as the possible benefits from new materials and technology like morphing, in terms of global performances and weight saving. Current topics of research interest include: Automatic generation of low-medium fidelity aero-structural models; Fast structural sizing, aeroelastic analysis and optimization; Multi-objective topological optimization of compliant structures for morphing application; and Active aeroelastic control, including wind tunnel validation.



Prof. Dr. Mehmet Yildiz is the Vice President for Research and Development at Sabanci University. He completed his undergraduate education in 1996 with the first rank in Metallurgical and Materials Engineering Department at Yıldız Technical University and received his MSc. degree also from the Metallurgical and Materials Engineering department at Istanbul Technical University in 2000. Between 1996-2000, he worked in two different companies as an R&D engineer and a project manager in the fields of welding and non-destructive testing. He obtained his Ph.D. degree in 2005 from Mechanical Engineering Department at the University of Victoria, BC, Canada in the fields of computational fluid dynamics and semiconductor single crystal growth and then worked as a research associate and lecturer in the same department until 2007. In 2007, he joined Sabancı University, Faculty of Engineering Natural Sciences as a faculty member in Materials Science and Engineering Program and since 2013, has been leading the effort of establishing Sabancı University-Integrated Manufacturing Technologies Research and Application Center and its industrial leg, Composite Technologies Center of Excellence with Kordsa. He sits in the R&D advisory board of Kastamonu Entegre. He contributed to the establishment of Manufacturing Engineering graduate program at Sabancı University. Dr. Yildiz's areas of expertise include advanced composite materials, nanocomposites, structural health monitoring and computational mechanics. He published more than 85 SCI indexed high impact factor journals, 8 book chapters, prepared and presented over 135 conference papers and graduated more than 28 MSc. and Ph.D. students.

Programme Overview

Wednesday, July 19 th	
8:20 - 8:45	Registration
8:45 - 9:00	Opening Ceremony
9:00 - 10:00	Keynote Lecture I Charbel Farhat, Stanford University
10:00 - 10:20	Coffee Break
10:20 - 12:00	Session 1 - Multi-Disciplinary Optimization I
12:00 - 13:00	Lunch
13:00 - 14:40	Session 2 - Discipline Analysis Models
14:40 - 15:00	Coffee Break
15:00 - 16:40	Session 3 - Systems Engineering and Integration
Thursday, July 20 th	
9:00 - 10:00	Keynote Lecture II Antônio da Costa, Airbus
10:00 - 10:20	Coffee Break
10:20 - 12:00	Session 4 - Design Optimization
12:00 - 13:00	Lunch
13:00 - 14:40	Session 5 - Aerospace Design and Integrated Systems
14:40 - 15:00	Coffee Break
15:00 - 16:40	Session 6 - Multi-Disciplinary Optimization II
19:30 - 22:00	Conference Dinner
Friday, July 21 st	
9:00 - 10:00	Keynote Lecture III Sergio Ricci, Politecnico di Milano
10:00 - 10:20	Coffee Break
10:20 - 11:40	Session 7 - Multi-Disciplinary Optimization III
11:40 - 12:40	Keynote Lecture IV Mehmet Yildiz, Sabanci University
12:40 - 12:50	Closing Ceremony

CONFERENCE PROGRAM

Wednesday, July 19th

Keynote Lecture I – Charbel Farhat, Stanford University

- 1 DIFFERENTIABLE EMBEDDED BOUNDARY METHOD FOR EFFICIENT MULTIDISCIPLINARY DESIGN ANALYSIS AND OPTIMIZATION
Charbel Farhat

Session 1 – Multi-Disciplinary Optimization I

- 2 DISCIPLINARY SURROGATES FOR GRADIENT-BASED OPTIMIZATION OF MULTIDISCIPLINARY SYSTEMS
I. Cardoso, S. Dubreuil, N. Bartoli, C. Gogu, M. Salaün and R. Lafage
- 22 AEROELASTIC ANALYSIS OF HIGH ASPECT RATIO AND STRUT-BRACED WINGS
Y. Le Lamer, J. Morlier, E. Benard and P. He
- 34 AN UNCERTAINTY QUANTIFICATION METHOD BASED ON PROPER ORTHOGONAL DECOMPOSITION AND POLYNOMIAL CHAOS EXPANSION
L. Battaglia, F. Carlini, A. Clarich and R. Russo
- 54 AEROELASTIC OPTIMIZATION OF A CELLULAR FLYING CAR WING USING THOMPSON SAMPLING EFFICIENT MULTIOBJECTIVE OPTIMIZATION (TS-EMO) ALGORITHM
S. Wu and T. Yokozeki
- 74 A PRELIMINARY LOW-FIDELITY MDO APPROACH FOR LOAD ALLEVIATION THROUGH MOVABLES ON HAR WING
D. Muradas, S. Marquier, J. Morlier and C. Gogu

Session 2 – Discipline Analysis Models

- 88 AN ARBITRARY LAGRANGIAN-EULARIAN ALGORITHM TO SOLVE COMPRESSIBLE FLOW PROBLEMS: HEMLAB
E. Aksoy and M. Sahin
- 102 INTEGRATION OF LIFE CYCLE ASSESSMENT METHODOLOGY AS AN ENVIRONMENT DISCIPLINE MODULE IN MULTIDISCIPLINARY ANALYSIS AND OPTIMIZATION FRAMEWORK
T. Bellier, J. Morlier, C. Bil, A. Urbano and A. Pudsey
- 120 FIXED-WING UAV MODEL IDENTIFICATION FOR LONGITUDINAL MOTION USING FIRST-ORDER MODELS AND LIMITED FLIGHT TESTING
N. M. B. Matos and A. C. Marta
- 142 VALIDATION OF EXTENDED FAILURE MODELS AND CRITERIA FOR AEROSPACE COMPOSITES
G. Corrado, J. Reinoso and A. Arteiro
- 157 DEVELOPMENT OF COMPUTATIONAL AEROELASTIC ANALYSIS TOOLS AND UNCERTAINTY QUANTIFICATION TECHNIQUES FOR RELIABLE FLUTTER PREDICTION
A. Fereidooni, G. Lue, M. Yano and A. Grewal

Session 3 – Systems Engineering and Integration

- 158 DESIGN OPTIMIZATION AND HUMAN SUBJECT SHAKER TESTING OF AN ACTIVE HELICOPTER SEAT SYSTEM
Y. E. Chen, A. Fereidooni, R. Laliberte and V. Wickramasinghe
- 173 ADAPTIVE AIRBAG SYSTEMS FOR PROTECTION OF GENERAL AVIATION
J. Holnicki-Szulc, R. Faraj, C. Graczykowski, G. Mikulowski, P. Pawlowski, A. Świercz, Z. Wolejsza, L. Knap, K. Sekula, D. Wiacek
- 184 OPTIMAL MULTI-SENSOR OBSTACLE DETECTION SYSTEM FOR SMALL FIXED-WING UAV
M. Portugal and A. C. Marta

- 209 SOFTWARE SUBSYSTEM AS A NEW CONCEPT IN SATELLITE SYSTEM ARCHITECTURE
J. P. L. Monteiro, P. J. S. Gil and R. M. Rocha
- 222 A SYSTEM-BASED APPROACH FOR TECHNOLOGY ROADMAPING ASSISTED BY VISUAL ANALYTICS
G. Krupa, A. Spinelli and T. Kipouros

Thursday, July 20th

Keynote Lecture II – António da Costa, Airbus

- 239 AIRBUS - PIONEERING SUSTAINABLE AEROSPACE
António da Costa

Session 4 – Design Optimization

- 240 WING STRUCTURAL DESIGN FOR A MAME UAV USING HIGH-FIDELITY NUMERICAL TOOLS
V. M. T. Silva, N. M. B. Matos and A. C. Marta
- 260 WING AERODYNAMIC DESIGN FOR A MAME UAV USING HIGH-FIDELITY NUMERICAL TOOLS
R. S. Gameiro, N. M. B. Matos and A. C. Marta
- 284 NEURAL LEVEL SET TOPOLOGY OPTIMIZATION USING UNFITTED FINITE ELEMENTS
C. N. Mallon, A. W. Thornton, M. R. Hill and S. Badia
- 310 DESIGN OPTIMIZATION OF TRUSS STRUCTURES USING A NON- UNIFORM CELLULAR AUTOMATA PARADIGM
M. E. Bouzouiki, R. Sedaghati and I. Stiharu
- 316 STRESS-BASED SPATIAL GRADIENT RECONSTRUCTION FOR SHAPE SENSITIVITY ANALYSIS
R. A. Canfield

Session 5 – Aerospace Design and Integrated Systems

- 368 FROM MDO TO MANUFACTURING: APPLICATION CASE FOR UNMANNED AERIAL VEHICLES
L. F. T. Fernandez, M. Bronz, T. Lefebvre and N. Bartoli
- 379 SKY SAILING OF TETHERED AEROSTATS FOR EFFICIENT AERIAL MONITORING
A. Świercz, C. Graczykowski, L. Knap, Z. Wolejsza and J. Holnicki-Szulc
- 388 EXPLORING THE POTENTIAL OF DEEP LEARNING IN OPTIMIZING AN AERIAL PHOTOGRAMMETRY MISSION
S. Sotouni, M. Tucso, K. Gupta, I. Mantegh and H. Najjaran
- 406 COMPUTATIONAL CHALLENGES IN THE MODULAR DESIGN OF FUTURE AIRCRAFT CONCEPTS WITH FLUTTER CONSTRAINTS
A. Cea, P. Nagy, N. Roussouly, R. Palacios, M. Fossati
- 439 INTEGRATION OF A THERMAL MANAGEMENT SYSTEM IN A HYBRID ELECTRIC AIRCRAFT – FUT-PRINT₅₀
F. R. Barbosa, H. F. Teza, F. I. K. Odaguil, R. Gandolfi, D. Eisenhut, F. Brenner, J. Mangold, N. Moebs, D. Bento

Session 6 – Multi-Disciplinary Optimization II

- 452 MULTI-OBJECTIVE BAYESIAN OPTIMIZATION WITH MIXED-CATEGORICAL DESIGN VARIABLES FOR EXPENSIVE-TO-EVALUATE AERONAUTICAL APPLICATIONS
N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J. H. Bussemaker, G. Donelli, J. M. Gomes de Mello, M. Mandorino, P. Della Vecchia
- 478 AN EXPLORATORY STUDY OF OPEN-SOURCE FRAMEWORKS FOR MDAO
R. di Giuseppe, S. Delbecq, V. Budinger and V. Pauvert
- 495 THERMO-MECHANICAL LEVEL-SET TOPOLOGY OPTIMIZATION OF A LOAD CARRYING BATTERY PACK FOR ELECTRIC AIRCRAFT
A. T. R. Guibert, M. Bookwala, A. Cronk, Y. S. Meng and H. A. Kim
- 505 TOPOLOGY OPTIMIZATION OF A SOLID GRAIN HYBRID ROCKET LAUNCHER
M. Melis, A. Souza and F. Afonso

- 525 MULTI-OBJECTIVE AEROELASTIC ANALYSIS AND OPTIMIZATION USING SURROGATE MODELS
A. Lunghitano, F. Afonso and A. Suleman

Friday, July 21st

Keynote Lecture III – Sergio Ricci, Politecnico di Milano

- 533 AEROELASTIC MULTIDISCIPLINARY OPTIMIZATION: ENHANCING AIRCRAFT PERFORMANCE AND SAFETY THROUGH NUMERICAL TOOLS AND EXPERIMENTAL VALIDATION
Sergio Ricci

Session 7 – Multi-Disciplinary Optimization III

- 534 LEVEL SET TOPOLOGY OPTIMIZATION FOR COUPLING MULTIPHYSICS WITH AUTOMATIC DIFFERENTIATION
A. Neofytou and H. A. Kim
- 545 DEVELOPMENT OF A FE CODE FOR ADJOINT-BASED COUPLED AEROSTRUCTURAL OPTIMISATION
L. Scalia, R. Cavallaro and A. Cini
- 565 AN EFFICIENT NUMERICAL MODELLING FOR AERODYNAMIC SHAPE OPTIMIZATION
S. Pucciarelli, F. Lau and A. Suleman
- 579 TOPOLOGY OPTIMIZATION OF ORIGAMI STRUCTURES BASED ON CREASE PATTERN AND AXIAL RIGIDITY
V. Cretella, A. Sohoulı, A. Pagani and A. Suleman

Keynote Lecture IV – Mehmet Yildiz, Sabanci University

- 599 BUILDING BLOCKS TOWARDS ADVANCED THERMOPLASTIC COMPOSITES FOR SUSTAINABLE AVIATION: INTEGRATION OF MATERIAL, PROCESS AND JOINING, DESIGN, VALIDATION, AND MONITORING
Mehmet Yildiz

- 601 **Index of Authors**



DIFFERENTIABLE EMBEDDED BOUNDARY METHODS FOR EFFICIENT MULTIDISCIPLINARY DESIGN ANALYSIS AND OPTIMIZATION

Charbel Farhat^{*1,2} and Jonathan Ho³

1: Department of Aeronautics and Astronautics
Stanford University
Stanford, CA 94305, USA
cfarhat@stanford.edu, <https://aa.stanford.edu/>

2: Institute for Computational and Mathematical Engineering
Stanford University
Stanford, CA 94305, USA
<https://icme.stanford.edu/>

3: Luminary Cloud
Redwood City, CA 94063, USA
SecondAuthor@company456.com, <https://www.luminarycloud.com>

Abstract. *In the context of CFD and fluid-structure interaction (FSI), embedded boundary methods (EBMs) are Eulerian methods that operate on non body-fitted fluid meshes in which discrete representations of obstacle surfaces are embedded. They are attractive for numerous reasons. They introduce such a high degree of automation in the discretization of complex computational fluid domains that they can almost be considered as mesh free methods. They are also the most robust solution methods for flow problems past obstacles that undergo large motions, deformations, shape changes, and/or surface topology changes. Such problems arise in FSI and multidisciplinary design, analysis, and optimization (MDAO). However, EBMs typically generate discrete events that are sources of roughness and spurious oscillations in the flow results computed at an embedded, discrete, boundary surface or in its vicinity. At best, such numerical flaws are sufficiently small not to affect the quality of the computations. However, they inhibit differentiation with respect to any evolutions of the embedded surfaces. Therefore, they hinder the application of EBMs to the gradient-based solution of MDAO problems, where they are however pressingly needed to avoid remeshing and the pitfalls of transferring numerical results from one CFD mesh to another. For this reason, this lecture will present a new approach for constructing discrete-event-free and differentiable EBMs based on a new concept of a nodal status, that of a smoothness indicator nodal function, and a moving least squares approach for suppressing spurious oscillations from integral quantities computed on embedded discrete interfaces. The enabling capabilities of the proposed approach will be demonstrated for complex FSI problems such as limit cycle oscillations of complete fighter jet configurations in the transonic flow regime; and MDAO of aircraft configurations characterized by large shape adjustments and large surface topology changes in various flow regimes.*

Keywords: CFD, fluid-structure interaction, embedded boundary method, gradient-based optimization, limit cycle oscillations, large shape adjustments



DISCIPLINARY SURROGATES FOR GRADIENT-BASED OPTIMIZATION OF MULTIDISCIPLINARY SYSTEMS

Inês Cardoso^{1*,2}, Sylvain Dubreuil¹, Nathalie Bartoli¹, Christian Gogu²,
Michel Salaün², Rémi Lafage¹

1: ONERA, DTIS
Toulouse, France

{ines.cardoso, sylvain.dubreuil, nathalie.bartoli, remi.lafage}@onera.fr

2: Institut de Clément Ader (ICA), CNRS, UPS, INSA, ISAE, Mines Albi, Université de
Toulouse

Toulouse, France

{christian.gogu, michel.salaun}@isae-superaero.com

Abstract. *Many engineering problems are described by complex multidisciplinary systems, whose behavior is dictated by a non-linear system of equations called multidisciplinary analysis (MDA). When optimizing these systems, the resolution of the MDA at each evaluated design space point often represents a heavy computational burden, particularly when high-fidelity solvers are used. In this work, we address the high computational cost of the MDA by replacing the disciplinary solvers by Gaussian Process (GP) surrogate models. This approach allows to uncouple the disciplinary solvers, similarly to what is done in the Individual Disciplinary Feasible formulation. Moreover, a procedure for the adaptive enrichment of the disciplinary surrogates is proposed to reduce the uncertainty of the surrogates in the explored regions of the design space. The use of GP surrogates further presents the advantage of an analytical gradient computation, which allows for an easy implementation with gradient-based solvers. The performance of the proposed approach has been tested on the analytical benchmark Sellar test case, as well as an aircraft design problem which couples the aerodynamics and structural disciplines. Both test cases show that the proposed approach requires less disciplinary solver calls than classical gradient-based solvers. Finally, the proposed methodology has been integrated in ONERA's WhatsOpt collaborative environment. WhatsOpt generates the OpenMDAO skeleton code, where the implementations of the disciplines can then be plugged into. Thanks to the graphical interface of WhatsOpt, users can easily implement their models and choose the proposed approach to solve the optimization problem.*

Keywords: Multidisciplinary optimization, Gaussian processes, Disciplinary surrogates

1 INTRODUCTION

Multidisciplinary Design and Optimization (MDO) has become a popular tool when designing complex systems involving several disciplines. By taking into account the interactions between the participating disciplines, MDO is capable of finding more relevant solutions, compared to when the disciplines are optimized sequentially. Formally, the non-linear coupled system which describes the interactions between the disciplines is called Multidisciplinary Design Analysis (MDA). The MDA is often solved using partitioned approaches, where the disciplinary solvers are called iteratively until the equilibrium of the non-linear coupled system is found. Due to the resulting numerical cost, the use of high fidelity solvers directly in the MDA is often challenging. Many studies have thus focused on the development of MDO formulations which aim to ease the computational burden [1]. Among these formulations are the multidisciplinary feasible (MDF) and the individual disciplinary feasible (IDF) approaches [2]. In the MDF approach, the MDA is solved at every optimizer iteration. As a result, a physically relevant solution is obtained at every optimization stage. On the other hand, in the IDF approach, the optimizer handles both design and coupling variables, essentially uncoupling the disciplinary solvers, but coupling the resolution of the optimization problem and of the MDA. The IDF approach often results in a lower computational cost than the MDF approach, but a physically non-feasible solution may be returned if the optimization fails or is terminated early.

Another way to address the high computational cost of the MDO is through the use of surrogate models. For instance, one can build an inexpensive approximate function of the MDA quantity of interest based on a given design of experiments. This strategy, however, is not dedicated to multidisciplinary systems and does not take advantage of the partitioned methods used to solve the MDA. Moreover, any enrichment of the built approximate function requires a new resolution of the whole MDA. Therefore, a more interesting solution is to build a surrogate model of each disciplinary solver. This approach has been applied in an MDO context in [3, 4], where disciplinary solvers were used to perform optimization using a genetic algorithm. Disciplinary surrogates have also been used in [5] to perform MDA of an aircraft wing. For the cited applications, disciplinary surrogates were shown to reduce the computational cost, but were required to be sufficiently accurate throughout the entire design space. In an optimization context, where the interest lies only in certain regions of the design space, this may represent a waste of computational effort. A more efficient approach is thus to use adaptive sampling strategies, where the disciplinary solvers are enriched only at points selected by the optimizer. Such an adaptive strategy is proposed in [6], where Gaussian Processes (GPs) are used to approximate each disciplinary solver. The uncertainty introduced by the disciplinary GPs in the resulting random MDA is then quantified and used to determine if the built disciplinary surrogates should be enriched.

In this work we propose to exploit the analytical computation of the disciplinary gradients that can be obtained thanks to the use of disciplinary GPs. Since gradient information is often not available for high-fidelity solvers, the use of gradient-based optimization with expensive disciplinary solvers has been limited. When necessary, gradient information is obtained using approximation methods, which, for coupled systems, can require long computational times. An alternative to these approaches is the coupled-adjoint method [7], which has been used in recent literature to perform MDO thanks to its computational efficiency [8–10]. The coupled-adjoint method remains, however, an intrusive approach

requiring derivative information at a disciplinary solver level. The methodology proposed in this work stands apart from the current state-of-the-art approaches as it avoids the expensive computation of the disciplinary gradients, while still retaining a non-intrusive approach, fit for use with black box disciplinary solvers.

Thanks to the simplicity of the proposed optimization strategy, the approach described in this work is easily integrated within ONERA’s WhatsOpt collaborative environment [11]. WhatsOpt generates the OpenMDAO [9] skeleton code, where implementations of the disciplinary solvers can then be plugged into. By selecting the dedicated option in WhatsOpt graphical interface, the required code to build the disciplinary surrogates is generated. Then, users can select any of the optimizers available in OpenMDAO (e.g. SLSQP [12], COBYLA [13]) to solve the optimization problem.

The remainder of this work is organized as follows. Section 2 introduces the methodology proposed in [6] to build the disciplinary surrogates and solve the resulting random MDA. The analytical computation of the disciplinary gradients used to perform gradient-based optimization is equally detailed in this section. Section 3 describes the existing implementations with ONERA’s WhatsOpt collaborative environment. In Section 4 the proposed approach is applied to the analytical test case as well as a wing structural sizing engineering problem. Finally, some conclusions are provided in Section 5.

2 DISCIPLINARY SURROGATES FOR MDO

2.1 Problem definition

The purpose of this work is to solve MDO problems of the form:

$$\begin{aligned} \arg \min_{\mathbf{x} \in \mathcal{X}} \quad & f_{\text{obj}}(\mathbf{x}, \mathbf{y}_{c_{\text{obj}}}^*(\mathbf{x})) \\ \text{s.t.} \quad & g_m(\mathbf{x}, \mathbf{y}_{c_{g_m}}^*(\mathbf{x})) \geq 0, \quad m = 1, \dots, n_g \\ & h_r(\mathbf{x}, \mathbf{y}_{c_{h_r}}^*(\mathbf{x})) = 0, \quad r = 1, \dots, n_h \end{aligned} \quad (1)$$

where f_{obj} denotes the objective function, which depends on the set of design variables $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ as well as on the set $\mathbf{y}_{c_{\text{obj}}}^*(\mathbf{x})$ composed of some (or all) of the converged coupling variables. Similarly, $g_m(\mathbf{x}, \mathbf{y}_{c_{g_m}}^*(\mathbf{x}))$ denotes the set of inequality constraints and $h_r(\mathbf{x}, \mathbf{y}_{c_{h_r}}^*(\mathbf{x}))$ is the set of equality constraints. In practice $\mathbf{y}_{c_{\text{obj}}}^*(\mathbf{x})$, $\mathbf{y}_{c_{g_m}}^*(\mathbf{x})$ and $\mathbf{y}_{c_{h_r}}^*(\mathbf{x})$ are subsets of $\mathbf{y}^*(\mathbf{x}) = \{y_i^*(\mathbf{x}), i = 1, \dots, n_d\}$, where n_d is the number of disciplines in the system and $\mathbf{y}^*(\mathbf{x})$ is the solution of the non-linear system (MDA), written as:

$$y_i = f_i(\mathbf{x}, \mathbf{y}_{c_i}), \quad i = 1, \dots, n_d \quad \forall \mathbf{x} \in \mathcal{X} \quad (2)$$

where f_i is the disciplinary solver corresponding to discipline i , \mathbf{y}_{c_i} is the vector of coupling variables that influence y_i and c_i is a subset of $\{1, \dots, n_d\} \setminus \{i\}$. It is assumed that Eq. (2) has a unique solution for any \mathbf{x} belonging to the design space \mathcal{X} .

As previously stated, it is proposed to solve the MDA defined in Eq. (2) by replacing the disciplinary solvers by GP surrogate models. In the following, the steps required to obtain the disciplinary GPs and solve the resulting random MDA are described.

2.2 Building disciplinary surrogates

The first step of the proposed approach is to replace all disciplinary solvers f_i , $i = 1, \dots, n_d$ by a GP surrogate. Each GP is built from a Design of Experiments DoE_{f_i} sampled

over the space $\mathcal{X} \times \mathcal{C}^i$, where \mathcal{C}^i denotes the space of the coupling variables y_{c_i} associated with discipline i . The following approximation for each disciplinary solver is obtained:

$$\hat{f}_i(\mathbf{x}, y_{c_i}) = \mu_i(\mathbf{x}, y_{c_i}) + \epsilon_i(\mathbf{x}, y_{c_i}) \quad (3)$$

where $\mu_i(\mathbf{x}, y_{c_i})$ is the mean function of the GP and $\epsilon_i(\mathbf{x}, y_{c_i})$ is a GP of zero mean conditioned on the respective disciplinary DoE. Note that the bounds for \mathcal{C}^i are often not known and an initial guess must be made. In this work, we assume that an expert judgment can be used to obtain a reasonable initial guess for the coupling variable space.

Once each disciplinary solver has been replaced by the corresponding GP approximation \hat{f}_i , we obtain the following non-linear system of equations:

$$\hat{y}_i(\mathbf{x}, \hat{\mathbf{y}}_{c_i}) = \mu_i(\mathbf{x}, \hat{\mathbf{y}}_{c_i}) + \epsilon_i(\mathbf{x}, \hat{\mathbf{y}}_{c_i}) \quad i = 1, \dots, n_d \quad (4)$$

An illustration of the proposed approximation for a two-discipline system at a fixed value $\mathbf{x} = \mathbf{x}^*$ is shown in Figure 1, where the dotted lines represent the real disciplinary solvers f_i while the mean values of the GP approximations \hat{f}_i are given by continuous lines.

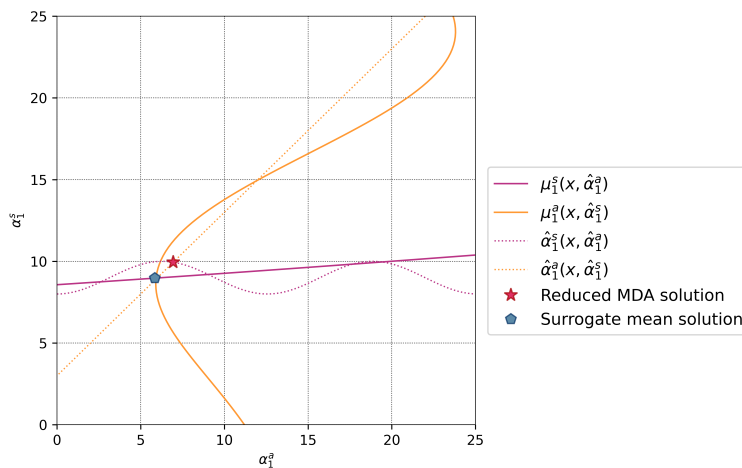


Figure 1: Illustration of the obtained MDA when the disciplinary solvers are approximated by GPs. Comparison of true MDA solution with the surrogate mean solution.

Figure 1 shows that, for the current disciplinary GP approximations, the surrogate mean solution does not correspond to the real MDA solution at $\mathbf{x} = \mathbf{x}^*$. In the following section, we will describe how to enrich the disciplinary surrogates until the surrogate mean solution is sufficiently accurate at $\mathbf{x} = \mathbf{x}^*$.

Before we proceed, however, it is worth addressing the case where the coupling variables are not scalar, but rather high dimensional vectors, such as displacement or pressure fields. Additional operations are required when this is the case for one or all disciplinary solvers, since, in order to build the disciplinary GPs, a low dimensional input space is required. Fortunately, this issue has already been addressed in [14], where it is proposed to reduce the coupling variable space using Proper Orthogonal Decomposition (POD) before obtaining the disciplinary GP surrogates. In this work, we will employ the same strategy proposed in [14] any time the coupling variables are high-dimensional vectors. A brief description of the main steps is provided in the following.

Let the high-dimensional vector y_i be the output of the disciplinary solver f_i . Using POD, y_i may be represented in a linear subspace according to:

$$y_i(\mathbf{x}, \mathbf{y}_{c_i}) \approx \tilde{y}_i(\mathbf{x}, \mathbf{y}_{c_i}) = \phi_0^i + \sum_{j=1}^{n_i} \alpha_j^i(\mathbf{x}, \mathbf{y}_{c_i}) \phi_j^i \quad (5)$$

where $\phi_j^i, j = 1, \dots, n_i$ are the basis vectors, α_j^i are the corresponding coefficients and ϕ_0^i is a constant vector. Note that n_i determines the number of vectors of the basis associated with solver f_i and is typically chosen so that a given percentage of the snapshots variance is captured by the POD basis.

Using the approximation defined in Eq. (5), we can rewrite the reduced MDA as follows:

$$\alpha_1^i(\mathbf{x}, \boldsymbol{\alpha}^{c_i}), \dots, \alpha_{n_i}^i(\mathbf{x}, \boldsymbol{\alpha}^{c_i}) = \Phi^i(\mathbf{x}, \boldsymbol{\alpha}^{c_i}) \quad i = 1, \dots, n_d \quad (6)$$

where Φ_i denotes the operation allowing to obtain the POD coefficients for the disciplinary solver f_i and $\boldsymbol{\alpha}^{c_i}$ denote the POD coefficients of the coupling variables influencing the value of $\alpha_j^i, j = 1, \dots, n_i$. In this new reduced space, it becomes possible to build the disciplinary GP approximations. For the disciplinary solver f_i , the idea is to replace each coefficient of the POD basis by a GP, and to write the resulting reduced MDA as:

$$\hat{\alpha}_j^i = \mu_j^i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i}) + \epsilon_j^i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i}) \quad \forall i = 1, \dots, n_d \quad \forall j = 1, \dots, n_i \quad (7)$$

Note that, in this case, expert judgment cannot be used to provide an initial guess for the coupling variable space. A solution for the construction of an appropriate DoE for the training of the POD basis is detailed in [14]. To avoid additional disciplinary solver calls, this same DoE is then used to train the disciplinary GPs.

2.3 Solving the random MDA

The previous section described how GP approximations of the disciplinary solvers can be obtained. In this section, we will focus on how to solve the random MDA given by Eq. (4). As was shown in Figure 1, using only the GPs mean value, we are unable to accurately solve the MDA. This is because the uncertainty of the disciplinary surrogates is not accounted for. This uncertainty is given by the GPs variance, which introduces randomness in the MDA. Thus, the first step in solving the MDA is to model the uncertainty of the disciplinary surrogates and, therefore, evaluate their accuracy.

It is proposed in [6] to model the uncertainty of the disciplinary GPs by considering perfectly dependent GPs, whose correlation function is constant and equal to one. The random MDA is rewritten as:

$$\hat{y}_i'(\mathbf{x}, y_{c_i}) = \mu_i(\mathbf{x}, y_{c_i}) + \sigma_i(\mathbf{x}, y_{c_i}) \xi_i \quad i = 1, \dots, n_d \quad (8)$$

where $\sigma_i(\mathbf{x}, y_{c_i})$ is the standard deviation and ξ_i is a standard Gaussian random variable. It should be noted that the solution of Eq. (8) depends on the sample $\Xi = \{\xi_i, i = 1, \dots, n_d\}$ drawn and thus, by drawing different samples, different solutions of the MDA are obtained. Figure 2 illustrates how different MDA solutions can be obtained at a given $\mathbf{x} = \mathbf{x}^*$ when different samples of the random vector Ξ are drawn.

To characterize the dispersion of the random MDA solutions, a sufficiently large number of samples should be drawn. Nonetheless, because the disciplinary GPs are inexpensive

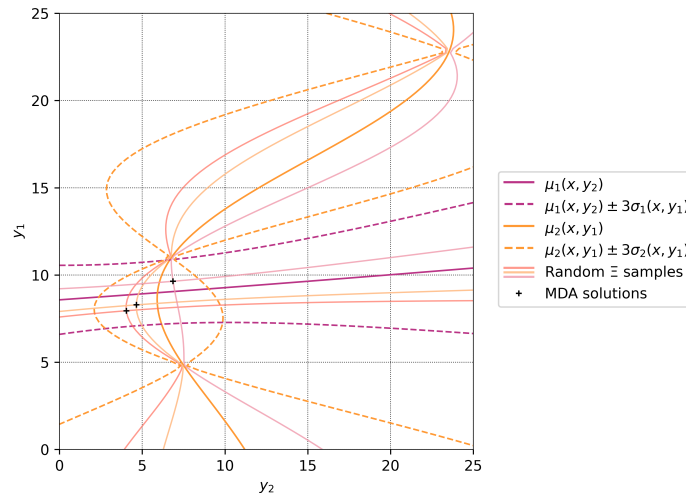


Figure 2: Illustration of the different MDA solutions obtained when drawing different samples of the random vector Ξ .

to evaluate, a direct Monte Carlo (MC) method may be used. Then, using the n_{MC} MC simulations, we can estimate the coefficient of variation (CV) of the system's quantities of interest (\widehat{QoI}), typically the objective and constraint functions. Note that, due to the use of disciplinary GPs, these quantities are now random variables. Their coefficient of variation may be written as:

$$CV(\widehat{QoI}(\mathbf{x}, \Xi)) = \frac{\sqrt{\mathbb{V}(\widehat{QoI}(\mathbf{x}, \Xi))}}{\mathbb{E}(\widehat{QoI}(\mathbf{x}, \Xi))} \quad (9)$$

where \mathbb{E} is the expected value and \mathbb{V} is the variance. If the CV of all QoI at a given $\mathbf{x} = \mathbf{x}^*$ is below a given threshold, then the disciplinary surrogates are sufficiently accurate and the surrogate mean solution at $\mathbf{x} = \mathbf{x}^*$ is approximately the real MDA solution. Otherwise, the disciplinary surrogates should be enriched at $\mathbf{x} = \mathbf{x}^*$.

In [6] it is proposed to enrich both disciplinary solvers simultaneously, however, this is not computationally efficient as only one of the disciplinary surrogates could be inaccurate. An alternative procedure is thus proposed in [14], where the less accurate disciplinary surrogate is chosen according to the Sobol sensitivity indices [15], given by:

$$S_i(\mathbf{x}) = \frac{\mathbb{V}[\mathbb{E}[\widehat{QoI}(\mathbf{x}) \mid \xi_i]]}{\mathbb{V}[\widehat{QoI}(\mathbf{x})]}, \quad i = \{1, \dots, n_d\} \quad (10)$$

where S_i is the Sobol index associated with discipline i , and $\widehat{QoI}(\mathbf{x})$ is chosen as the quantity of interest with the highest CV. We note that, in this work, the Sobol sensitivity indices are estimated using a Polynomial Chaos Expansion approximation [16].

Once all sensitive indices are obtained, the discipline with the highest S_i is selected for enrichment. This process is repeated iteratively until the CV of all QoI is sufficiently small at $\mathbf{x} = \mathbf{x}^*$. To illustrate the enrichment process, Figure 3 shows the evolution of the disciplinary GPs until they are sufficiently accurate at $\mathbf{x} = \mathbf{x}^*$. Note that, in the given example, both disciplinary surrogates are enriched. However, there may be points in the

design space where the enrichment of only one of the disciplinary surrogates is enough to have a sufficiently accurate solution.

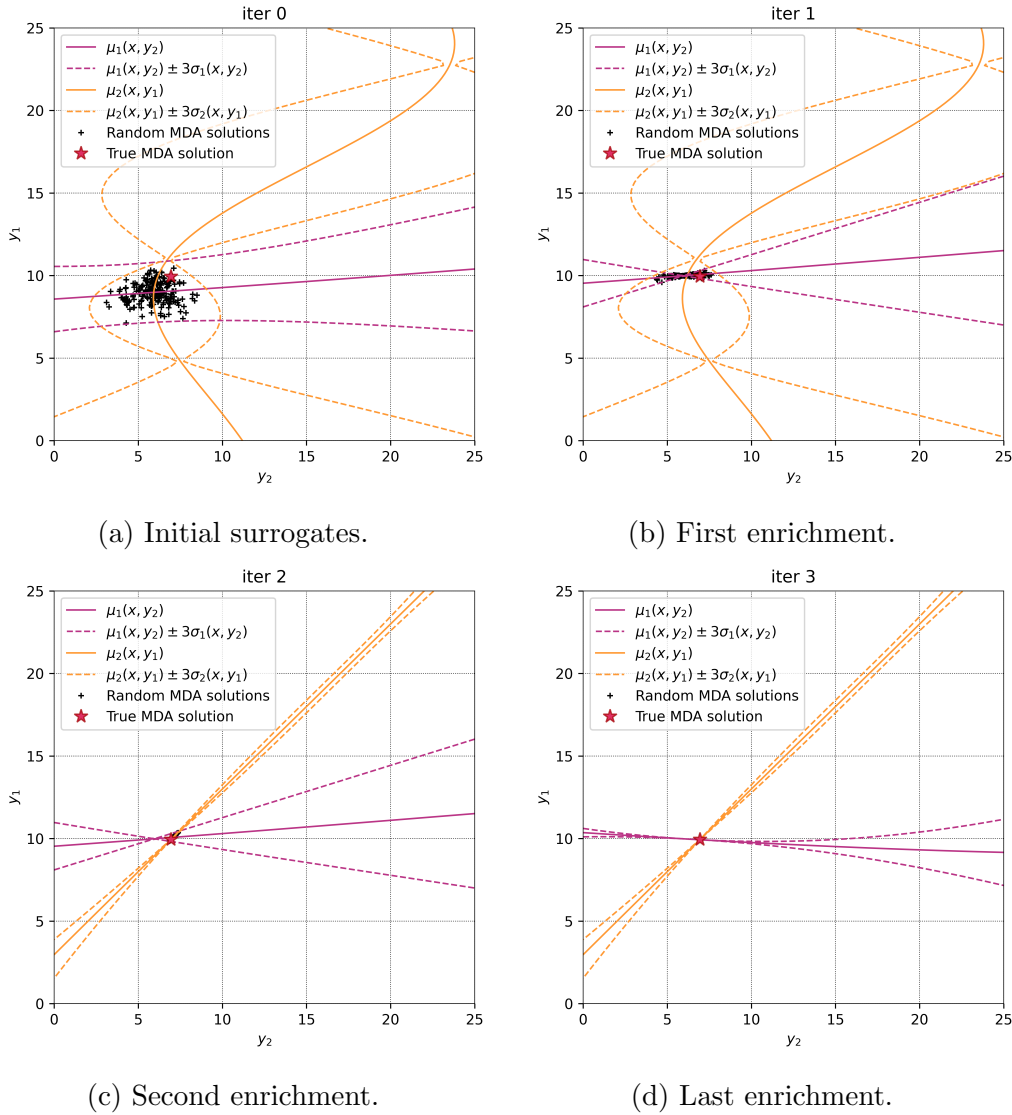


Figure 3: Illustration of the disciplinary surrogate enrichment process at $\mathbf{x} = \mathbf{x}^*$. (a) Initial disciplinary GPs. (b) Enrichment of first disciplinary surrogate. (c) Enrichment of the second disciplinary surrogate. (d) Re-enrichment of the first disciplinary surrogate. After this iteration, the disciplinary GPs are sufficiently accurate at $\mathbf{x} = \mathbf{x}^*$.

Using the above described enrichment strategy, it is possible to find the solution to Eq. (4), with a relatively small error, at any point $\mathbf{x} \in \mathcal{X}$. In the context of gradient-based optimization, enrichment of the disciplinary surrogates is made, whenever necessary, at points queried by the optimizer. This limits the computational effort to regions of the design space relevant for the optimization problem. The following section details how the derivatives of the disciplinary GPs outputs with respect to the design variables are obtained. Note that the derivatives of the objective and constraint functions with respect to the design variables may then be obtained using the chain rule.

2.4 Gradient computation

In the previous section we detailed how to solve the random MDA obtained when the disciplinary solvers are replaced by perfectly dependent GPs (see Eq. (8)). In this section, we are interested in obtaining the derivatives of the disciplinary surrogates with respect to the design variables in order to perform gradient-based optimization. For this purpose, we distinguish the case where the coupling variables are scalars from the case where they are high-dimensional vectors.

For the first case, where the coupling variables are scalars, the partial derivative of $\hat{y}'_i(\mathbf{x}, y_{c_i})$ with respect to the design variable x_k is written as:

$$\frac{\partial \hat{y}'_i(\mathbf{x}, y_{c_i})}{\partial x_k} = \frac{\partial \mu_i(\mathbf{x}, y_{c_i})}{\partial x_k} + \frac{\partial \sigma_i(\mathbf{x}, y_{c_i})}{\partial x_k} \xi_i \quad i = 1, \dots, n_d \quad (11)$$

where the partial derivatives of the mean and standard deviation functions depend on the choice of mean trend and correlation function used.

For the second case, where the coupling variables are high-dimensional vectors, we will here assume that model order reduction using POD approximation was first performed and that $\hat{y}'_i(\mathbf{x}, y_{c_i})$ is of the form:

$$\hat{y}'_i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i}) = \phi_0^i + \sum_{j=1}^{n_i} \hat{\alpha}_j^i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i}) \phi_j^i \quad (12)$$

where $\hat{\alpha}_j^i(\mathbf{x}, y_{c_i})$ denotes the GP approximation of the POD coefficient associated with the basis vector ϕ_j^i . In this case, the partial derivative of $\hat{y}'_i(\mathbf{x}, y_{c_i})$ with respect to the design variable x_k can be written as:

$$\frac{\partial \hat{y}'_i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i})}{\partial x_k} = \sum_{j=1}^{n_i} \left(\frac{\partial \mu_j^i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i})}{\partial x_k} + \frac{\partial \sigma_j^i(\mathbf{x}, \hat{\boldsymbol{\alpha}}^{c_i})}{\partial x_k} \xi_j^i \right) \phi_j^i \quad i = 1, \dots, n_d \quad (13)$$

The expressions presented above allow an analytical computation of the disciplinary gradients with respect to the design variables, which can be used to perform gradient-based optimization while still treating the disciplinary solvers as black boxes. The following section describes the integration of the proposed strategy within ONERA's WhatsOpt collaborative environment.

3 THE WHATSOPT COLLABORATIVE ENVIRONMENT

An important objective envisioned for the approach presented in this work is its rapid implementation and compatibility with any black box solver input in the system. To that end, the approach proposed in this work has been integrated with ONERA's WhatsOpt collaborative environment.

3.1 Basic WhatsOpt usage

WhatsOpt is a web application which allows users to define multidisciplinary analysis problems in terms of disciplines and data exchanges [11]. Once the problem has been defined, WhatsOpt generates the OpenMDAO [9] skeleton code where implementations of the disciplinary solvers can be plugged into. Figure 4 displays the user graphical interface when a generic two-discipline MDO problem is generated using WhatsOpt.

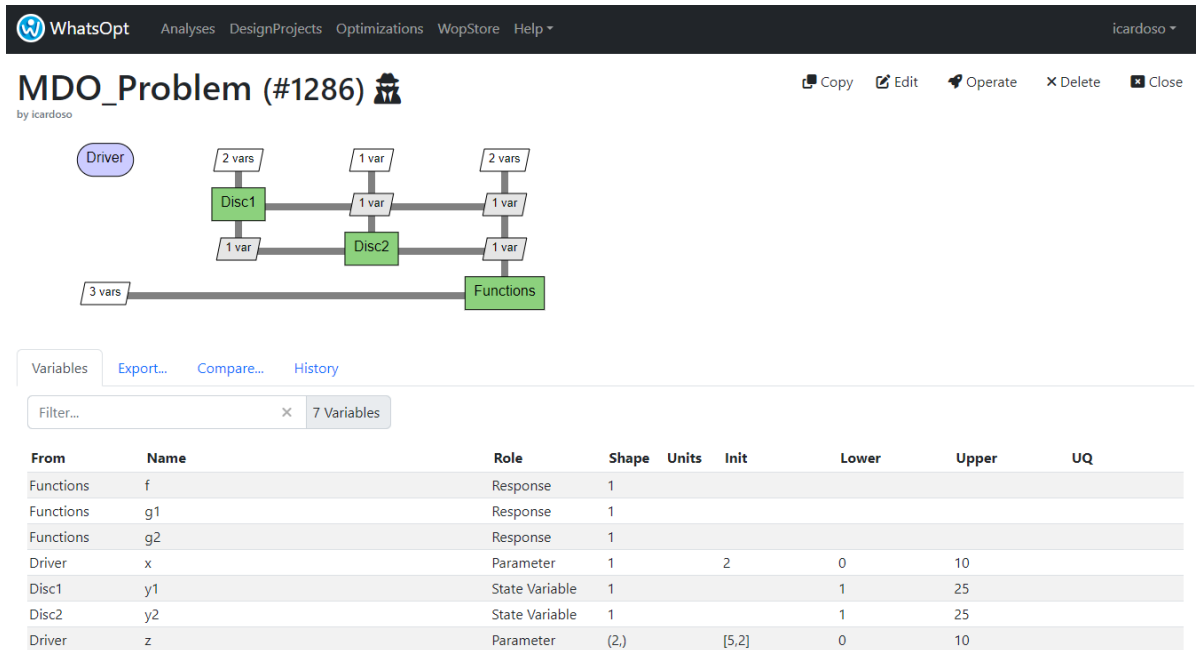


Figure 4: WhatsOpt graphical interface. Implementation of a generic two-discipline MDO problem.

Figure 4 shows that when implementing an MDO problem in WhatsOpt an interactive eXtended Design Structure Matrix (XDSM) [17] of the problem is generated automatically, allowing users to easily control the flow of information between the different disciplines. Additionally, an interactive table allows users to define all disciplinary inputs and outputs as design or state variables as well as objective or constraint functions. The bounds for the design variables can also be defined at this step.

Once users are done with the problem definition, they can request WhatsOpt to generate the required scripts to run the problem. For all disciplines, objective and constraint functions, WhatsOpt will generate a python script where the respective output computations should be implemented by the user. WhatsOpt will equally generate the required files to solve the MDA and MDO problems. The latter will use by default OpenMDAO solver options. The list of generated files for the problem represented in Figure 4 is:

- `Disc1_base.py`, `Disc2_base.py`, `Functions_base.py`: these files contain connection information (inputs and outputs) for all defined disciplines;
- `Disc1.py`, `Disc2.py`, `Functions.py`: these are python class files where the disciplines computations should be implemented by the user;
- `MDO_Problem_base.py`, `MDO_Problem.py`: these files contain the definition of the problem as a whole;
- `mda_init.py`, `run_mda.py`, `run_mdo.py`: these files allow to solve the MDA and MDO problems using the real disciplinary solvers.

Some additional files are also generated, but they are not relevant for the implementations proposed in this work. As a result we will abstain from providing further details on them.

3.2 Disciplinary surrogate option

To solve the MDA according to the approach proposed in Section 2 users must state that they want the disciplinary solvers to be replaced by surrogates, as proposed in the Efficient Global Multidisciplinary Design Optimization (EGMDO) algorithm [6]. This option is available under the OpenMDAO tab of the WhatsOpt graphical interface. Figure 5 displays the OpenMDAO menu, where the option 'EGMDO surrogate' has been ticked for the two disciplinary solvers of the MDO problem defined in Figure 4.

The screenshot shows the 'OpenMDAO' configuration window. It is divided into several sections:

- General:** Includes checkboxes for 'Use units', 'Use scaling', and 'Parallel Group (MPI)'. There is a 'Packaging' section with a 'Package Name' field containing 'mdo_problem' and a 'Driver' section with an 'Optimization' dropdown set to 'Scipy - SLSQP'. At the bottom are 'Save' and 'Reset' buttons.
- Components:** Contains two sections, 'Disc1' and 'Disc2'. Each has checkboxes for 'Implicit component', 'Support derivatives', and 'EGMDO surrogate'. The 'EGMDO surrogate' checkbox is checked for both.
- Nonlinear solver:** For 'Disc1', the 'Solver name*' is 'NonlinearBlockGS', 'Absolute error tolerance*' is '1e-8', 'Relative error tolerance*' is '1e-8', 'Maximum number of iterations (maxiter)*' is '10', and 'Level of solver traces*' is '1'. There is a 'Fail if not converged' checkbox.
- Linear solver:** For 'Disc2', the 'Solver name*' is 'ScipyKrylov', 'Absolute error tolerance*' is '1e-8', 'Relative error tolerance*' is '1e-8', 'Maximum number of iterations (maxiter)*' is '10', and 'Level of solver traces*' is '1'. There is a 'Fail if not converged' checkbox.

Figure 5: WhatsOpt OpenMDAO menu. Option 'EGMDO surrogate' selected for both disciplinary solvers.

It should be noted that when ticking the 'EGMDO surrogate' option users are required to provide an initial guess for the coupling variable space. This can be done in the interactive table shown in Figure 4. Then, all users have left to do is to ask WhatsOpt to generate the code. Since the option 'EGMDO surrogate' has been selected, WhatsOpt will automatically generate the necessary scripts to solve the problem using the approach presented in Section 2. In addition to the scripts described in Section 3.1, the following files are generated:

- **algorithms.py:** this is the main file containing the implementation of the proposed methodology, as it was described in Section 2;
- **doe_factory.py:** this file is responsible for the initial sampling used to build the disciplinary surrogates;
- **gp_factory.py:** this file serves both for the creation of the disciplinary surrogates as well as for the updating of the surrogates whenever enrichments are performed;
- **random_analysis.py:** this file solves the random MDA at n_{MC} samples of the random vector Ξ ;
- **MDO_Problem_egmda.py:** this file contains the problem definition, when the disciplinary solvers are replaced by their respective surrogates;
- **run_egmda.py**, **run_egmdo.py:** these files allow to solve the MDA and MDO problems using the disciplinary surrogates.

Assuming that the disciplinary solvers have been previously implemented in the respective `Disc1.py` and `Disc2.py` files, users don't need to make any additional imple-

mentations in order to solve the MDA or MDO problems using the approach proposed in Section 2. Some remarks should nonetheless be made concerning the generated scripts. Sampling of the design and coupling variable spaces is made using a Latin Hypercube Sampling [18] strategy. Moreover, the GP surrogates are built using the KRG class of the SMT: Surrogate Modelling Toolbox [19]. By default constant mean and squared exponential correlation functions are used. Finally, WhatsOpt generates all necessary code to compute the derivatives of the disciplinary surrogates outputs with respect to the design variables using the SMT `predict_derivatives` function. OpenMDAO will then use this information when gradient-based optimizers are chosen to solve the MDO problem. In the following we will use the code generated by WhatsOpt to solve an analytical benchmark test case, as well as an engineering structural sizing problem.

4 APPLICATIONS

In this section we will use the methodology described in Section 2 to solve two MDO problems. To that end, both problems will be defined in the WhatsOpt graphical interface and the generated scripts will be used to solve the optimization.

4.1 Analytical Sellar benchmark test case

The Sellar benchmark test case [20] is a constrained MDO problem using analytical disciplinary solvers. It is defined as:

$$\begin{aligned} \arg \min_{z \in \mathcal{Z}} \quad & z_3^3 + z_2 + y_1^* + \exp(-y_2^*) \\ \text{s.t.} \quad & 3.16 - y_1^* \leq 0 \\ & y_2^* - 24 \leq 0 \end{aligned} \tag{14}$$

where $z = \{z_1, z_2, z_3\}$ is the set of design variables, \mathcal{Z} is the design space defined as $\mathcal{Z} = [-10, 10] \times [0, 10] \times [0, 10]$, and $y^* = \{y_1^*, y_2^*\}$ is the solution of the following non-linear system of equations:

$$\begin{cases} y_1 = z_1^2 + z_2 + z_3 - 0.2y_2 \\ y_2 = \sqrt{y_1} + z_1 + z_2 \end{cases} \tag{15}$$

The Sellar problem has a local minimum at $z' \approx \{-1.7171, 0.1384, 0.1128\}$ and a global minimum at $z^* \approx \{1.9776, 0, 0\}$.

To assess the performance of the approach proposed in this work, we first implemented the Sellar problem in the WhatsOpt graphical interface. There, we ticked the options 'EGMDO surrogate' for both disciplinary solvers and provided the initial guess for the coupling variable spaces as $y_1 \in [0, 25]$ and $y_2 \in [0, 25]$. Once the problem was defined, WhatsOpt generated all codes required to solve the optimization problem using both the real disciplinary solvers as well as the disciplinary GPs. We then manually implemented the disciplinary, objective and constraint function computations in the respective python class files. Finally we solved the optimization problem by running the `run_mdo.py` and `run_egmdo.py` files from five randomly generated starting points. The chosen optimizer was the gradient-based Sequential Least Squares Programming (SLSQP) [12] for both approaches and the number of points used to build the initial disciplinary GPs was four. Table 1 shows the results obtained when the real disciplinary solvers were used in the

Inês Cardoso, Sylvain Dubreuil, Nathalie Bartoli, Christian Gogu, Michel Salaün and Rémi Lafage

MDA while Table 2 presents the results obtained when the disciplinary surrogates were used. We note that the same five starting points were used for the two methods.

Table 1: Optima found and number of disciplinary solver calls obtained when solving the MDO using the real disciplinary solvers in an MDF approach. Optimization results obtained from five different starting points.

z^0	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1	n_2
#1	1.9776	0.0	0.0	3.1834	127	127
#2	1.9776	0.0	0.0	3.1834	125	125
#3	-1.7171	0.1384	0.1128	4.1307	188	188
#4	-1.7171	0.1384	0.1128	4.1307	142	142
#5	1.9776	0.0	0.0	3.1834	110	110
Average number of disciplinary solver calls					138.4	138.4

Table 2: Optima found and number of disciplinary solver calls obtained when solving the MDO using the disciplinary surrogates in an MDF approach. Optimization results obtained from five different starting points.

z^0	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1	n_2
#1	1.9776	0.0	0.0	3.1834	15	14
#2	1.9783	0.0	0.0	3.1828	10	10
#3	-1.7089	0.1287	0.1492	4.1223	13	11
#4	-1.7157	0.1407	0.1153	4.1306	13	9
#5	1.9776	0.0	0.0	3.1834	12	10
Number of disciplinary DoE points					4	4
Average number of disciplinary solver calls					16.6	14.8

As is shown, when using the real disciplinary solvers, the optima found depend solely on the starting point. Contrarily, when using the disciplinary surrogates, the optima found vary slightly depending on the quality of the surrogate approximations (for instance, for starting point #2 the solution does not correspond exactly to the global minimum and a small error is committed). Concerning the computational cost, we note that when using the real disciplinary solvers, an average of 138.4 calls were made to each disciplinary solver. On the other hand, when disciplinary GPs were used, an average of $n_1 = 16.6$ calls were made to the first disciplinary solver, while an average of $n_2 = 14.8$ calls were made to the second disciplinary solver.

The results presented in Table 1 were obtained using an MDF approach. Nevertheless, the IDF approach often results in a lower computational cost since the MDA is not solved in every iteration. For this reason, we have implemented the Sellar problem using the IDF approach. The resulting XDSM, obtained using the WhatsOpt graphical interface, is shown in Figure 6. As is shown, when using the IDF approach, consistency constraints must be added to the problem to assure the feasibility of the optimal solution.

The results obtained when solving the MDO using the real disciplinary solvers in an IDF approach are displayed in Table 3. We remark that, while the selected starting

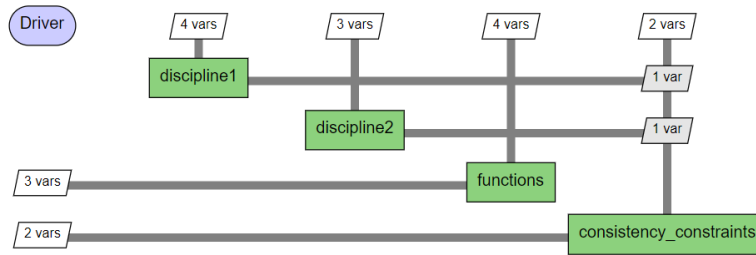


Figure 6: Implementation of the Sellar problem in using the IDF approach. XDSM obtained using the WhatsOpt graphical interface.

points for the design variables are the same as the ones used in Tables 1 and 2, for the IDF approach we must also provide an initial guess for the coupling variable values. For each initial starting point a random value was generated to initialize the coupling variables.

Table 3: Optima found and number of disciplinary solver calls obtained when solving the MDO using the real disciplinary solvers in an IDF approach.

z^0	z_1^*	z_2^*	z_3^*	f_{obj}^*	n_1	n_2
#1	1.9776	0.0	0.0	3.1834	56	56
#2	1.9776	0.0	0.0	3.1834	50	50
#3	-1.5655	0.7879	0.1210	4.3304	61	61
#4	1.9776	0.0	0.0	3.1834	67	67
#5	1.9776	0.0	0.0	3.1834	62	62
Average number of disciplinary solver calls					59.2	59.2

Table 3 shows that, when using the real disciplinary solvers, the IDF approach presents a computational cost that is half of the one obtained when the MDF approach was used. Nevertheless, for starting point #3 the IDF approach did not find the local nor the global optimum. Finally we note that, while the methodology proposed in this work can only be used in an MDF approach, it still requires significantly less disciplinary solver calls than when the real disciplinary solvers are used with either MDF or IDF approaches.

4.2 Wing structural sizing test case

An aircraft wing is a multidisciplinary system, whose coupled behavior is described by the aerodynamic and structural disciplines. Together, these disciplines define the following non-linear system of equations:

$$\begin{cases} u_s = \mathcal{M}_s(\mathbf{x}, f_a) \\ f_a = \mathcal{M}_a(\mathbf{x}, u_s) \end{cases} \quad (16)$$

where $u_s \in \mathcal{C}^s \subset \mathbb{R}^{d^s}$ is the structural displacement of the wing, $f_a \in \mathcal{C}^a \subset \mathbb{R}^{d^a}$ is the vector of aerodynamic forces acting upon the wing, \mathcal{M}_s and \mathcal{M}_a are, respectively, the chosen structural and aerodynamics models and $\mathbf{x} \in \mathcal{X}$ is some set of the design variables which influences the aerodynamic behavior, the structural behavior, or both. In this section,

we will optimize the wing structure in order to minimize the wing mass under structural failure constraints. To that end, we consider the following disciplinary solvers.

To obtain the structural displacement u_s , the wing is modeled using the finite element solver Code Aster [21] and a linear elastic behavior is assumed. To obtain the aerodynamic forces f_a , the non-linear Euler equations are solved using the finite volume solver SU2 [22]. Figure 7 illustrates the structural mesh (Figure 7(a)) and the aerodynamic surface mesh (Figure 7(b)) used to obtain u_s and f_a , respectively. We note that, for the structural mesh, the upper skin was removed to facilitate the visualization of the wing internal structure.

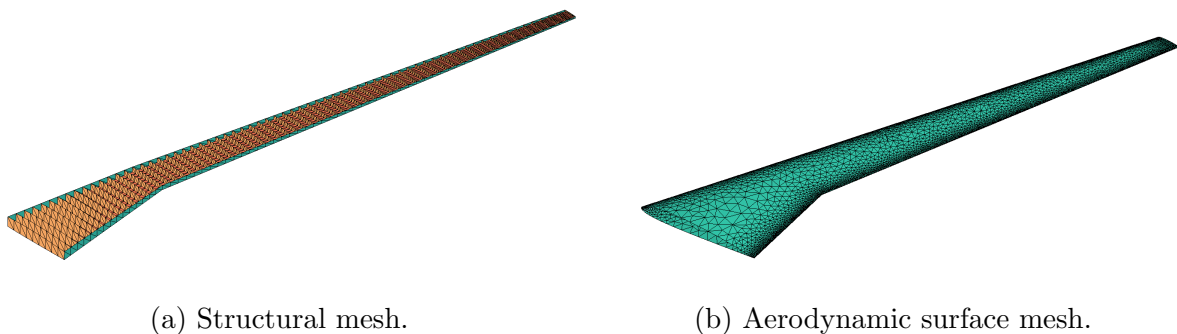


Figure 7: Structural mesh (a) and aerodynamic surface mesh (b) used in the wing structural sizing test case.

As is shown, the wing structure is composed of shell elements used to model the wing's ribs, skins and spars. In terms of geometry, a high aspect ratio wing ($AR = 15$) is defined, presenting a half span of 20.53 meters. The number of spars was set to two, placed at 35% and 60% of the chord and the number of ribs was set to 68. For the flight conditions ground level (0ft) and subsonic flight (Mach number of 0.64) are chosen.

To transfer the loads and displacements between the non-coincident structural and aerodynamic meshes, interpolation across the fluid-surface interface is made. It seeks to ensure the conservation of energy, force and moment, according to:

$$\delta W = \delta u_s^T f_s = \delta u_a^T f_a \quad (17)$$

where δW is the virtual work, δu are the virtual displacements and f are the force vectors. An interpolation matrix $H \in \mathbb{R}^{d^a \times d^s}$ is then defined using Radial Basis Functions (RBF), as proposed in [23]. This matrix allows to write the displacements of the aerodynamic nodes as $u_a = H u_s$ and the structural loading at the nodal points as $f_s = H^T f_a$.

A structural sizing optimization problem is now defined. It seeks to minimize the mass of the wing structure with respect to failure constraints, according to:

$$\begin{aligned} \arg \min_{\mathbf{x} \in \mathcal{X}} \quad & m_w \\ \text{s.t.} \quad & \text{VMIS} \leq \frac{\sigma_y}{2} \end{aligned} \quad (18)$$

where \mathbf{x} is a set of structural design parameters, m_w is the wing mass, VMIS denotes the maximum value of the Von Mises stress and σ_y is the material yield stress, set to 270 MPa. In practice the failure constraint is evaluated for a 2.5g pull-up maneuver condition. As a result, for each set of structural input parameters \mathbf{x} queried by the optimizer, a sub-problem is solved to find the angle of attack for which $L = 2.5W$ is verified, where L

Inês Cardoso, Sylvain Dubreuil, Nathalie Bartoli, Christian Gogu, Michel Salaün and Rémi Lafage

is the lift and W is the aircraft weight. In WhatsOpt this is achieved by defining a single-discipline problem for the structural sizing, as shown in Figure 8(a). The discipline 'trim_mdo' is then defined as a sub-problem where the MDA is included (Figure 8(b)).

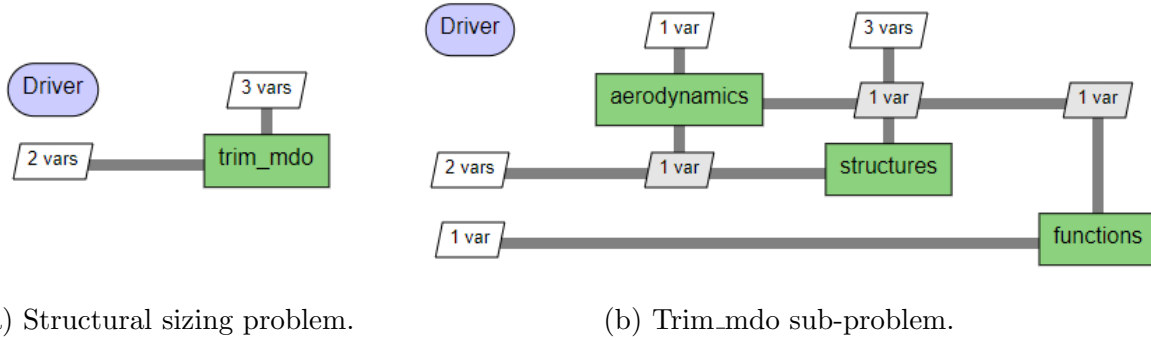


Figure 8: XDSM of the structural sizing problem (a) and trim sub-problem (b) obtained in the WhatsOpt graphical interface.

The structural input parameters are chosen as the thicknesses of the different structural components, i.e., the thickness of the wing ribs (t_r), skins (t_{sk}) and spars (t_{sp}). The range of variation of each input is scaled to take values in $[0, 1]$, however, the unscaled thickness bounds were chosen so that a sufficient variation of the MDA outputs is obtained. Table 4 summarizes the defined optimization problem, whereas Figure 9 illustrates the displacement fields obtained for the $L = 2.5W$ condition when all input parameters are set to either zero or one. As is shown, for the same flight condition, the wing tip displacement is much greater when all input parameters are set to zero, with a maximum observed wing tip displacement of 2.32 meters.

Table 4: Structural sizing problem. Bounds for design variables and constraint functions are provided, where applicable.

	Quantity	Lower bound	Upper bound
Minimize	Wing mass (m_w)	--	--
w.r.t.	Wing rib thickness (t_r)	0	1
	Wing skin thickness (t_{sk})	0	1
	Wing spar thickness (t_{sp})	0	1
s.t.	Von Mises stress (VMIS) $\leq \sigma_y/2$		

The structural sizing problem is first solved using the real disciplinary solvers. For that, the OpenMDAO non-linear block Gauss-Seidel solver (NLBGS) was chosen to solve the MDA and a relative tolerance of 0.01 was selected. For the optimizer the gradient-based SLSQP optimizer is used, with a relative tolerance set to 10^{-3} . Additionally, since gradient information for the real disciplinary solvers is not available, the derivatives are approximated at each iteration using finite differences with a step size of 10^{-4} . The computational cost of the gradient approximation is included in the overall cost of the framework. Finally, to assess the existence of local minima, the optimizer is run from three different starting points. The obtained results are summarized in Table 5.

Table 5 shows that, for all three starting points tested, the optimizer found similar solutions, suggesting that there is only one optimum. Moreover, during the three runs

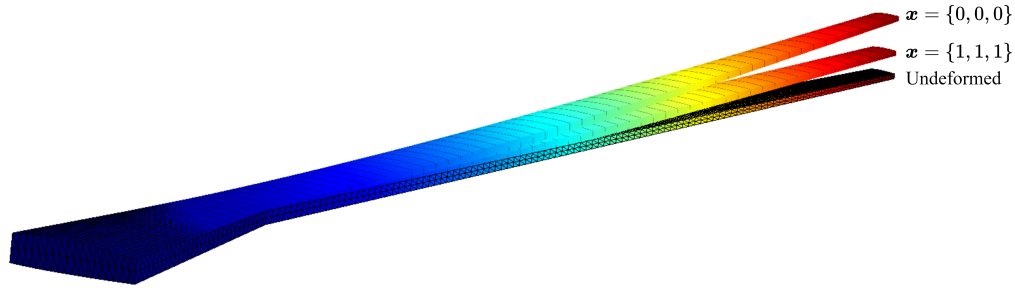


Figure 9: Wing displacement fields obtained for different sets of input parameters. The undeformed wing shape is given as reference.

Table 5: Mass minimization problem. Optima found and number of disciplinary solver calls obtained when solving the MDO using the real disciplinary solvers. Optimization ran from three different starting points.

\mathbf{x}^0	m^0 [kg]	t_r^*	t_{sk}^*	t_{sp}^*	m_w^* [kg]	VMIS* [MPa]	n_{eval_a}	n_{eval_s}
#1	14175.7	0.0	0.1943	0.1988	5167.8	135.03	787	787
#2	14132.7	0.0	0.1950	0.1973	5172.1	134.98	492	492
#3	14360.1	0.0	0.1934	0.2034	5170.4	135.00	590	590
Average number of disciplinary solver calls							623	623

performed, an average of 623 calls were made to each disciplinary solver. This number can be expected to increase if a more complex problem is considered, for instance, by increasing the number of design variables or constraint functions. This justifies the need for an optimization framework that is computationally more efficient.

To solve the problem using the approach described in this work, we first need to reduce the dimension of the coupling variable space. We remark that the dimension of u_s is the number of degrees of freedom obtained for the defined structural model (in this work $d^s = 51048$) while the dimension of f_a is the number of nodes in the wing surface mesh (in this work $d^a = 15697$). Using the POD approximation described in Section 2, these dimensions can be reduced, allowing for the construction of the disciplinary GPs. It is worth noting that, while the POD bases cannot be built using the code generated by WhatsOpt, the DoE used to train the POD approximations can be fed directly to the `gp_factory.py` file to construct the initial disciplinary GPs. Table 6 shows the number of POD coefficients, as well as the number of disciplinary solver calls made during the training of the disciplinary POD bases.

Table 6 shows that, using POD approximation, the aerodynamics discipline may be represented by as little as 10 coefficients while the structural discipline is represented by as little as 3 coefficients. Moreover, we see that, during the training of the POD bases, an average of 36 calls are made to the aerodynamics disciplinary solver while an average of 26 calls are made to the structural disciplinary solver. In the following, the computational cost of this initial training stage will be accounted for in the overall cost of the framework.

As was done before, the problem is solved from different initial starting points. For each starting point, new disciplinary POD bases are generated and the initial disciplinary GPs

Table 6: Number of coefficients in the aerodynamics (n_a) and structural (n_s) POD bases obtained for three different runs. Number of disciplinary solver evaluations (n_{eval_a} and n_{eval_s}) made during the training stage.

	n_a	n_s	n_{eval_a}	n_{eval_s}
Run #1	10	4	36	26
Run #2	11	3	36	26
Run #3	10	3	36	26
\mathbb{E}	10.3	3.3	36	26

are built. The default options of constant mean trend and squared exponential correlation function are used. The random MDA is then solved at each point queried by the optimizer according to the methodology described in Section 2.3. A sample size n_{MC} of 2500 is used and a threshold of 0.03 is set for the CV of both objective and constraint functions. For the optimizer, the SLSQP solver is chosen with the same convergence criteria used for the real disciplinary solvers. The analytical disciplinary gradients are obtained as described in Section 2.4 The obtained results are presented in Table 7.

Table 7: Mass minimization problem. Optima found and number of disciplinary solver calls obtained when solving the MDO using the disciplinary GPs.

\mathbf{x}^0	m^0 [kg]	t_r^*	t_{sk}^*	t_{sp}^*	m_w^* [kg]	VMIS* [MPa]	n_{eval_a}	n_{eval_s}
#1	14174.7	0.0	0.1980	0.1199	4959.66	134.97	46	53
#2	14132.5	0.0	0.1800	0.2102	5013.87	135.04	44	43
#3	14360.1	0.0	0.1593	0.3139	5076.71	135.02	54	51
Average number of disciplinary solver calls							48	49

Table 7 shows that, when using the disciplinary GPs, the optimum value of mass obtained close to the one found using the real disciplinary solvers, with the disciplinary GPs being slightly more optimistic. The maximum relative error is obtained for starting point #1 where the mass obtained using the disciplinary GPs is 4.03% lower than the one given by the real disciplinary solvers. Despite obtaining close objective function values, the optimal sets of design variables found using the disciplinary GPs were not the same as those presented in Table 5, suggesting that either objective or constraint functions are not sufficiently well captured by the random MDA. To test this hypothesis the true MDA was solved at the points found using the disciplinary surrogates. Table 8 compares the results obtained using the true and random MDAs.

Table 8 shows that, as expected, the random MDA commits a small error in the value of the constraint function, leading the algorithm to believe that the points found belong to the feasible region and therefore allowing for more optimistic results. This a consequence of the chosen criterion for the CV, which was set to 0.03. More accurate results could possibly be obtained by choosing a stricter criterion, however, some conducted tests showed that this often leads the algorithm to enrich the disciplinary surrogates several times at the same design space point, resulting in poorly constructed GP approximations.

Finally, we compare the computational cost obtained using the real disciplinary solvers with that obtained using the disciplinary GP approximations. Table 7 shows that, when

Table 8: Mass minimization problem. Optima found and number of disciplinary solver calls obtained when solving the MDO using the disciplinary GPs.

\mathbf{x}^*	m_w [kg]			VMIS [MPa]		
	Real	GPs	Error [%]	Real	GPs	Error [%]
#1	4959.69	4959.66	$\leq 10^{-3}$	137.99	134.97	2.19
#2	5013.88	5013.87	$\leq 10^{-3}$	137.02	135.04	1.44
#3	5076.72	5076.71	$\leq 10^{-3}$	136.97	135.02	1.44

using the disciplinary GPs, an average of 48 calls were made to structural disciplinary solver while an average of 49 calls were made to the aerodynamics solver. This represents a reduction by a factor of 12 compared to the number of disciplinary solver calls made when the real disciplinary solvers were used directly in the MDA (see Table 5).

5 CONCLUSIONS

This work proposes a gradient-based optimization framework that uses disciplinary surrogates to reduce the computational cost of solving the MDA. To that end, the methodology proposed in [6] is used to build and enrich the disciplinary surrogates. The derivatives of the disciplinary surrogates are then provided to the optimizer to perform gradient-based optimization. Thanks to ONERA’s WhatsOpt [11] collaborative environment the proposed methodology is fully implemented within NASA’s OpenMDAO framework [9].

Application to an analytical benchmark test case as well as an engineering test case confirmed the interest of the proposed methodology. For the analytical benchmark test case, both MDF and IDF approaches were tested and it was shown that the proposed methodology required less disciplinary solver calls than either approach. For the structural sizing problem, where the disciplinary solvers are high-fidelity solvers, the obtained results were subject to a small error, leading to slightly optimistic results compared to those obtained using the real disciplinary solvers.

Lastly, the main difficulty of the proposed methodology lies in handling high-dimensional coupling variables. In this work, this challenge was tackled using the DPOD+I strategy proposed in [14], which uses POD approximations to obtain a low dimensional coupling variable space, over which the disciplinary surrogates are built. Nonetheless, it can be expected that for more non-linear disciplinary solvers the number of POD coefficients required to obtain a small projection error may be too large for the combined use with GP interpolation. Future work is thus expected to address this challenge by exploring alternative model order reduction techniques [24, 25].

ACKNOWLEDGEMENTS

This work is a part of the activities of the ONERA-ISAE-ENAC joint research group.

REFERENCES

- [1] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013. doi:[10.2514/1.J051895](https://doi.org/10.2514/1.J051895).
- [2] E. J. Cramer, J. E. Dennis, Jr., P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem

- formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4): 754–776, 1994. doi:[10.1137/0804044](https://doi.org/10.1137/0804044).
- [3] M. Zhang, W. Gou, L. Li, F. Yang, and Z. Yue. Multidisciplinary design and multi-objective optimization on guide fins of twin-web disk using kriging surrogate model. *Structural and Multidisciplinary Optimization*, 55:361–373, 2017. doi:[10.1007/s00158-016-1488-0](https://doi.org/10.1007/s00158-016-1488-0).
- [4] X. Wang, M. Li, Y. Liu, W. Sun, and X. Song. Surrogate based multidisciplinary design optimization of lithium-ion battery thermal management system in electric vehicles. *Structural and Multidisciplinary Optimization*, 56:1555–1570, 2017. doi:[10.1007/s00158-017-1733-1](https://doi.org/10.1007/s00158-017-1733-1).
- [5] W. Scholten and D. Hartl. Uncoupled method for static aeroelastic analysis. *Journal of Fluids and Structures*, 101:103221, 2021. doi:[10.1016/j.jfluidstructs.2021.103221](https://doi.org/10.1016/j.jfluidstructs.2021.103221).
- [6] S. Dubreuil, N. Bartoli, T. Lefebvre, and C. Gogu. Towards an efficient global multidisciplinary design optimization algorithm. *Structural and Multidisciplinary Optimization*, page 1739–1765, 2020. doi:[10.1007/s00158-020-02514-6](https://doi.org/10.1007/s00158-020-02514-6).
- [7] J. Martins, J. Alonso, and J. Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6, 2005. doi:[10.1023/B:OPTE.0000048536.47956.62](https://doi.org/10.1023/B:OPTE.0000048536.47956.62).
- [8] J. P. Jasa, J. T. Hwang, and J. R. R. A. Martins. Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4): 1815–1827, Apr. 2018. doi:[10.1007/s00158-018-1912-8](https://doi.org/10.1007/s00158-018-1912-8).
- [9] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization. *Structural and Multidisciplinary Optimization*, 59:1075–1104, 2019. doi:[10.1007/s00158-019-02211-z](https://doi.org/10.1007/s00158-019-02211-z).
- [10] N. P. Bons and J. R. R. A. Martins. Aerostructural design exploration of a wing in transonic flow. *Aerospace*, 7(8), 2020. doi:[10.3390/aerospace7080118](https://doi.org/10.3390/aerospace7080118).
- [11] R. Lafage, S. Defoort, and T. Lefebvre. Whatsopt: a web application for multidisciplinary design analysis and optimization. In *AIAA Aviation 2019 Forum*, page 2990, 2019. doi:[10.2514/6.2019-2990](https://doi.org/10.2514/6.2019-2990).
- [12] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Tech. Rep. DFVLR-FB 88-28. DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, 1988.
- [13] M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands, 1994. doi:[10.1007/978-94-015-8330-5_4](https://doi.org/10.1007/978-94-015-8330-5_4).

- [14] G. Berthelin, S. Dubreuil, M. Salaün, N. Bartoli, and C. Gogu. Disciplinary proper orthogonal decomposition and interpolation for the resolution of parameterized multidisciplinary analysis. *International Journal for Numerical Methods in Engineering*, 123(15):3594–3626, 2022. doi:[10.1002/nme.6981](https://doi.org/10.1002/nme.6981).
- [15] I. Sobol’. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4): 86–112, 1967. doi:[10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9).
- [16] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008. doi:[10.1016/j.ress.2007.04.002](https://doi.org/10.1016/j.ress.2007.04.002).
- [17] A. B. Lambe and J. R. R. A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, 2012. doi:[10.1007/s00158-012-0763-y](https://doi.org/10.1007/s00158-012-0763-y).
- [18] M. McKay, R. Beckman, and W. Conover. A comparison of three methods for selecting vales of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979. doi:[10.1080/00401706.1979.10489755](https://doi.org/10.1080/00401706.1979.10489755).
- [19] M. A. Bouhleb, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins. A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, page 102662, 2019. ISSN 0965-9978. doi:[10.1016/j.advengsoft.2019.03.005](https://doi.org/10.1016/j.advengsoft.2019.03.005).
- [20] R. Sellar, S. Batill, and J. Renaud. *Response surface based, concurrent subspace optimization for multidisciplinary system design*. 34th Aerospace Sciences Meeting and Exhibit, 1996. doi:[10.2514/6.1996-714](https://doi.org/10.2514/6.1996-714).
- [21] EDF. Finite element *code_aster*, analysis of structures and thermomechanics for studies and research. Open source on www.code-aster.org, 1989–2017.
- [22] T. Economon, F. Palacios, S. Copeland, T. Lukaczyk, and J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54:1–19, 2015. doi:[10.2514/1.J053813](https://doi.org/10.2514/1.J053813).
- [23] T. C. S. Rendall and C. B. Allen. Unified fluid-structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, 2008. doi:[10.1002/nme.2219](https://doi.org/10.1002/nme.2219).
- [24] G. Boncoraglio and C. Farhat. Piecewise-global nonlinear model order reduction for pde-constrained optimization in high-dimensional parameter spaces. *SIAM Journal on Scientific Computing*, 44(4):A2176–A2203, 2022. doi:[10.1137/21M1435343](https://doi.org/10.1137/21M1435343).
- [25] J. Barnett and C. Farhat. Quadratic approximation manifold for mitigating the kolmogorov barrier in nonlinear projection-based model order reduction. *Journal of Computational Physics*, 464:111348, 2022. doi:[10.1016/j.jcp.2022.111348](https://doi.org/10.1016/j.jcp.2022.111348).



AEROELASTIC ANALYSIS OF HIGH ASPECT RATIO AND STRUT-BRACED WINGS

Yoann Le Lamer^{1*}, Joseph Morlier², Emmanuel Benard¹ and Ping He³

1: ISAE-SUPAERO, Université de Toulouse
10 Avenue Édouard Belin, 31400, Toulouse, France
yoann.le-lamer,emmanuel.benard@isae-supero.fr, <http://www.isae-supero.fr>

2: ISAE-SUPAERO, Université de Toulouse, ICA, MINES ALBI, UPS, INSA, CNRS
10 Avenue Édouard Belin, 31400, Toulouse, France
joseph.morlier@isae-supero.fr, <http://www.isae-supero.fr>

3: Department of Aerospace Engineering, Iowa State University
537 Bissell Road, Ames, IA 50011, United States of America
phe@iastate.edu, <http://www.aere.iastate.edu>

Abstract. *High Aspect Ratio (HAR) and Strut-Braced Wings (SBW) configurations represent promising avenues of research for meeting the challenge of reducing aviation carbon emissions. Both high and low fidelity models have already been developed for the above-mentioned configurations as well as for the NASA Common Research Model (CRM) that will serve as a baseline for performance comparisons. These aero-structural models are then implemented within an in-house aeroelastic framework for analysis and optimization. High-fidelity models correspond to a RANS aerodynamic analysis associated to a 3D wingbox finite-elements (FE) analysis. Conversely, low-fidelity models resort to a VLM aerodynamic analysis associated to a 1D beam model FE analysis. The objective of this work is to use the so developed models to perform multidisciplinary optimization in order to assess performance improvements potential at preliminary design stage. Different levels of fidelity were developed with a multifidelity approach in mind for further developments. The use of a multifidelity approach makes it possible to reduce computational costs by mainly resorting to low-fidelity computations and only running high-fidelity computations when necessary. Then, aeroelastic optimization is carried out on a modified version of the CRM wing with a higher aspect ratio to study aerodynamic gains. Afterwards, a SBW configuration, here the PADRI geometry, is examined to evaluate its mass reduction potential in addition to the drag reduction provided by the increased aspect ratio. This work currently focuses on aeroelastic optimization using a single fidelity, but is to be further extended to a multifidelity approach.*

Keywords: High Aspect Ratio (HAR) wings, Strut-Braced wings (SBW), Aeroelasticity, Multidisciplinary Design Analysis and Optimization (MDAO)

1 INTRODUCTION

Reducing aviation carbon emissions could potentially be achieved through the use of High Aspect Ratio (HAR) [1] and Strut-Braced Wings (SBW) configurations. Multidisciplinary design optimization (MDO), and in particular aerostructural optimization is necessary in order to make the most of these configurations [2]. Preliminary design studies need to be performed in order to evaluate the performance of a large number of aircraft configurations. Specific models has to be developed to this purpose to mitigate initially high computational costs [3].

The issue of computational costs [3] is critical when it comes to aerostructural studies. This can be dealt with by resorting to surrogate models to approximate a system [4], be it for CFD or finite-elements analysis (FEA), using a certain number of samples.

Multifidelity can also be a promising strategy to reduce computational costs. Indeed, when choosing such a method, surrogate models are trained using models with different levels of fidelity, and thus accuracy and computational costs. An algorithm helps find a balance between computation time and physical accuracy.

In this work we use an aeroelastic analysis and optimization framework developed by [5], the aerostructure package [6], which is implemented within the OpenMDAO framework [7].

The final goal of this study is to use aerostructural models of both high and low fidelity to apply a multifidelity approach in order to assess the performance of several configurations of interest in an efficient way. This paper will present the use of high and low fidelity aeroelastic models for the optimization of different configurations. High and low fidelity aeroelastic optimizations are then performed and results compared.

2 TARGETED CONFIGURATIONS

The objective of this work is to study different configurations to assess potential performance improvements through optimization.

The first goal is to study the effects of increasing wing aspect ratio. To this purpose we chose to focus on the uCRM-13.5 model developed in [8]. This model corresponds to a modified version of the NASA CRM model, with an aspect ratio increased by 50%. Main geometrical dimensions are summarized in table 1. Also, uCRM-9.0 geometry will be used as a baseline in order to assess the benefits of increasing the aspect ratio, particularly in term of induced drag reduction. This model corresponds to the undeformed version of NASA Common Research Model (CRM) [9]. The so developed models have been validated using data provided in [8], for both uCRM-9.0 and uCRM-13.5 (see previous paper). Figure 1 presents the geometry of the NASA CRM wing.

The second goal is to study the effects of adding a strut to a high aspect ratio wing, particularly in terms of potential mass reduction. To this purpose we are focusing on the PADRI configuration. Main geometrical dimensions are summarized in table 1. The strut-wing junction is positioned at 42% of half-span and at 50% of the corresponding chord. Figure 2 presents the geometry of the PADRI configuration.

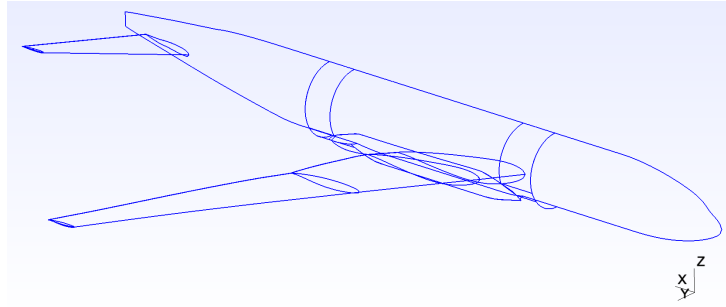


Figure 1: Geometry of the NASA Common Research Model (CRM).

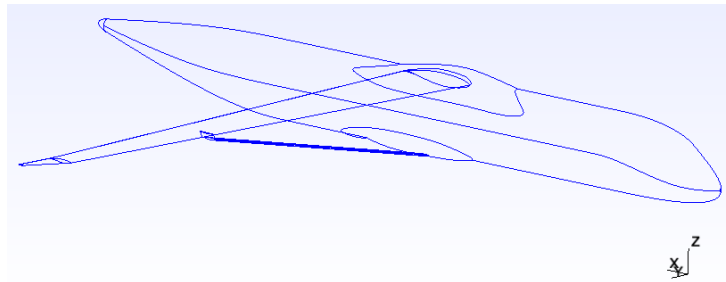


Figure 2: Geometry of the PADRI configuration (Platform for Aircraft Drag Reduction Innovation).

Table 1: Main geometrical parameters of targeted configurations.

Name	uCRM-9.0	uCRM-13.5	PADRI
Planform area [m^2]	383.74	384.05	161.00
Span [m]	58.76	72.00	55.6
MAC [m]	7.01	5.36	3.264
Aspect ratio	9.0	13.5	19.2
Taper ratio	0.275	0.250	0.256
1/4 Chord sweep [deg]	35.0	35.0	15.0
Dihedral [deg]	0.0	0.0	-4.0
Cruise Mach	0.85	0.85	0.72

3 METHODOLOGY

3.1 Aerostructural framework implementation

The aeroelastic analysis and optimization framework used in this study, developed by [5], is written in Python language and makes use of the OpenMDAO platform that is well suited for MDAO problems. This framework is able to solve both static and dynamic aeroelasticity problems. The way it works is presented in figure 3, each block (green boxes) corresponds to a specific component. The other boxes are used for linking the components together, feeding inputs from different outputs. Here two main disciplines run within the framework. First, the aerodynamic problem is solved either with DAFOAM or AVL depending on the chosen level of fidelity. Then, solving the structural problem is done by running MSC Natran finite-element analysis (FEA), either for a 1D beam model or a 3D wingbox model depending on the desired level of fidelity.

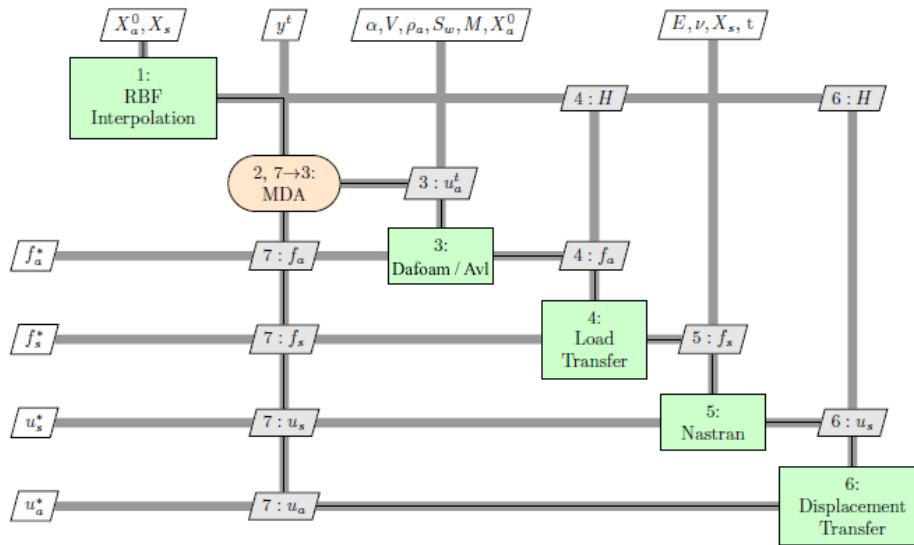


Figure 3: XDSM diagram of the static aeroelastic OpenMDAO HF/LF solver.

A mesh interpolation scheme that depends on the selected models is used for transferring loads from the aerodynamic solver to the structural mesh, and displacements from the structural solver to the aerodynamic mesh. In this work, low-fidelity computations are performed using both LF structural and aerodynamic models. Likewise, HF computations are carried out with both HF models.

The fluid-structure interaction problem is solved using an iterative partitioned approach. This process runs iteratively until convergence is achieved, based on selected quantities of interest.

3.2 Model description and generation workflow

For both high and low fidelity (HF and LF) simulations, only a half wing is considered, because a symmetry condition is applied in order to reduce computation time.

3.2.1 High-fidelity (HF) computations

In order to perform HF aerodynamic computations, the DAFOAM [10] solver is used to run CFD. For now, this aerodynamic code is used to solve steady Navier-Stokes equations. Airflow is considered viscous, turbulent and compressible. For generating the CFD mesh, the Pointwise meshing software is used based on the wing geometry in IGS format. The workflow to be followed in order to obtain HF aerodynamic loads is illustrated in figure 4.

The objective being to perform aerostructural analysis, the previously described aerodynamic solver needs to be coupled with a finite-elements analysis (FEA) solver, here the MSC Nastran software [11] is used. A three-dimensional wingbox finite-elements model (FEM) is generated as a BDF file. This wingbox shell-model breaks down into several components, upper and lower skin, ribs, spars and stiffeners. Skin, ribs and spars are represented using shell elements (CQUAD4), and stiffeners are represented using beam elements (CBAR). A Python module has been specifically developed to generate this model based on wing geometrical and material properties inputs. An example of a so

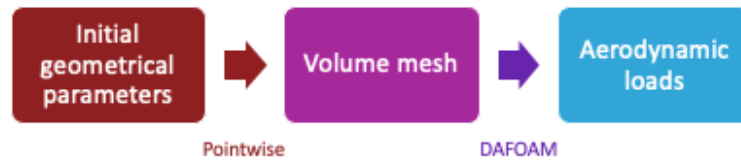


Figure 4: HF aerodynamic computational workflow.

developed FE mesh for the CRM wing is presented in figure 5.

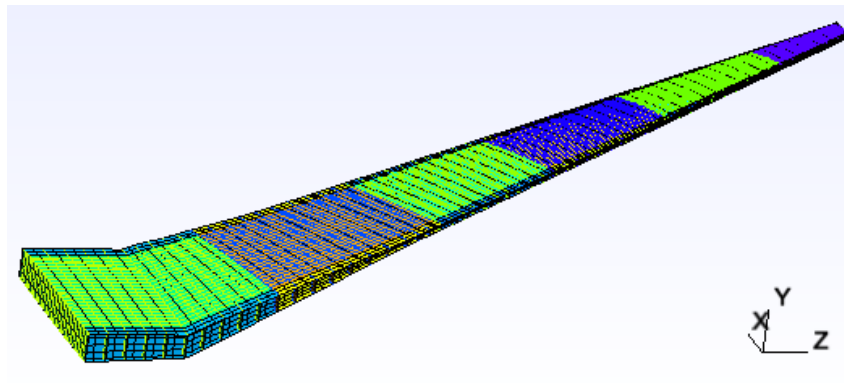


Figure 5: CRM structural FE model generated using dedicated module.

The wingbox structural models used in this work for uCRM-9.0 and uCRM-13.5 have been validated by comparing with results from [8]. Interpolation between aerodynamic and structural meshes is used to transfer loads and displacements. This is performed thanks to radial basis functions (RBF), using Thin Plate Splines (TPS) approximation developed in [12].

For now only linear FEA has been considered and tested, but the possibility to easily switch from a linear to a non-linear structural solver is also implemented for later use.

3.2.2 Low-fidelity (LF) computations

In order to perform LF aerodynamic computations, the AVL code [13] is used to carry out VLM (Vortex Lattice Method) computations, the corresponding model is generated through a dedicated Python module based on wing geometrical properties inputs. The model is defined through a certain number of control points, and associated chords, and twists. The workflow to be followed in order to obtain LF aerodynamic loads is presented in figure 6.

This VLM solver is also coupled to MSC Nastran [11]. Contrary to HF computations, a one-dimensional finite-elements model (FEM) is generated as a BDF file. The wing is split into a certain number of beam elements (CBAR). Another specific Python module has been developed to generate this model based on the wing geometrical and material properties.

The beam elements used in this model correspond to a rectangular beam section. Geometrical parameters, namely height, length, skin and spar thicknesses are the parameters,



Figure 6: LF aerodynamic computational workflow.

are used as inputs, and corresponding stiffness properties are computed thanks to the script.

3.3 Optimizer and design variables

In this study, given the size of the problem, we have chosen to use a gradient-free optimizer. COBYLA optimizer is used for the rest of this work for both HF and LF optimizations. Relative tolerance of the optimizer is set to 10^{-3} .

The design variables considered in this study are the following: Structural thicknesses controlled using B-splines (4 control points), Aspect ratio and Wing twist controlled using B-splines (4 control points).

B-splines are used here in order to reduce the number of design variables. These are curves interpolated between a given number of control points which can then be used as design variables.

Initial values of the geometric design variables are set to the values of initial wing geometry. As for structural thicknesses, there are set to be decreasing linearly from 20mm to 2mm in the spanwise direction.

We describe here how modifications were applied to the CFD model, more specifically to the CFD mesh. The initial CFD mesh is deformed using FFD boxes rather than re-generated to avoid this time-consuming step. A FFD box is a box of control points that embeds the aircraft surface mesh. Any changes in FFD shape are then mapped onto the surface mesh.

Modifications of other models are not described as they are simply generated again as the process is fast and easy to automate.

4 APPLICATION CASES

The material considered in this study for structural models components is Aluminum alloy 7000 series. Its mechanical properties are presented in 2. Structural elements are sized to ensure that their maximum Von Mises stress is inferior to the material yield stress 2 with a safety factor of 1.5, corresponding to 280 MPa. Computations are considering both 1g and 2.5g load cases. The 2.5g extreme load case is the one being used for structural sizing. To be more specific, the 2.5g load case corresponds to the cruise condition where lift balances 2.5 times aircraft mass.

Our previous study [14] established that models developed are consistent with reference

Table 2: Material properties.

Parameter	Value
Density [kg/m ³]	2780
Young's Modulus [MPa]	73100
Poisson Ratio	0.33
Yield Stress [MPa]	420

studies for uCRM-9.0 and uCRM-13.5 (see [8]).

4.1 Application to a HAR configuration

In this subsection, the optimization method described in section 3 is applied to the uCRM-13.5 high aspect ratio configuration to evaluate wing mass and examine how induced drag is affected when increasing aspect ratio. The optimization problem is described in table 3. Results are then compared to reference data for uCRM-9.0 from [14]. Structural and aerodynamic models are generated based on model chord and twist distributions, and wingpress geometry developed and presented in [8]. Flight conditions for each load case are described in table 4.

Table 3: Optimization problem applied to both application configurations (see subsection 3.3).

	Function/Variable	Bound	Comment
minimize	$\frac{CD_i}{CD_{i_{ref}}} + \frac{mass}{mass_{ref}}$		with $CD_{i_{ref}} = 0.0250$ and $mass_{ref} = 15000$.
with respect to	Aspect ratio	≤ 28.0	
	Wing twist [deg.]	$\geq -10.0, \leq 10.0$	B-spline parameters
	Skin thickness [mm]	≥ 2.0	B-spline parameters
	Spar thickness [mm]	≥ 2.0	B-spline parameters
subject to	Maximum stress at 2.5g	≤ 280.0 MPa	Von Mises stress
	$CL_{1g} = 0.55$		
	$CL_{2.5g} = 0.55*2.5$		

Table 4: Flight conditions for each load case for MDO on uCRM-13.5 geometry.

Load case	1 g	2.5 g
CL (half-wing)	0.55	0.55*2.5
Mach number	0.85	0.85
Flight altitude [ft]	37,000	37,000

Results of HF and LF optimizations are summarized in table 5. As expected HF mass evaluation shows a 42.4% increase with regard to uCRM-9.0 reference case. Also, we observe a greater flexibility of the HAR configuration as wingtip displacement is 90.2%

Table 5: Results of HF and LF aeroelastic MDO for uCRM-13.5 geometry (uCRM-9.0 reference data from [8]).

Computation	High fidelity	Low fidelity
1g AoA [deg.]	2.49	1.68
Estimated cruise CDi [drag counts]	76.0	89.0
Baseline CDi (uCRM-9.0) [drag counts]	88.0	97.0
Deviation wrt baseline [%]	-13.6	-8.2
Optimized aspect ratio	14.9	15.5
Cruise wingtip displacement [m]	5.02	4.93
Baseline wingtip displacement (uCRM-9.0) [m]	2.64	2.85
Deviation wrt baseline [%]	90.2	73.0
Mass [kg]	15,952	13,865
Baseline mass (uCRM-9.0) [kg]	11,201	9,433
Deviation wrt baseline [%]	42.4	47.0
Number of iterations	11	6
Computation time [s]	17,237	1,548

higher than the one of uCRM-9.0. Finally, it appears that induced drag is reduced by 13.6% for the uCRM-13.5 configuration compared to the uCRM-9.0 reference configuration. Similar trends are observed when performing LF optimization. HF results shows a wing mass higher than LF results. It is to be noticed that LF optimization runs approximately 11 times faster than HF optimization.

4.2 Application to a HAR SBW configuration

Finally, we will evaluate mass reduction that can be obtained when adding a strut to a HAR configuration, here the PADRI configuration. The optimization problem is described in table 3. For comparison purposes, we have designed a reference configuration where we have removed the strut from the original PADRI configuration. This geometry will be referred as PADRI-WS in the rest of the paper. The structure of the strut considered in this paper has a rectangular shape, with two parameters for each section, namely height (length is twice the height) and panel thickness. Flight conditions for each load case are described in 6.

Table 6: Flight conditions for each load case for MDO on PADRI SBW geometry.

Load case	1 g	2.5 g
CL (half-wing)	0.43	0.43*2.5
Mach number	0.72	0.72
Flight altitude [ft]	30,000	30,000

Results of HF and LF optimizations are summarized in table 7. Wingtip displacement is reduced by 32.9% with the HF computations. That is consistent with the strut reducing wing structure flexibility. It can also be observed that the HF mass is significantly reduced with regard to the PADRI-WS configuration. Indeed, the mass reduction is of 21.2%. This

Table 7: Results of HF and LF aeroelastic MDO for PADRI SBW geometry (PADRI-WS reference data).

Computation	High fidelity	Low fidelity
1g AoA [deg.]	1.32	0.94
Estimated cruise CDi [drag counts]	92.0	101.0
Baseline CDi (PADRI-WS) [drag counts]	89.0	97.0
Deviation wrt baseline [%]	3.4	4.1
Optimized aspect ratio	22.2	23.1
Cruise wingtip displacement [m]	1.06	1.43
Baseline wingtip displacement (PADRI-WS) [m]	1.58	1.95
Deviation wrt baseline [%]	-32.9	-26.7
Mass [kg]	9,878	9,102
Baseline mass (PADRI-WS) [kg]	12,541	11,982
Deviation wrt baseline [%]	-21.2	-24.0
Number of iterations	14	9
Computation time [s]	22,428	1,215

confirms the mass reduction potential of SBW configurations. Finally, induced drag is similar when compared to the PADRI-WS baseline configuration. Indeed, the deviation is below 5%. This is expected as the contribution of the strut to total lift is not significant. Similar trends are observed when performing LF optimization. HF results shows a wing mass higher than LF results. It is to be noticed that LF optimization runs approximately 18 times faster than HF optimization.

5 RESULTS

The comparisons with baseline values for both application cases are presented in figures 7, 8 and 9. Also, we have confirmed the expected result that induced drag is reduced when increasing the aspect ratio as displayed in figure 8. Finally, the performed optimizations have exemplified the mass reduction potential of a SBW configuration.

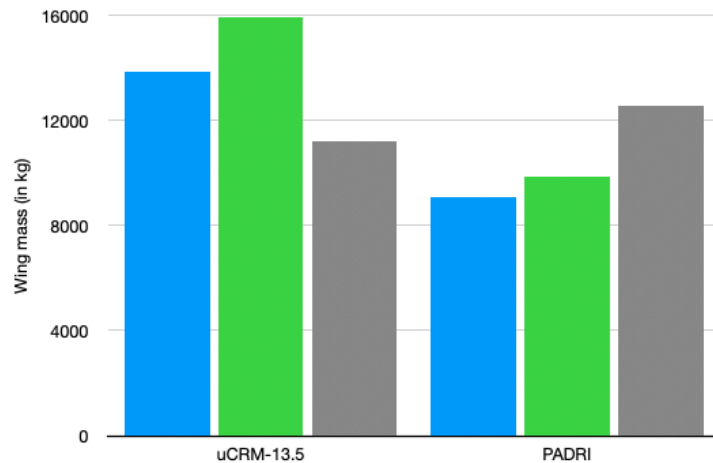


Figure 7: Synthesis of wing mass values for each configuration of interest (HF in green, LF in blue, HF baseline in grey).

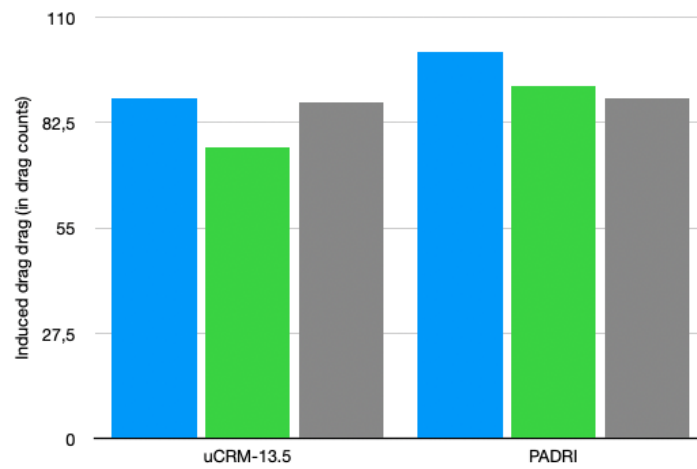


Figure 8: Synthesis of induced drag values for each configuration of interest (HF in green, LF in blue, HF baseline in grey).

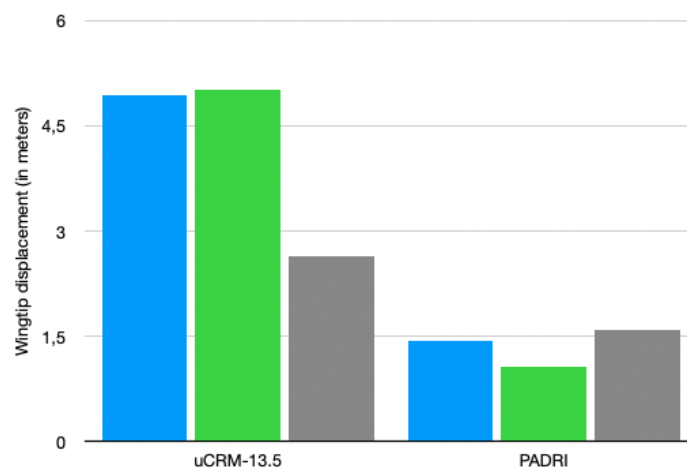


Figure 9: Synthesis of wingtip displacement values for each configuration of interest (HF in green, LF in blue, HF baseline in grey).

6 CONCLUSIONS

This work has demonstrated the aeroelastic optimization capabilities of our in-house framework applied to the HAR version of the NASA CRM, the uCRM-13.5 configuration as well as to the strut-braced PADRI configuration. This approach is effective for the two levels of fidelity mentioned in this paper.

In order to be able to evaluate the interest of a large number of configurations at preliminary design stage, it is necessary to make computation time more affordable. To this purpose, we are planning to use a multifidelity optimization approach that would allow us to mitigate computation costs. This is deemed to be a promising approach as the presented optimizations show that LF simulations run more than 10 times faster than HF simulations. The first step could be to use a strategy close to the one we described in [15], and later we would use MFEGO methodology as presented in [16].

ACKNOWLEDGMENT

The work presented in this paper has been funded by the European Union in H2020-EU.3.4.5.10 through the CleanSky2 project U-HARWARD, under grant Grant agreement ID: 886552

REFERENCES

- [1] T. Brooks, G. Kennedy, and J. Martins. High-fidelity multipoint aerostructural optimization of a high aspect ratio tow-steered composite wing. 01 2017. doi:10.2514/6.2017-1350.
- [2] G. K. W. Kenway and J. R. R. A. Martins. Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration. *Journal of Aircraft*, 51(1):144–160, Jan. 2014. doi:10.2514/1.C032150. URL <https://arc-aiaa-org.rev-doc.isae.fr/doi/10.2514/1.C032150>. Publisher: American Institute of Aeronautics and Astronautics.
- [3] M. C. Kennedy and A. O’Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000. ISSN 00063444.
- [4] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, Dec 2001.
- [5] J. Mas-Colomer. *Aeroelastic Similarity of a Flight Demonstrator via Multidisciplinary Optimization*. PhD thesis, 12 2018.
- [6] J. M. Colomer. Aerostructure package, 2019. URL <https://github.com/mid2SUPAERO/aerostructures>.
- [7] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. Openmdao: an open-source framework for multidisciplinary design, analysis, and optimization. 2019.
- [8] T. R. Brooks, G. K. Kenway, and J. R. R. A. Martins. Undelected Common Research Model (uCRM): An Aerostructural Model for the Study of High Aspect Ratio

- Transport Aircraft Wings. In *35th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2017-4456.
- [9] NASA. Nasa common research model, 2017. URL <https://commonresearchmodel.larc.nasa.gov/>.
- [10] P. He, C. Mader, J. Martins, and K. Maki. Dafoam: An open-source adjoint framework for multidisciplinary design optimization with openfoam. *AIAA Journal*, page 2019, 10 2019. doi:10.2514/1.j058853.
- [11] R. S. Lahey, M. P. Miller, and M. Reymond. Msc/nastran reference manual, version 68. *The MacNeal-Schwendler Corporation*, 1994.
- [12] T. C. S. Rendall and C. B. Allen. Unified fluid–structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, 74(10):1519–1559, 2008. doi:<https://doi.org/10.1002/nme.2219>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2219>.
- [13] M. Drela and H. Youngren. Avl overview, 2013. URL <http://web.mit.edu/drela/Public/web/avl/>.
- [14] E. B. Yoann Le Lamer, Joseph Morlier and P. He. Aeroelastic analysis of high aspect ratio and strut-braced wings. In *ICAS 2022*, Stockholm, Sweden, 2022. URL https://www.icas.org/ICAS_ARCHIVE/ICAS2022/data/papers/ICAS2022_0560_paper.pdf.
- [15] Y. L. Lamer, G. Quaglia, E. Bénard, and J. Morlier. Multifidelity aeroelastic optimization applied to har wing. In *Aerobest 2021*, pages 454–472, Virtual event, PT, 2021. URL <https://oatao.univ-toulouse.fr/28740/>.
- [16] N. Bartoli, M. Meliani, J. Morlier, T. Lefebvre, M.-A. Bouhleb, and J. Martins. *Multifidelity efficient global optimization: Methodology and application to airfoil shape design*. 2019. doi:10.2514/6.2019-3236. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2019-3236>.



AN UNCERTAINTY QUANTIFICATION METHOD BASED ON PROPER ORTHOGONAL DECOMPOSITION AND POLYNOMIAL CHAOS EXPANSION

Luca Battaglia¹, Federico Carlini¹, Alberto Clarich¹ and Rosario Russo¹

1: ESTECO Spa
Padriciano 99, 34149 Trieste, Italy
e-mail: info@esteco.com, www.esteco.com

Abstract. *Many industrial processes and product performance are affected by uncertainties that can arise from different factors, such as dimensional tolerances, manufacturing errors, and fluctuations in operating conditions. In engineering design problems, the uncertainties of the design parameters are transferred to the system responses. In these cases it would be better to describe both the parameters and the responses by statistical distributions rather than single deterministic values. To deal with problems affected by several uncertainties and expensive simulation time, e.g. CFD analyses, it is particularly important to develop methodologies that are at the same time accurate and that rely on a limited number of sample evaluations.*

In this paper, we propose an Uncertainty Quantification (UQ) method based on non-intrusive Snapshot Proper Orthogonal Decomposition (POD) Reduced Order Model (ROM) and Polynomial Chaos Expansion (PCE) to efficiently compute the uncertainty propagation on a vectorial field of interest due to uncertain input parameters. The method is applied to the CFD problem of the flow over an airfoil with parameterized uncertain angle of attack and inflow velocity to estimate the uncertainties of the pressure field in each mesh cell.

Keywords: Industrial Design, Proper Orthogonal Decomposition, Reduced Order Model, Uncertainty Quantification, Polynomial Chaos Expansion

1 INTRODUCTION

Most of the engineering problems are affected by uncertainties due to various factors such as the design specifications and the operating conditions of the system. Uncertainties in the input parameters lead to uncertainties in the responses or output variables, and thus to uncontrolled variability in the performance of the system. Considering the input variables as random variables that vary with a certain statistical distribution, the ideal goal is to determine the resulting distribution of the output variables.

Uncertainty Quantification (UQ) is the process of assessing and characterizing the statistical properties of the output parameters of a system due to the stochasticity of its input parameters. Although UQ can be used to determine the probability distribution function (PDF) of the output variables, in industrial cases it is often sufficient to estimate the first and second statistical moments, i.e., the mean and standard deviation values. Polynomial Chaos Expansion (PCE) [1] is one of the most efficient and accurate techniques to estimate these two quantities.

To deal with the uncertainties in complex systems with many input and output variables, even if an efficient UQ method is used, the required computational resources can be prohibitive due to the high cost in terms of model simulation time. Model Order Reduction (MOR) is a technique for reducing the computational complexity of mathematical models in numerical simulation by reducing the degrees of freedom associated with the model. The resulting model, commonly called Reduced Order Model (ROM), is an approximation of the original model. There are many approaches to obtain a ROM such as the Snapshot Proper Orthogonal Decomposition (POD) method [2, 3], on which this paper is based.

UQ techniques can be applied to many possible scenarios and applications. Without loss of generality, this paper is focused on the use case problem of the flow over a NACA 0012 airfoil with uncertain angle of attack and inflow velocity. Section 2 presents the technical details of the selected use case and the involved tools. Section 3 introduces the theoretical background of the PCE and the Snapshot POD, and section 4 describes how they can be combined to develop different UQ methods. Finally, section 5 compares the results obtained by applying the various UQ techniques to the test case itself.

2 USE CASE AND FRAMEWORK

The reference case chosen for this paper is the numerical simulation of an airfoil with a variable angle of attack and inflow velocity. The CFD model, implemented in OpenFOAM, is based on the *2DN00: 2D NACA 0012 Airfoil Validation Case* from the NASA Langley Turbulence Modeling Resource [4].

The numerical domain, shown in Fig.1 and Fig.2, is characterized by the following properties:

- Chord length of the airfoil: $c = 1.0 [m]$
- Domain longitudinal direction: x
- Domain vertical direction: y
- Domain spanwise direction: z
- Domain dimensions: $(x, y, z) = (985.5, 1015.6, 1.0) [m]$

- Mesh type: hexahedral cells
- Mesh cells number: $m = 230529$, $(m_x, m_y, m_z) = (257, 897, 1)$

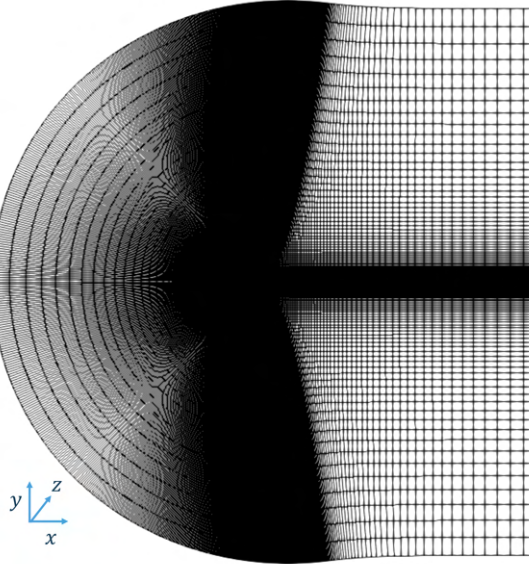


Figure 1: OpenFOAM mesh

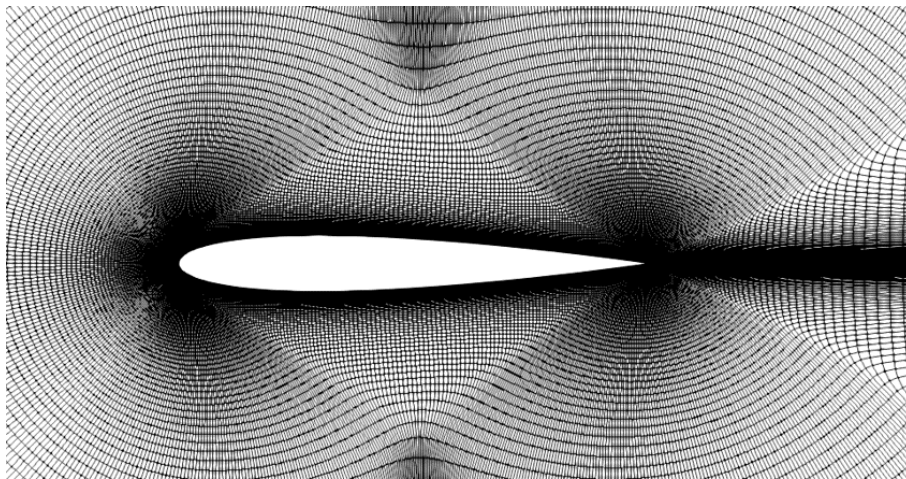


Figure 2: OpenFOAM mesh details

The physical model is characterized by:

- Angle of attack nominal value: $\alpha = 4 [deg]$
- Mach number nominal value: $M = 0.6$
- Kinematic viscosity of fluid: $\nu = 1.50833 \times 10^{-5} [m^2s^{-1}]$
- Speed of sound: $= 340.29 [ms^{-1}]$
- Streamwise far-field flow speed: $U_{inf} = M U_s = 204.174 [ms^{-1}]$

- Longitudinal direction far-field flow speed: $U_x = U_{inf} \cos(\alpha) = 203.67 [ms^{-1}]$
- $U_y = U_{inf} \sin(\alpha) = 14.24 [ms^{-1}]$
- Reynolds number based on chord length: $Re = U_x c \nu^{-1} \simeq 13.5 \times 10^6 [ms^{-1}]$
- Turbulence model: Spalart-Allmaras

The design variables α and M are statistically distributed random variables and the pressures in each j -th mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ are the uncertain dependent variables. The goal is to estimate the mean μ_{p_j} and standard deviation σ_{p_j} to evaluate the mean and standard deviation pressure fields

$$\boldsymbol{\mu}_p = [\mu_{p_1} \quad \dots \quad \mu_{p_m}] \quad (1)$$

$$\boldsymbol{\sigma}_p = [\sigma_{p_1} \quad \dots \quad \sigma_{p_m}] \quad (2)$$

The fields can be visualized, as in Fig.3, to identify the model regions where the effect of the design variables uncertainties is higher.

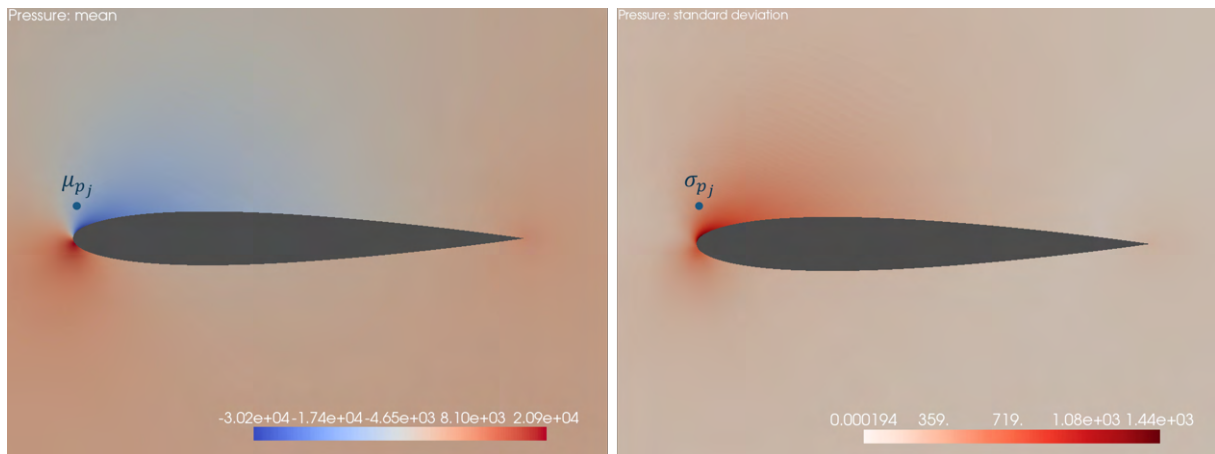


Figure 3: Mean and standard deviation pressure fields

Let's consider α and M distributed according to Gaussian PDF respectively characterized by the following means and standard deviations

$$\begin{aligned} \mu_\alpha &= 4 [deg] & \sigma_\alpha &= 0.1 [deg] \\ \mu_M &= 0.6 & \sigma_M &= 0.01 \end{aligned} \quad (3)$$

It is possible to generate n pseudo-random samples around the mean values of the input variables following the prescribed distributions Fig.4 and to compute for each of them the corresponding \mathbf{p} field by the CFD analysis.

Considering the i -th sample with $i = 1, \dots, n$:

- α_i and M_i are the values assumed by the variables α and M ,
- $p_j(\alpha_i, M_i)$ with $j = 1, \dots, m$ is the pressure in the j -th mesh cell Fig.5,

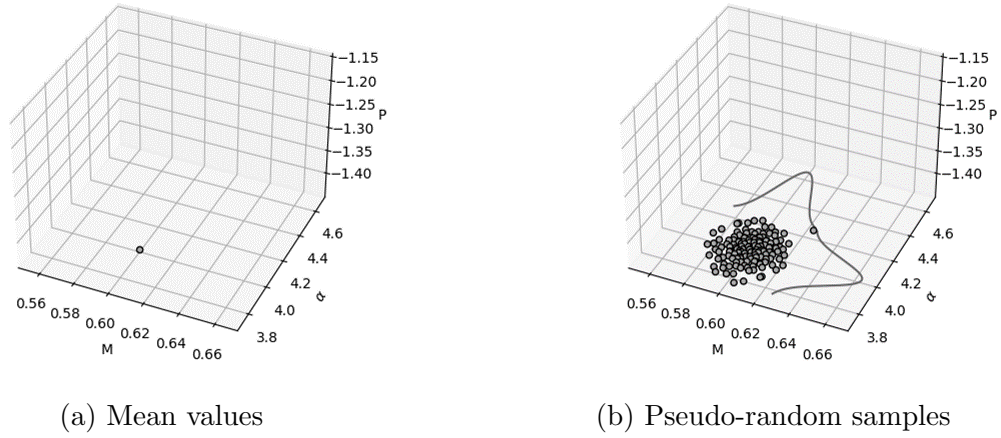
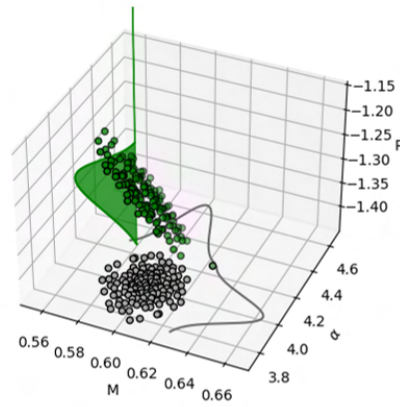


Figure 4: Design variables distributions


Figure 5: Pressure in the j – th mesh cell $p_j(\alpha_i, M_i)$ with $i = 1, \dots, n$

- $\mathbf{p}(\alpha_i, M_i) = [p_1(\alpha_i, M_i) \cdots p_m(\alpha_i, M_i)]$ is the pressure field in every mesh cell and is named i – th snapshot.

The n corresponding snapshots are collected in the $n \times m$ snapshots matrix

$$\mathbf{S}_p = \begin{bmatrix} \mathbf{p}(\alpha_1, M_1) \\ \vdots \\ \mathbf{p}(\alpha_n, M_n) \end{bmatrix} = \begin{bmatrix} p_1(\alpha_1, M_1) & \cdots & p_m(\alpha_1, M_1) \\ \vdots & \ddots & \vdots \\ p_1(\alpha_n, M_n) & \cdots & p_m(\alpha_n, M_n) \end{bmatrix} \quad (4)$$

The snapshots computations have been automated through modeFRONTIER, software from ESTECO, specialized in process automation and optimization which allows to create workflows that integrate external software packages and programming languages. The workflow, shown in Fig.6, integrates the OpenFOAM model through the application node EasyDriver. It repeats the CFD analysis and computes the performance of the airfoil several times by changing the values of the Mach number and of the angle of attack. The design variables are represented by the input parameter nodes M and $angle$, the airfoil performance is represented by the output parameter nodes c_l , lift coefficient, and c_d , drag coefficient. At run time, the values of the design variables are used to evaluate

the far-field velocity along the x- and y-axes and the lift and drag directions provided as boundary conditions to the CFD analysis. The EasyDriver modifies the OpenFOAM model files accordingly, submits the CFD run, and retrieves the lift and drag coefficients values from the post-processing files. The process is automatically iterated to evaluate multiple designs by changing the design variables values and updating the lift and drag coefficients values.

The flexibility of EasyDriver allows to run the external software on a local machine as well as to distribute the computations to remote resources available in the network. In this specific case, the CFD environment, i.e. OpenFOAM, has been packaged in a Docker container that is started for each CFD execution request. Running the analysis in a container allows to easily run OpenFOAM on different operating systems using the same commands and to fully exploit the parallelization capabilities of modeFRONTIER, which can evaluate multiple designs at the same time, i.e. multiple sets of values of the design variables, running multiple OpenFOAM instances even on a local machine.

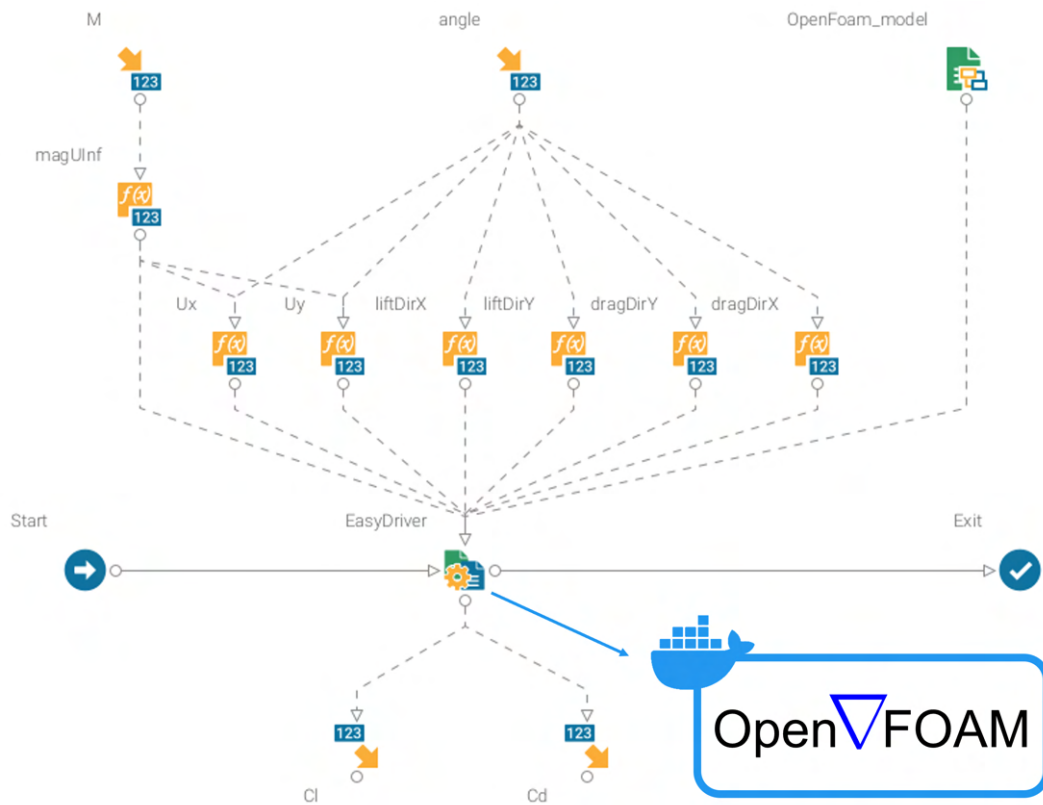


Figure 6: modeFRONTIER process automation workflow

3 THEORETICAL BACKGROUND

3.1 The Polynomial Chaos Expansion

Consider a random variable y , defined as a function f of a set of s independently distributed random variables $x^{(1)}, \dots, x^{(s)}$

$$y = f(x^{(1)}, \dots, x^{(s)}) \quad (5)$$

The general idea of PCE methodology is to represent y as a series of a family of polynomials, referred as *chaos*. The statistical moments can be analytically computed once the coefficients of the series are estimated. The maximum degree of the series expansion is called *chaos order*.

If y depends on only one uncertain variable $y = f(x)$ and x is distributed according to a probability density function $w(x)$, then there exists a special family of orthogonal polynomials

$$\boldsymbol{\psi} = \{\psi_0(x), \psi_1(x), \dots, \psi_{10}(x), \dots\} \quad (6)$$

such that the w -scalar product is

$$\langle \psi_i(x), \psi_j(x) \rangle_w = \int \psi_i(x) \psi_j(x) w(x) dx = 0 \text{ whenever } i \neq j. \quad (7)$$

Assume that $f(x)$ is a linear combination of this family

$$f(x) = \sum_{d=0}^{\infty} \beta_d \psi_d(x) \quad (8)$$

It is possible to demonstrate, by exploiting the orthogonality condition, that the mean and the standard deviation of y are

$$\mu_y = \beta_0 \quad (9)$$

$$\sigma_y^2 = \sum_{d=1}^{\infty} \beta_d^2 \|\psi_d(x)\|_w^2 \quad (10)$$

where the $w(x)$ -norm is

$$\|\psi_d(x)\|_w^2 = \langle \psi_d(x), \psi_d(x) \rangle_w = \int \psi_d(x) \psi_d(x) w(x) dx \quad (11)$$

In particular, if x is standard normal and $w(x) = \frac{e^{-0.5x^2}}{\sqrt{2\pi}}$ is Gaussian, the family of orthogonal polynomials is formed by the Hermite polynomials:

$$\begin{aligned} \psi_0(x) &= He_0(x) = 1, \\ \psi_1(x) &= He_1(x) = x, \\ \psi_2(x) &= He_2(x) = x^2 - 1, \\ \psi_3(x) &= He_3(x) = x^3 - 3x, \\ &\vdots \\ \psi_{10}(x) &= He_{10}(x) = x\psi_9(x) - 10\psi_8(x), \\ &\vdots \end{aligned} \quad (12)$$

Their $w(x)$ -norm is

$$\|\psi_d(x)\|_w^2 = d! \quad (13)$$

In multidimensional problems, the multivariate chaoses are written considering the tensorial products of the polynomials corresponding to the distributions of each individual

variable. In the two dimensional case $y = f(x^{(1)}, x^{(2)})$ with two identical normal distributions, the family of orthogonal polynomials orthogonal associated to the joint probability distribution $w_1(x^{(1)}) \otimes w_2(x^{(2)})$ starts with:

$$\begin{aligned}
\psi_0(x^{(1)}, x^{(2)}) &= 1, \\
\psi_1(x^{(1)}, x^{(2)}) &= He_1(x^{(1)}), \\
\psi_2(x^{(1)}, x^{(2)}) &= He_1(x^{(2)}), \\
\psi_3(x^{(1)}, x^{(2)}) &= He_2(x^{(1)}), \\
\psi_4(x^{(1)}, x^{(2)}) &= He_1(x^{(1)})He_1(x^{(2)}), \\
\psi_5(x^{(1)}, x^{(2)}) &= He_2(x^{(2)}), \\
&\vdots
\end{aligned} \tag{14}$$

and the previous equations Eq.(8) Eq.(9) Eq.(10) become respectively

$$y = f(x^{(1)}, x^{(2)}) = \sum_{d=0}^{\infty} \beta_d \psi_d(x^{(1)}, x^{(2)}) \tag{15}$$

$$\mu_y = \beta_0 \tag{16}$$

$$\sigma_y^2 = \sum_{d=1}^{\infty} \beta_d^2 \|\psi_d(x^{(1)}, x^{(2)})\|_w^2 \tag{17}$$

In practice, the chaos expansion is usually truncated at a fixed chaos order as

$$y = f(x^{(1)}, x^{(2)}) \simeq \sum_{d=0}^t \beta_d \psi_d(x^{(1)}, x^{(2)}) \tag{18}$$

$$\mu_y \simeq \beta_0 \tag{19}$$

$$\sigma_y^2 \simeq \sum_{d=1}^t \beta_d^2 \|\psi_d(x^{(1)}, x^{(2)})\|_w^2 \tag{20}$$

Thus, writing a 2-variable polynomial chaos of order 2 requires 6 polynomials ($t = 5$) as shown in Eq.(14).

The estimation of the statistical moments is given by the approximation of the coefficients β_d of the polynomial chaos expansion. A possible strategy to determine the coefficients β_d consists in a nonlinear regression, a procedure called *chaos collocation* [1]. The β_d are chosen in order to minimize the sum of the squared differences between $f(x^{(1)}, x^{(2)})$ and the chaos expansion, over a given set of points.

3.2 The Snapshot Proper Orthogonal Decomposition

The Snapshot POD algorithm is preferred to the so called direct method because it is faster when the number of the mesh cells m is greater than the number of snapshots n , as shown in [2].

From Eq.(4), it is possible to obtain the snapshot matrix \mathbf{P} of the pressure fluctuations by removing the average pressures from their respective columns

$$\begin{aligned}
\mathbf{P} &= \begin{bmatrix} p_1(\alpha_1, M_1) - \bar{p}_1 & \cdots & p_m(\alpha_1, M_1) - \bar{p}_m \\ \vdots & \ddots & \vdots \\ p_1(\alpha_n, M_n) - \bar{p}_1 & \cdots & p_m(\alpha_n, M_n) - \bar{p}_m \end{bmatrix} = \\
&= \begin{bmatrix} p'_1(\alpha_1, M_1) & \cdots & p'_m(\alpha_1, M_1) \\ \vdots & \ddots & \vdots \\ p'_1(\alpha_n, M_n) & \cdots & p'_m(\alpha_n, M_n) \end{bmatrix} = \\
&= \begin{bmatrix} p'_{11} & p'_{12} & \cdots & p'_{1m} \\ \vdots & \ddots & \ddots & \vdots \\ p'_{n1} & p'_{n2} & \cdots & p'_{nm} \end{bmatrix}
\end{aligned} \tag{21}$$

with

$$\bar{p}_j = \frac{\sum_{i=1}^n p_j(\alpha_i, M_i)}{n} \text{ with } j = 1, \dots, m \tag{22}$$

It is possible to evaluate the $n \times n$ correlation matrix

$$\mathbf{C}_s = \frac{\mathbf{P}\mathbf{P}^T}{n-1} = \frac{1}{n-1} \begin{bmatrix} \sum_{j=1}^m p'_{1j}{}^2 & \cdots & \sum_{j=1}^m p'_{1j}p'_{nj} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^m p'_{nj}p'_{1j} & \cdots & \sum_{j=1}^m p'_{nj}{}^2 \end{bmatrix} \tag{23}$$

and its n eigenvalues and n eigenvectors collected respectively in the matrices $\boldsymbol{\lambda}_s$ and \mathbf{A}_s

$$\boldsymbol{\lambda}_s, \mathbf{A}_s = \text{eig}(\mathbf{C}_s) \tag{24}$$

$$\boldsymbol{\lambda}_s = [\lambda_1 \quad \cdots \quad \lambda_n] \tag{25}$$

$$\mathbf{A}_s = [\mathbf{a}_1 \quad \cdots \quad \mathbf{a}_n] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} a_1(\alpha_1, M_1) & \cdots & a_n(\alpha_1, M_1) \\ \vdots & \ddots & \vdots \\ a_1(\alpha_n, M_n) & \cdots & a_n(\alpha_n, M_n) \end{bmatrix} \tag{26}$$

As a convention, the eigenvalues are ordered from the largest to the smallest. Consequently, the first eigenvector \mathbf{a}_1 corresponds to the largest eigenvalue λ_1 and so on and so forth. The element $a_k(\alpha_i, M_i)$ of the \mathbf{A}_s matrix is referred as the k -th coefficient of the i -th snapshot with $k = 1, \dots, n$ and $i = 1, \dots, n$. The eigenvectors of \mathbf{C}_s are named parameter modes. The $m \times n$ matrix $\boldsymbol{\Phi}_s$ is obtained, by projecting the snapshot matrix \mathbf{P} onto each orthogonal parameter mode

$$\boldsymbol{\Phi}_s = \mathbf{P}^T \mathbf{A}_s = \begin{bmatrix} p'_{11} & \cdots & p'_{n1} \\ p'_{12} & \ddots & p'_{n2} \\ \vdots & \ddots & \vdots \\ p'_{1m} & \cdots & p'_{nm} \end{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} \phi_{11} & \cdots & \phi_{n1} \\ \phi_{12} & \ddots & \phi_{n2} \\ \vdots & \ddots & \vdots \\ \phi_{1m} & \cdots & \phi_{nm} \end{bmatrix} = [\boldsymbol{\phi}_1 \quad \cdots \quad \boldsymbol{\phi}_n] \tag{27}$$

The matrix Φ_s contains n spatial coefficients ordered from the strongest to the weakest. The original snapshot matrix \mathbf{P} can be reconstructed from the spatial coefficients and parameter modes as

$$\Phi_s = \mathbf{P}^T \mathbf{A}_s \Rightarrow \mathbf{P}^T = \Phi_s \mathbf{A}_s^{-1} = \Phi_s \mathbf{A}_s^T \Rightarrow \mathbf{P} = \mathbf{A}_s \Phi_s^T \quad (28)$$

or, written explicitly, as

$$\begin{aligned} \mathbf{P} = \mathbf{A}_s \Phi_s^T &= \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1m} \\ \vdots & \ddots & \ddots & \vdots \\ \phi_{n1} & \phi_{n2} & \cdots & \phi_{nm} \end{bmatrix} = \\ &= \begin{bmatrix} a_{11}\phi_{11} + \cdots + a_{1n}\phi_{n1} & \cdots & a_{11}\phi_{1m} + \cdots + a_{1n}\phi_{nm} \\ \vdots & \ddots & \vdots \\ a_{n1}\phi_{11} + \cdots + a_{nn}\phi_{n1} & \cdots & a_{n1}\phi_{1m} + \cdots + a_{nn}\phi_{nm} \end{bmatrix} = \\ &= \begin{bmatrix} a_{11}\phi_{11} & \cdots & a_{11}\phi_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1}\phi_{11} & \cdots & a_{n1}\phi_{1m} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1n}\phi_{n1} & \cdots & a_{1n}\phi_{nm} \\ \vdots & \ddots & \vdots \\ a_{nn}\phi_{n1} & \cdots & a_{nn}\phi_{nm} \end{bmatrix} = \\ &= \begin{bmatrix} a_{11} \\ \vdots \\ a_{n1} \end{bmatrix} \begin{bmatrix} \phi_{11} & \cdots & \phi_{1m} \end{bmatrix} + \cdots + \begin{bmatrix} a_{1n} \\ \vdots \\ a_{nn} \end{bmatrix} \begin{bmatrix} \phi_{n1} & \cdots & \phi_{nm} \end{bmatrix} = \\ &= \sum_{k=1}^n \begin{bmatrix} a_{1k} \\ \vdots \\ a_{nk} \end{bmatrix} \begin{bmatrix} \phi_{k1} & \cdots & \phi_{km} \end{bmatrix} = \\ &= \sum_{k=1}^n \tilde{\mathbf{P}}_k \end{aligned} \quad (29)$$

The snapshot matrix \mathbf{P} is expressed as the sum of the contributions from the n modes. It is possible to get an approximation of the snapshot matrix \mathbf{P} by truncating the series of Eq.(29) to consider only the contributions of the most important modes. Usually, the importance of the modes can be retrieved by computing their *energy* contribution as

$$\frac{1}{\sum_{k=1}^n \lambda_k} [\lambda_1 \quad \cdots \quad \lambda_n] = [e_1 \quad \cdots \quad e_n] \quad (30)$$

The POD is used to build a ROM capable of predicting the pressure field $\mathbf{p}(\alpha, M)$ for new values of α and M not included in the n pseudo-random samples initially generated. Each component of \mathbf{P} of Eq.(29) represents the pressure fluctuation in the j – *th* cell for the i – *th* sample and can be written as

$$p'_{ij} = \sum_{k=1}^n a_{ik} \phi_{kj} \quad \text{with } i = 1, \dots, n \text{ and } j = 1, \dots, m \quad (31)$$

or as a function of α and M as

$$p'_j(a_1(\alpha, M), a_2(\alpha, M), \dots, a_n(\alpha, M)) = \sum_{k=1}^n a_k(\alpha, M) \phi_{kj} \quad \text{with } j = 1, \dots, m \quad (32)$$

To make a prediction of p'_j for a new set of values of α and M , it is sufficient to estimate, using any regression model, the new values of the n coefficients $a_k(\alpha, M)$.

The pressure in the $j - th$ cell for a new sample is consequently

$$p_j(\alpha, M) = p'_j(a_1(\alpha, M), a_2(\alpha, M), \dots, a_n(\alpha, M)) + \bar{p}_j = \sum_{k=1}^n a_k(\alpha, M)\phi_{kj} + \bar{p}_j \quad (33)$$

4 UNCERTAINTY QUANTIFICATION METHODS

The Snapshot POD and the PCE can be used alone or in combination to develop various UQ methods that allow estimating the statistical moments, the mean μ_{p_j} and the standard deviation σ_{p_j} , of the pressure in each $j - th$ mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ and to obtain the vectors $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p$ defined in Eq.(1) and in Eq.(2).

4.1 Direct method

The Direct method is a classic UQ technique that requires neither the POD nor the PCE described in section 3.

The $j - th$ column of Eq.(4) contains the n evaluated values of the pressure in the $j - th$ mesh cell $p_j(\alpha_1, M_1), \dots, p_j(\alpha_n, M_n)$ with $j = 1, \dots, m$. It is possible to directly compute the desired mean μ_{p_j} and standard deviation σ_{p_j} as

$$\mu_{p_j} = \frac{\sum_{i=1}^n p_j(\alpha_i, M_i)}{n} \quad (34)$$

$$\sigma_{p_j} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (p_j(\alpha_i, M_i) - \mu_{p_j})^2} \quad (35)$$

Depending on the algorithm used to generate the n samples $(\alpha_1, M_1), \dots, (\alpha_n, M_n)$ distributed according to the prescribed probability density functions, the accuracy of the moment estimation changes [1]. When the Monte Carlo algorithm is used, many samples are needed to obtain reliable results. It can be shown that by using Latin Hypercube, Sobol, or other algorithms, fewer samples are required to provide better results. If a sufficient number of samples is chosen, it is possible to consider the moments estimated by the Direct method as reference values, especially when the analytical solutions are not available.

4.2 PCE method

The PCE method relies only on the concepts described in section 3.1. Remembering Eq.(18), the uncertain pressure in the $j - th$ mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ as a function of the uncertain variables α and M can be written as

$$p_j(\alpha, M) \simeq \sum_{d=0}^t \beta_d \psi_d(\alpha, M) \quad (36)$$

Consequently the mean μ_{p_j} and the standard deviation σ_{p_j} are estimated as

$$\mu_{p_j} \simeq \beta_0 \quad (37)$$

$$\sigma_{p_j}^2 \simeq \sum_{d=1}^t \beta_d^2 \|\psi_d(\alpha, M)\|_w^2 \quad (38)$$

To evaluate the moments in the j -th mesh cell, it is necessary to determine the coefficients β_d of the polynomial chaos expansion through the chaos collocation, i.e. a nonlinear regression. It is necessary to repeat the chaos collocation m times, one for each mesh cell.

4.3 ROM method

The ROM method relies only on the concepts described in section 3.2. It is possible

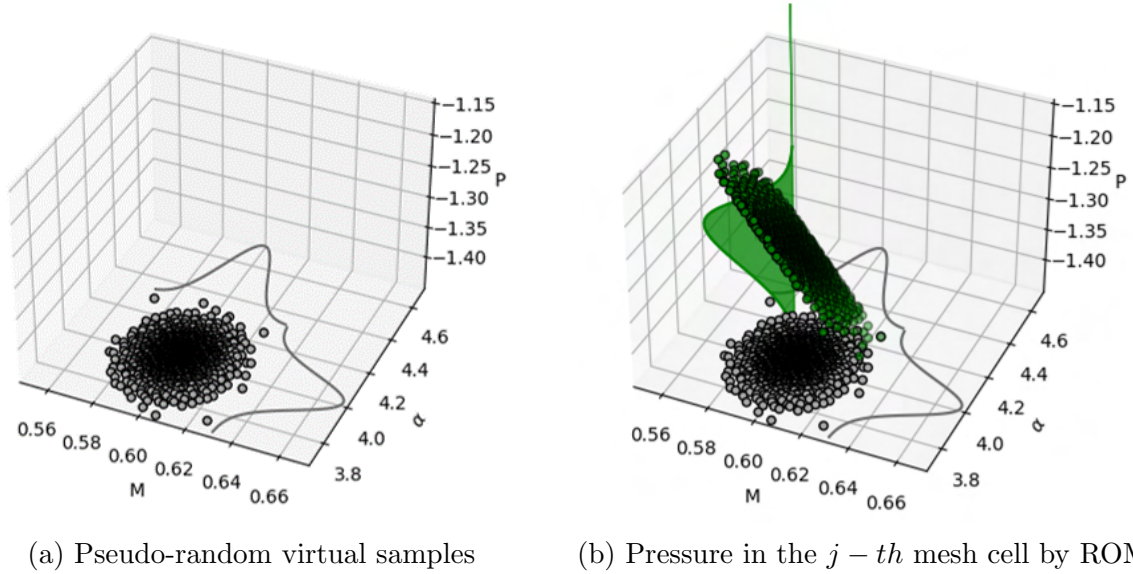


Figure 7: ROM method sampling

to generate other n_{virt} virtual pseudo-random samples around the mean values following the prescribed distributions Fig.7a, with $n_{virt} \gg n$. For each i_{virt} -th virtual sample with $i_{virt} = 1, \dots, n_{virt}$, it is possible to estimate the pressure field $\mathbf{p}(\alpha_{i_{virt}}, M_{i_{virt}}) = [p_1(\alpha_{i_{virt}}, M_{i_{virt}}) \cdots p_m(\alpha_{i_{virt}}, M_{i_{virt}})]$ using the Reduced Order Model Fig.7b. For each j -th mesh cell with $j = 1, \dots, m$ the mean μ_{p_j} and standard deviation σ_{p_j} are computed as

$$\mu_{p_j} = \frac{\sum_{i=1}^{n_{virt}} p_j(\alpha_{i_{virt}}, M_{i_{virt}})}{n_{virt}} \quad \text{with } j = 1, \dots, m \quad (39)$$

$$\sigma_{p_j} = \sqrt{\frac{1}{n_{virt} - 1} \sum_{i=1}^{n_{virt}} (p_j(\alpha_{i_{virt}}, M_{i_{virt}}) - \mu_{p_j})^2} \quad \text{with } j = 1, \dots, m \quad (40)$$

The moments are computed exactly as in the Direct method. However, since $n_{virt} \gg n$, the results are expected to be more accurate.

4.4 POD method

The POD method relies only on the concepts described in section 3.2. Eq.(33) expresses the uncertain pressure in the j -th mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ as a function of the n uncertain POD coefficients $a_k(\alpha, M)$. The desired mean μ_{p_j} and variance $\sigma_{p_j}^2$ can

be computed as

$$\mu_{p_j} = \mu_{p'_j} + \bar{p}_j = \sum_{k=1}^n \mu_{a_k} \phi_{kj} + \bar{p}_j \quad \text{with } j = 1, \dots, m \quad (41)$$

$$\sigma_{p_j}^2 = \sigma_{p'_j}^2 = \sum_{k=1}^n \sigma_{a_k}^2 \phi_{kj}^2 \quad \text{with } j = 1, \dots, m \quad (42)$$

due to the error propagation of the uncorrelated variables $a_k(\alpha, M)$. It is possible to analytically evaluate the mean μ_{a_k} and the standard deviation σ_{a_k} of the POD coefficients $a_k(\alpha, M)$ as

$$\mu_{a_k} = \frac{\sum_{i=1}^n a_k(\alpha_i, M_i)}{n} \quad (43)$$

$$\sigma_{a_k}^2 = \frac{\sum_{i=1}^n (a_k(\alpha_i, M_i) - \mu_{a_k})^2}{n - 1} \quad (44)$$

It is possible to truncate the series of Eq.(33) and Eq.(41)-Eq.(44) to keep only the modes with the highest energy contributions.

4.5 POD PCE method

The POD PCE method combines the Polynomial Chaos Expansion and the Snapshot Proper Orthogonal Decomposition described respectively in section 3.1 and 3.2. As in the POD method, the uncertain pressure in the j -th mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ as a function of the n uncertain POD coefficients $a_k(\alpha, M)$ is written as in Eq.(33), eventually retaining only the important modes by truncating the series. Similarly, the pressure mean μ_{p_j} and variance $\sigma_{p_j}^2$ are written as in Eq.(41) and Eq.(42). To compute the moments, it is sufficient to estimate the mean μ_{a_k} and the standard deviation σ_{a_k} of the POD coefficients $a_k(\alpha, M)$. Remembering Eq.(18), $a_k(\alpha, M)$ can be expressed as

$$a_k(\alpha, M) \simeq \sum_{d=0}^t \beta_d \psi_d(\alpha, M) \quad (45)$$

and the mean and the variance as

$$\mu_{a_k} \simeq \beta_0 \quad (46)$$

$$\sigma_{a_k}^2 \simeq \sum_{d=1}^t \beta_d^2 \|\psi_d(\alpha, M)\|_w^2 \quad (47)$$

To evaluate the moments in every mesh cell, it is necessary to determine the coefficients β_d of the polynomial chaos expansion by the chaos collocation, i.e. a nonlinear regression, for each POD coefficient $a_k(\alpha, M)$. The chaos collocation is performed n times or less, depending on the number of retained modes. The method is more efficient than the PCE method, which performs a nonlinear regression m times, one for each mesh cell.

5 UNCERTAINTY QUANTIFICATION METHODS COMPARISON

By choosing one of the previously described UQ methods and generating an appropriate number of n pseudo-random samples $(\alpha_1, M_1), \dots, (\alpha_n, M_n)$ around the means of the design variables α and M , it is possible to estimate the mean μ_{p_j} and the standard deviation σ_{p_j} of the pressure in each j -th mesh cell $p_j(\alpha, M)$ with $j = 1, \dots, m$ and reconstruct the mean μ_p and standard deviation σ_p pressure fields shown in Fig.8. The region where the effect of design variable uncertainties is greater, i.e. where the standard deviation pressure field has the highest values, is near the leading edge of the airfoil. In the far field, near the inlet of the CFD domain, the values of the standard deviation pressure field are lower.

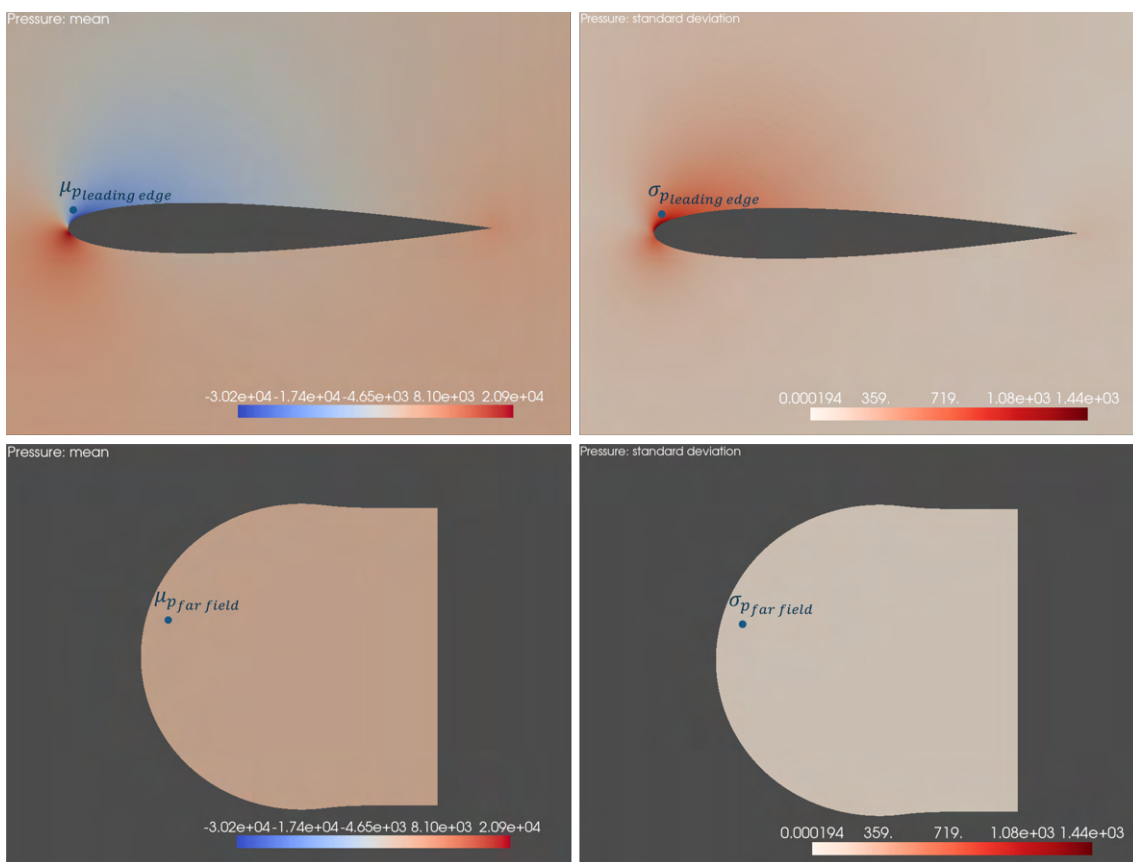


Figure 8: Pressure mean and standard deviation fields

To accurately estimate the fields μ_p and σ_p , each UQ method requires a different number of n pseudo-random samples, i.e. a different number of CFD simulations. By progressively increasing the number n of samples, it is possible to compare the values provided by each UQ method for the mean and standard deviation of the pressure in a given mesh cell.

The accuracy of the various UQ methods, as shown for the Direct one, also depends on the algorithm used to generate the n samples distributed accordingly to the prescribed probability density functions. A comparison of the different sampling strategies is beyond the scope of this study.

Two representative mesh cells, one near the leading edge of the airfoil and one in the

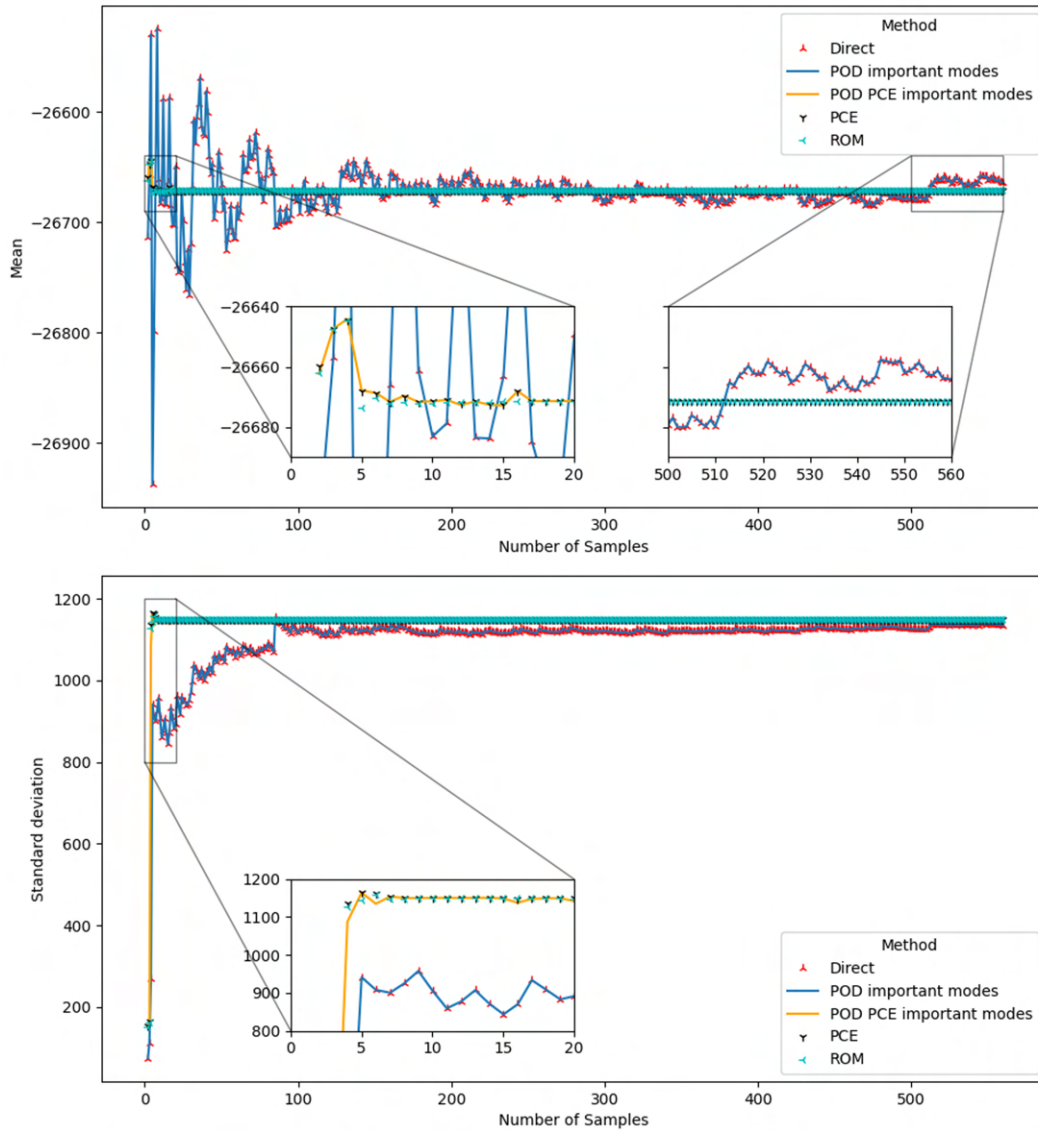


Figure 9: Comparison among UQ methods of convergence of mean (top) and standard deviation (bottom) of pressure near the airfoil leading edge

far field, have been selected for the comparison. Fig.9 shows the convergence of the mean and standard deviation values evaluated in the cell near the airfoil, Fig.10 shows the convergence in the far field cell. The computational time required to compute the mean and standard deviation pressure fields is reported in Fig.11.

The direct method is reliable, but requires many samples to be accurate. Especially near the leading edge, the convergence is not achieved even when more than 550 designs are evaluated.

The PCE method is very accurate and requires only a few CFD snapshots to converge near the leading edge where the standard deviation value is relevant. The method requires more samples in the far field, Fig.10, where the pressure values are lower and more uniform among the snapshots. The technique requires the PCE to be computed for each mesh cell; the computational time can increase dramatically with the number of mesh

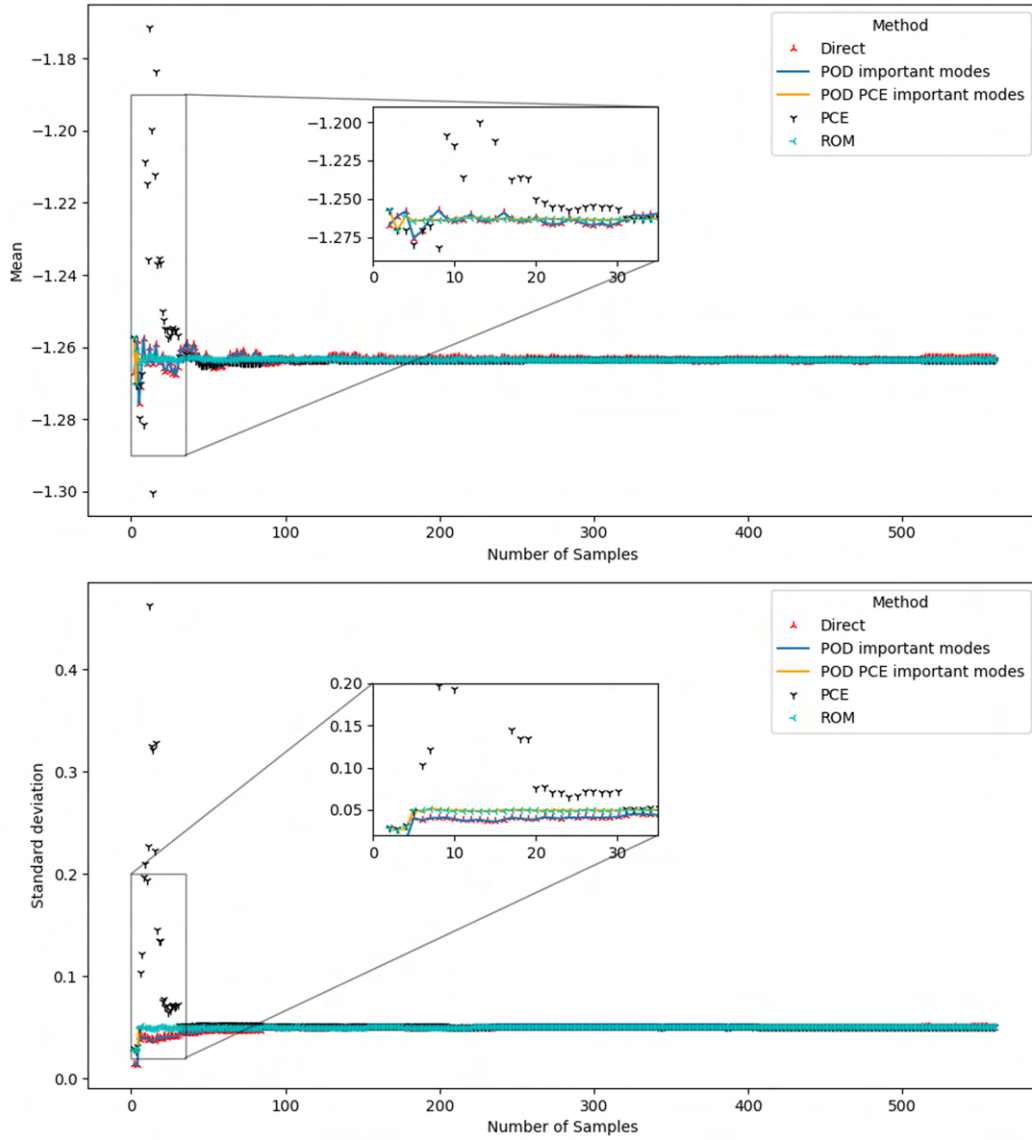


Figure 10: Comparison among UQ methods of convergence of mean (top) and standard deviation (bottom) of pressure in far field cell

cells. It is recommended to perform a convergence study by changing the order of the polynomial chaos when using the PCE method. The main drawback of the PCE technique is that the number of polynomial coefficients scales more than exponentially with the number of uncertain input variables and the chaos order. The number of coefficients of the polynomial expansion of order o involving n_{var} variables is given by

$$\frac{(n_{var} + o)!}{n_{var}!o!} \quad (48)$$

The number n of samples must be at least equal to or greater than the number of coefficients to perform the chaos collocation. This means that it may require many samples, i.e. many CFD snapshots, if there are many uncertain input variables. It is possible to improve this behavior by adopting the Adaptive Sparse Polynomial Chaos Expansion method [5].

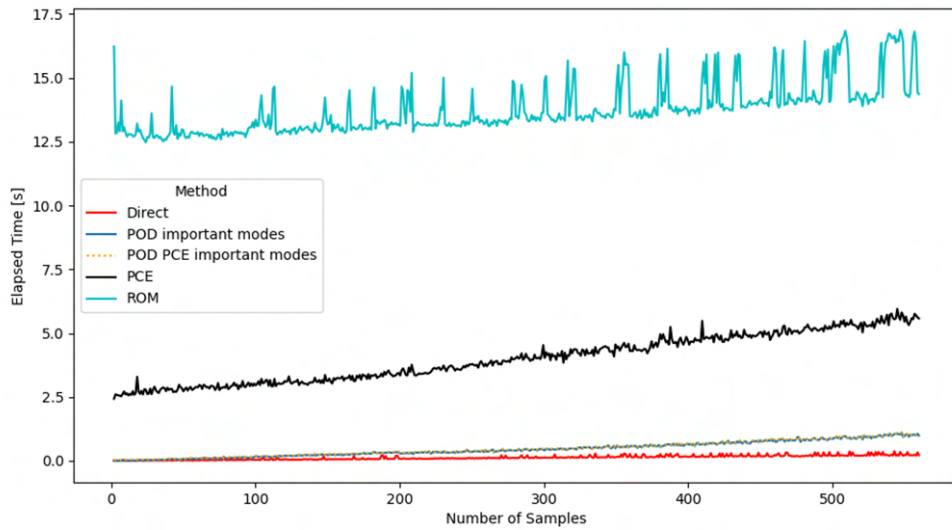


Figure 11: Computational time comparison among UQ methods

The ROM method is accurate both near the leading edge and in the far field and quickly converges. The accuracy is similar to the PCE method if enough n_{virt} virtual samples are estimated to compute the mean and the standard deviation of the pressure. The computational time required is longer. Fig.12 shows the effect of the number of virtual samples on the accuracy and on the time. It is recommended to perform a convergence study by changing the number of virtual samples when using the ROM method.

The POD method provides exactly the same results as the Direct method in terms of accuracy. It introduces only a small runtime overhead compared to the Direct method itself if are retained only the most important modes.

The POD PCE method is similar to the ROM approach in terms of accuracy and takes almost the same amount of time as the POD method if are retained only the most important modes. Relying on PCE, the method could require more samples if the number of uncertain input variables is high. Again, it could be improved by using the Adaptive Sparse Polynomial Chaos Expansion technique.

In the airfoil use case, since their cumulative energy contribution is close to 1, only the first two modes are considered important modes. Fig.13 shows the effect of the number of modes on the accuracy and on the computational time. Both POD and POD PCE considering only the two important modes are at least as accurate as the methods considering all modes, but the computational time advantage is significant.

Luca Battaglia, Federico Carlini, Alberto Clarich and Rosario Russo

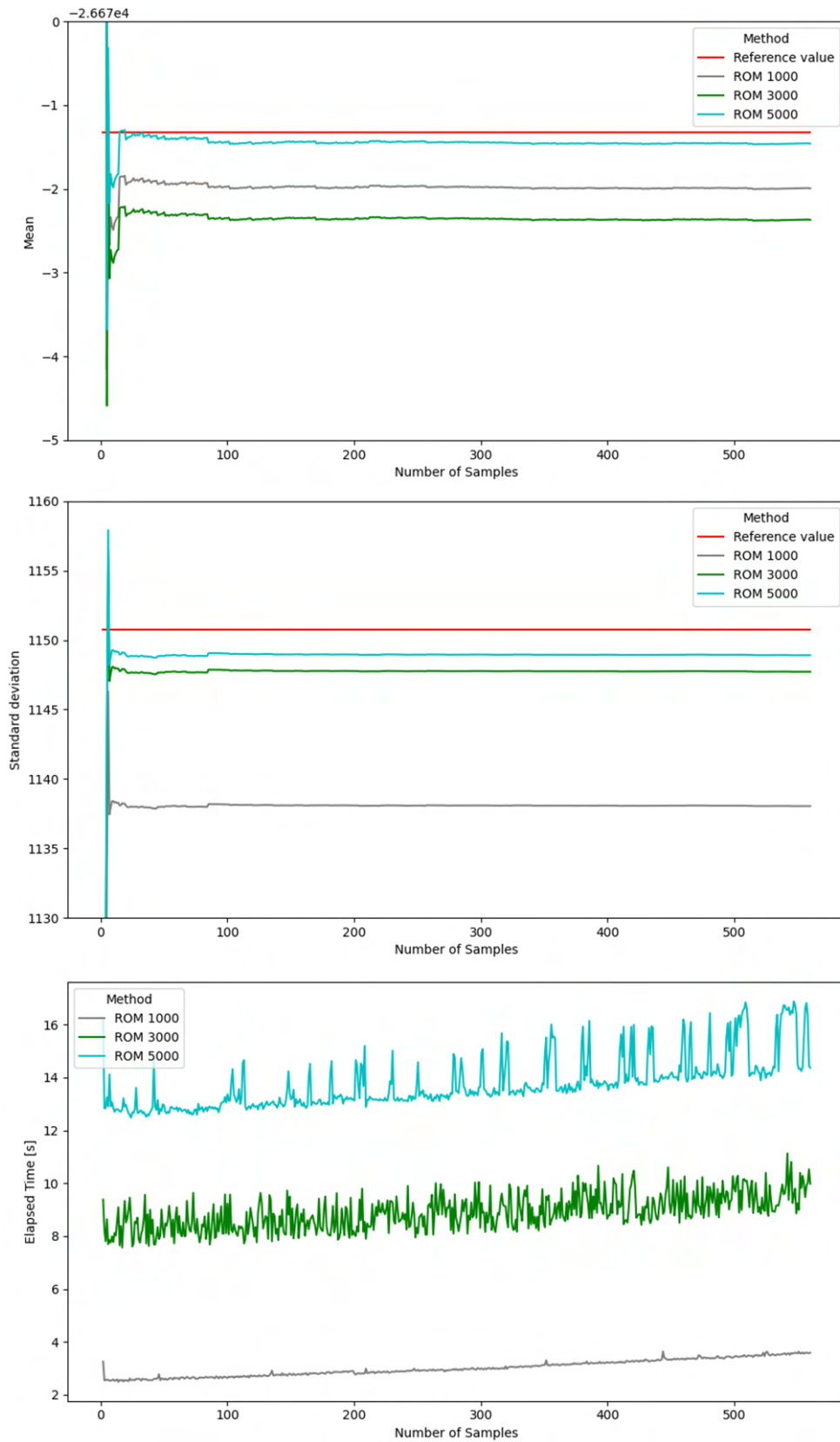


Figure 12: Convergence of mean (top) and standard deviation (center) of pressure near the airfoil leading edge evaluated by ROM method and computational time (bottom)

Luca Battaglia, Federico Carlini, Alberto Clarich and Rosario Russo

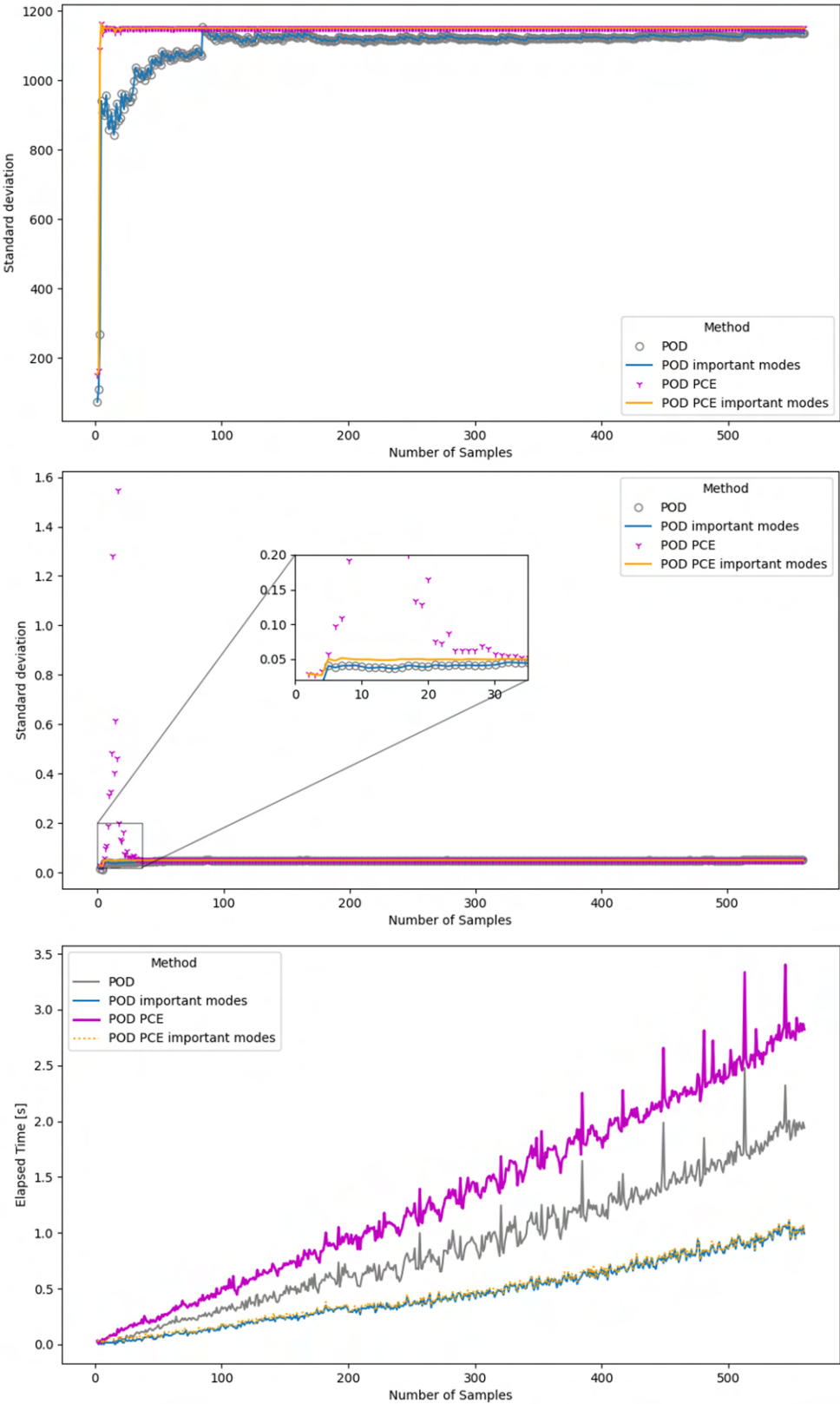


Figure 13: Convergence of standard deviation of pressure near the airfoil leading edge (top) and in a far field cell (center) evaluated by POD and POD PCE methods and computational time (bottom)

6 CONCLUSIONS

An efficient POD PCE uncertainty quantification methodology based on non-intrusive Snapshot Proper Orthogonal Decomposition and Polynomial Chaos Expansion has been developed. It inherits the advantages of both POD and PCE approaches. Several other techniques based only on PCE or ROM have been proposed during this study and their main differences, strengths and weaknesses with respect to the POD PCE have been highlighted. Considering a mesh with millions of cells, the POD PCE method is more convenient than the PCE method. If the number of uncertain variables is not excessive, it requires less computational time than the ROM method to provide results with similar accuracy. In the other cases, the PCE or ROM methods can still provide good performance and are easier to implement.

REFERENCES

- [1] S. Poles and A. Lovison. A Polynomial Chaos Approach to Robust Multiobjective Optimization. In K. Deb, S. Greco, K. Miettinen, and E. Zitzler, editors, *Hybrid and Robust Approaches to Multiobjective Optimization*, volume 9041 of *Dagstuhl Seminar Proceedings (DagSemProc)*, pages 1–15, Dagstuhl, Germany, 2009. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/DagSemProc.09041.7. URL <https://drops.dagstuhl.de/opus/volltexte/2009/2000>.
- [2] J. Weiss. *A Tutorial on the Proper Orthogonal Decomposition*. doi:10.2514/6.2019-3333. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2019-3333>.
- [3] L. Sirovich. Turbulence and the dynamics of coherent structures. i. *Quarterly of Applied Mathematics - QUART APPL MATH*, 45, 10 1987. doi:10.1090/qam/910463.
- [4] Christopher Rumsey. 2D NACA 0012 Airfoil Validation Case — Turbulence Modeling Resource. https://turbmodels.larc.nasa.gov/naca0012_val.html, 2022. [Online; accessed 6-April-2023].
- [5] G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345–2367, 2011. ISSN 0021-9991. doi:<https://doi.org/10.1016/j.jcp.2010.12.021>. URL <https://www.sciencedirect.com/science/article/pii/S0021999110006856>.



AEROELASTIC OPTIMIZATION OF A CELLULAR FLYING CAR WING USING THOMPSON SAMPLING EFFICIENT MULTIOBJECTIVE OPTIMIZATION (TS-EMO) ALGORITHM

Sen Wu* and Tomohiro Yokozeki

Department of Aeronautics and Astronautics
Graduate School of Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo City, Tokyo, Japan
globaswu@g.ecc.u-tokyo.ac.jp

Abstract *A computational model of a wing with octahedron cellular internal structure designated for a flying car has been proposed. The wing model is subjected to variation by around 20 aerodynamic and structural design variables. The objective is to find the best combinations of design variables that maximize the aerodynamic performance and minimize the structural weight, subjected to maximum allowable stress, and static and dynamic aeroelastic instability constraints, including divergence and flutter speeds. The lifting-line theory for low-speed subsonic flight has been adopted as the aerodynamic model, while the structural and aeroelastic properties were calculated using finite element (FEM) software package MSC Nastran. By incorporating the aerodynamic, structural and aeroelastic simulations with an algorithm called Thompson Sampling Efficient Multiobjective Optimization (TS-EMO) developed by E. Bradford et. al. [1], a two-dimensional Pareto front depicted by several points demonstrating the compromise between aerodynamic and structural weight has been obtained. The combinations of design variables behind these points provide Pareto-optimal design ideas of the flying car wing. It is anticipated that the Pareto-optimal designs could provide ideas about how a structurally optimal geometry of octahedron cells should look. Similar design and optimization methods might be implemented to other novel designs of wing internal structures.*

Keywords: Thompson Sampling Efficient Multiobjective Optimization (TS-EMO); cellular structured wings; multiobjective optimization (MOO); multidisciplinary optimization (MDO); flying cars

1. INTRODUCTION

In recent years, the morphing wing concept attracted the attention from aircraft design professionals. Compared to the conventional aircraft wings, the morphing wings would be more flexible to achieve a wider range of performance. Some researchers develop morphing wings by replacing part of the wing internal structure with more flexible metamaterials, say Yokozeki et. al. [2] developed a variable camber morphing airfoil with flexible corrugated structure near the trailing edge and obtained a better aerodynamic performance. Some researchers develop bio-inspired morphing wings, say Matloff et. al. [3] developed a soft morphing wing with feathers imitating the actuation mechanism of real birds. A comprehensive review on evolution of the morphing wings can be found from Li et. al.'s paper [4].

Among different morphing-wing concepts, the morphing wing with cellular structures is one of the research hotspots. There is an abundant amount of research on properties of cellular structures and metamaterials, and some researchers attempted to construct wings with cellular structures. Tsushima et. al. [5] constructed a wing with NACA 0010 airfoil with simple-cubic (SC) structure and investigated the mechanical properties of the wing by computational homogenization method, which has been verified by experimental testing. Jenett et. al. [6] developed a small octahedron-cellular structured wing by assembling discrete building-block elements, which can exhibit torsional morphing. Zhang et. al. [7] developed a wing computational model with octahedron structure by defining the density of the internal octahedron structures at different planform locations as design variables. Cramer et. al. from MIT and NASA [8] developed a large-scale multi-layer octahedron-cellular morphing wing of approximately 4-m span which achieved high stiffness at ultralight densities and an increased aerodynamic efficiency. Moon et. al. [9] manufactured a lightweight Unmanned Aerial Vehicle (UAV) wing structure with 3D printing technology.

Despite the arduous work by previous research, the current literature is still far from exploiting the potential of vast variety of cellular structures on aerospace applications. In particular, to the best of the author's knowledge, it is lack of literature in terms of multidisciplinary optimization (MDO) of both aerodynamic and structural objectives for aircraft wings with cellular internal structures. Although some literatures in the previous paragraph attempted to build some wings with cellular internal structures either computationally or realistically, the practicability and optimality of these designs remain questionable due to the lack of data and design varieties.

In this paper, a computational wing finite element model with octahedron internal structure which is specifically designed for a flying car application is proposed. The flying car chosen is the SD-05 prototype from SkyDrive Inc. in Japan [10]. The SD-05 prototype is a battery electric flying vehicle with Maximum Take-Off Weight (MTOW) of 1100 kg. The officially announced operational flight range is 10 km. Due to its drone-like design, although it has the Vertical Take-Off and Landing (VTOL) capability, the energy dissipation during the steady-level cruise would be as high as helicopters, in which the thrust of the rotors are mainly used for overcoming the whole weight of the vehicle. If a morphing wing could be expanded when the flying car enters a reasonably sparse air space at certain altitude, it would enable the flying car to glide in the air like a conventional civil aircraft. By adjusting the direction of the propeller, the thrust could be used just to overcome the gliding drag, greatly increasing the range of the flight.

It is believed that the flying car would become a more popular and cost-effective Urban Air

Mobility (UAM) product with an increased range of flight. For example, the distance between the center of Tokyo and the Tokyo Narita International Airport (NRT) is around 60 km. It is not even appropriate to use the 10-km range flying car to catch an urgent flight. The gliding wing of a flying car could be a game changer that enables more flexible and convenient intra- and inter-city trips.

This paper serves as an early phase research on the MDO of novel morphing wings with cellular / lattice internal structures for flying cars. Although we use “morphing” to describe the wing, the evaluation of objective functions and the analysis itself does not consider the active morphing of the wing. The term “morphing” is defined in the sense that the wing is ideally being contracted (folded up) when the flying car is situated in crowded urban low-altitude environment where expansion of the wing is not feasible.

It is worthy to note that the unconventionality of the internal structural design often means that it is difficult to find low-cost analytical solution to structural and aeroelastic characteristics. In this paper, the structural and aeroelastic properties of the proposed wing design was carried out by commercial FEM package MSC NASTRAN 2019. In terms of aerodynamic performance, we assume the flying car, hence the proposed morphing wing, is to be operated under subsonic incompressible flow ($Ma \leq 0.3$). Hence, it is not necessary to introduce a sweep angle as one of the design parameters, and it would be plausible to use Prandtl’s Lifting Line Theory (LLT) to predict the lift coefficients as well as to account for total drag with reasonable fidelity. The aerodynamic characteristics of the airfoils of the proposed wing is evaluated at several spanwise stations as suggested by LLT using the 2D version of the Computational Fluid Dynamics (CFD) solver UTCart developed by Imamura et. al. [11] [12]. The scope of this paper will be limited to wing optimization under steady level flight. It is anticipated that extra energy would be required in the take-off and landing phases of the flight, which could lead to a shortened practical range of flight.

Since the evaluation of the aerodynamic, structural and aeroelastic characteristics of the proposed wing design require FEM and CFD software packages, the evaluation of each design configuration would be time consuming. If we take the design variables of the proposed wing as the input variables of the aerodynamic and structural objective functions, running the simulation using FEM and CFD software package is a time-consuming costly process of evaluating the objective functions. The objective functions are de facto black-box functions since we have limited idea about how the aerodynamic and structural characteristics would be correlated with the design input variables. Bayesian Optimization (BO) is chosen as the method of solving this optimization problem owing to the costly and black-box feature of the objective functions [13]. In particular, the Thompson-Sampling Efficient Multiobjective Optimization (TS-EMO) Bayesian Optimization algorithm developed by E. Bradford et. al. [1] comes to our particular interest due to its open-source nature and the examples of successful implementations outside aerospace industry [14] [15].

This paper will be structured as follows. Section 1 described the development of morphing wings with lattice structures, their applications in aerospace applications, the proposed usage of lattice structures in flying car applications and the combination of software packages used. Section 2 will elaborate on the problem of optimization, including the definition and construction of the objective functions and constraint functions. The design variables of the 3D wing will be introduced, in which the airfoil geometry and internal arrangement and lining of the nonuniform octahedron cells will be detailed. Section 3 describes the interconnection between all the software packages used, namely MATLAB, 2D UTCart CFD, MSC

NASTRAN 2019 and TS-EMO. Section 4 shows the results of the optimization problem. In particular, a Pareto-front demonstrating the compromise between aerodynamic and structural objective functions has been obtained. Section 5 summarizes the paper and describes the future works.

2. OPTIMIZATION PROBLEM AND MODEL PARAMETERIZATION

2.1. Definition of Optimization Problem

We aim to design a morphing wing with an octahedron internal structure designated for a flying car prototype as described in Section 1. The optimization problem is to maximize the range of the vehicle under steady level flight, while minimizing the weight of the wing attached to the vehicle, subjected to no-yield and no-aeroelastic-instability requirements.

TS-EMO itself is an unconstrained minimization algorithm. Therefore, the objective functions need to be modified and take the constraints into account. Since we are maximizing the flight range, the objective function $f_1(\mathbf{x})$ in Eq. (1) is modified to be the negative of the range, i.e., $-R$. In order to strictly impose the constraints in unconstrained solver, death penalty is imposed to mathematically reject the designs with violation of structural and aeroelastic constraints. The modified objective functions could be written as:

$$\text{minimize: } \begin{cases} f_1(\mathbf{x}) = -R(\mathbf{x}) = -C \cdot \frac{(L/D)_{\max}}{m_0 + m(\mathbf{x})} + \infty \cdot \sum_{j=1}^3 C_j(\mathbf{x}) \\ f_2(\mathbf{x}) = m(\mathbf{x}) = \sum_{i \in \text{elem. ID}} m_i(\mathbf{x}) + \infty \cdot \sum_{j=1}^3 C_j(\mathbf{x}) \\ \begin{cases} C_1(\mathbf{x}) = \max \{0, \sigma_{\max}(\mathbf{x}) - \sigma_Y / 1.1\} \\ C_2(\mathbf{x}) = \max \{0, 1.2U_{\infty} - U_D\} \\ C_3(\mathbf{x}) = \max \{0, 1.2U_{\infty} - U_F\} \end{cases} \end{cases} \quad (1)$$

where

\mathbf{x} : vector of all aerodynamic and structural design variables to be elaborated in Section 2.2.

$R(\mathbf{x})$: range of flight, C is a constant related to the battery weight, battery efficiency and vehicle efficiency.

$m_0, m(\mathbf{x})$: original mass (1100 kg) of the flying car and the mass of the attached wing.

$\sigma_{\max}(\mathbf{x}), \sigma_Y$: the maximum stress across the model under steady level flight and the yield stress of the material being used.

U_{∞}, U_D, U_F : freestream air velocity, divergence speed and flutter speed.

Upon the violation of the constraints, an infinite penalty is added onto the objective functions to be minimized, hence leading to the rejection of such data points from being included in the set of Pareto-optimal solutions. Here 1.1 and 1.2 in C_1 through C_3 are safety factors imposed to no-yield condition and no-aeroelastic-instability condition respectively.

The objective function $f_1(\mathbf{x})$ is extracted from [16]. The constant C is related to the battery capacity and overall efficiency. The maximum lift-to-drag ratio, $(L/D)_{\max}$, is obtained by using Prandtl's Lifting Line Theory (LLT), which will be elaborated in later sections. The original mass of the vehicle is obtained from the official website of the SkyDrive SD-05 prototype,

while the mass of the added wing is obtained by MSC NASTRAN WEIGHTCHECK function. The objective function $f_2(\mathbf{x})$ is simply the summation of the weight of all elements in the FEM model constructed. The details of the structural FEM model will be described in later sections.

It is inevitable to bring extra weight onto the flying car by introducing a morphing extendable wing onto the vehicle. The range of flight is also sensitive to the weight of the vehicle. It is hence necessary to minimize the structural weight of the designed wing as well as to maximize the maximum lift-drag ratio $(L/D)_{\max}$ of the wing during steady gliding so as to increase the range of the flight. Although the range of the flight itself is a function of both structural weight and the $(L/D)_{\max}$, we deem them as two separate objective functions as the structural weight might also be related to the manufacturing cost.

Meanwhile, the wing design should be subjected to several constraints. Firstly, we should get rid of the yielding of the material to ensure structural integrity. Secondly, we should get rid of the occurrence of aeroelastic instabilities. In another word, we need to make sure that the maximum stress across the whole wing structure under steady level flight to be less than the yield stress of the material chosen. Also, we need to ensure that the designed flight speed of the flying vehicle should not be too close to the divergence and flutter speed to get rid of the material fracture caused by aeroelastic instability.

2.2. Definition of Wing Model and Design Variables

The wing model is defined by 18 aerodynamic and structural variables. The aerodynamic design variables specifies the geometry of the airfoil, while the structural variables specifies the skin thickness, cross-sectional properties of structural members and bring variations to internal octahedron structures.

2.2.1. Aerodynamic Design Variables: PARSEC Airfoil Parameterization

The airfoil geometry of the wing is defined by PARSEC Airfoil Parameterization. The detailed formulation of the PARSEC parameterization could be found in a paper from Vecchia et. al. [17] and a program written by Wu [18]. Although both [17] and [18] contain typos, readers are advised to consult them for a complete picture of the parameterization.

PARSEC airfoil uses eleven parameters, from p_1 to p_{11} , to define an airfoil. Each of the coefficient represent a geometric feature of an airfoil. The accuracy of the PARSEC airfoil parameterization has been cross-checked with the data of the RAE 2822 airfoil from Khurana et. al.'s [19] paper. In the optimization problem demonstrated in this paper, the design variables p_8 and p_9 are set to be zero.

2.2.2. Structural Design Variables and The Internal Octahedron Structures

The internal space bounded within the airfoil of the wing is entirely filled by a single layer of octahedron structure. The initial design is obtained by first constructing a rectangular wing without taper and washout, then add the taper and washout feature by coordinate transform. Figure 1 shows the top view of a half-span rectangular wing model designed with a chord length of $c = 1000$ mm and a span of $b = 7400$ mm (i.e., half-span of 3700 mm). The coordinate system is set up in accordance with the standard of MSC NASTRAN. The origin is located at the leading edge of the wing root. The x -axis is the chordwise direction, while the y -axis is the spanwise direction. The chord length c is a design variable and would be rounded to the nearest 100 mm for convenience of model generation.

Let P and Q be the number of chordwise and spanwise cells respectively. In Figure 1, each octahedron cell is occupying a $100 \text{ mm} \times 100 \text{ mm}$ space in the xy -plane in this example. The details of the NASTRAN grid point ID allocation and element connectivity is not elaborated in this paper. For any given values of c , b , P and Q , each octahedron cell is occupying a $(c/P) \times (b/Q)$ rectangular space in the xy -plane. The optimization problem in this paper will fix a $(c/10) \text{ mm} \times 100 \text{ mm}$ space for each octahedron cell in the xy -plane, i.e., P is fixed to be 10. The model is programmed such that the octahedron internal structure with any values of c , b , P and Q could be generated, as long as the number of cells in spanwise and chordwise directions, P and Q , do not exceed 98. The limitation arises from the 8-digit grid ID limit in NASTRAN. The octahedron cells are “uniform” in the sense of their allocation in xy -plane. The variation of the octahedron cells arises from the way the cells fill up the entire space bounded by the airfoil. The planform area is approximately 7.4 m^2 . Since both c and b were rounded to the nearest 100 mm for convenience of model generation, the actual planform area would usually slightly deviate from the proposed planform area.

Figure 2 shows an example of the proposed wing model with skin and internal structure, with taper ratio $\lambda = 0.4$ and washout angle $\alpha_{\text{washout}} = 15^\circ$ introduced. Figure 3 removed the skin and shows the internal octahedron structure only. Notice that the examples are just for demonstrating the capability of design variables of this wing model to amend wing taper and washout angle. In the optimization problem introduced in this paper, only a small washout angle will be introduced. A huge washout angle would make airfoils at some planform locations stall early, not only hindering the aerodynamic efficiency but also falsifies the lifting-line theory aerodynamic predictions.

Before introducing the design variables, it is necessary to introduce and clarify some terminology. Since some design variables are related to several special edges of octahedron cells, it is important to point them out. Figure 4 shows a single octahedron cell $HWXYZK$, with H and K respectively lying on the upper and lower skin. For a rectangular wing, we have $x_x = x_z$ (refer to Figure 1). An octahedron can be understood as two pyramids of identical bases being adhered together. The coordinates of the identical base $WXYZ$ is subjected to change of several design variables which define a curve bounded **inside** the airfoil. We name all such points W , X , Y and Z on each octahedron cell as **midplane vertices**. The curve is defined such that all midplane vertices (x_i, y_i, z_i) satisfies $z_i = h(x_i)$, where $h(x)$ is the curve defined by design variables.

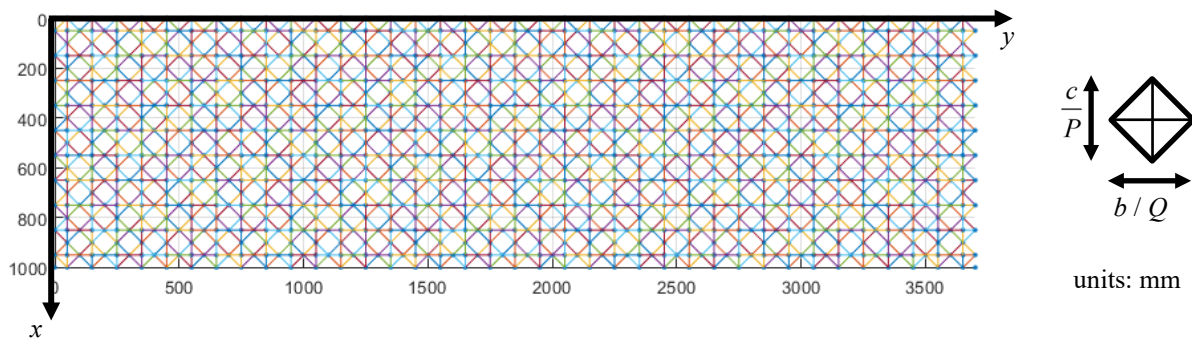


Figure 1: Top View of the Internal Octahedron Structure of a Rectangular Half-Span Wing, $c = 1000 \text{ mm}$, $b = 3700 \text{ mm}$

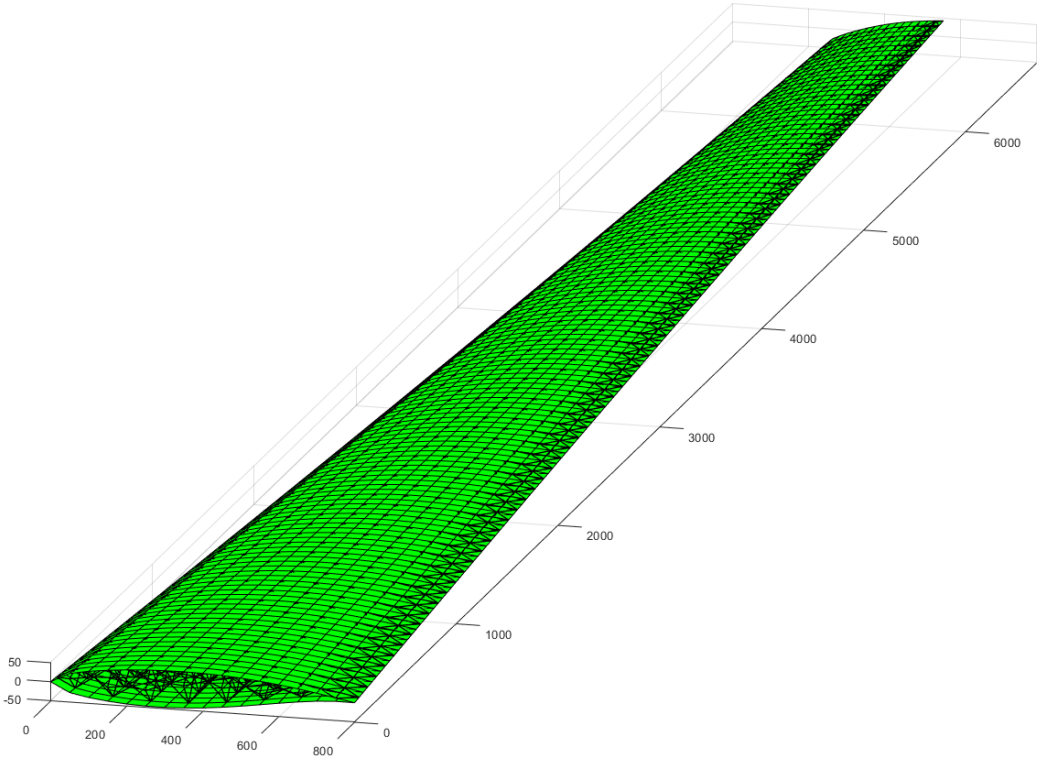


Figure 2: A Design with Skin and Internal Structure, $c = 800$ mm, $b = 6700$ mm, Taper ratio = 0.4, Washout angle = 15°

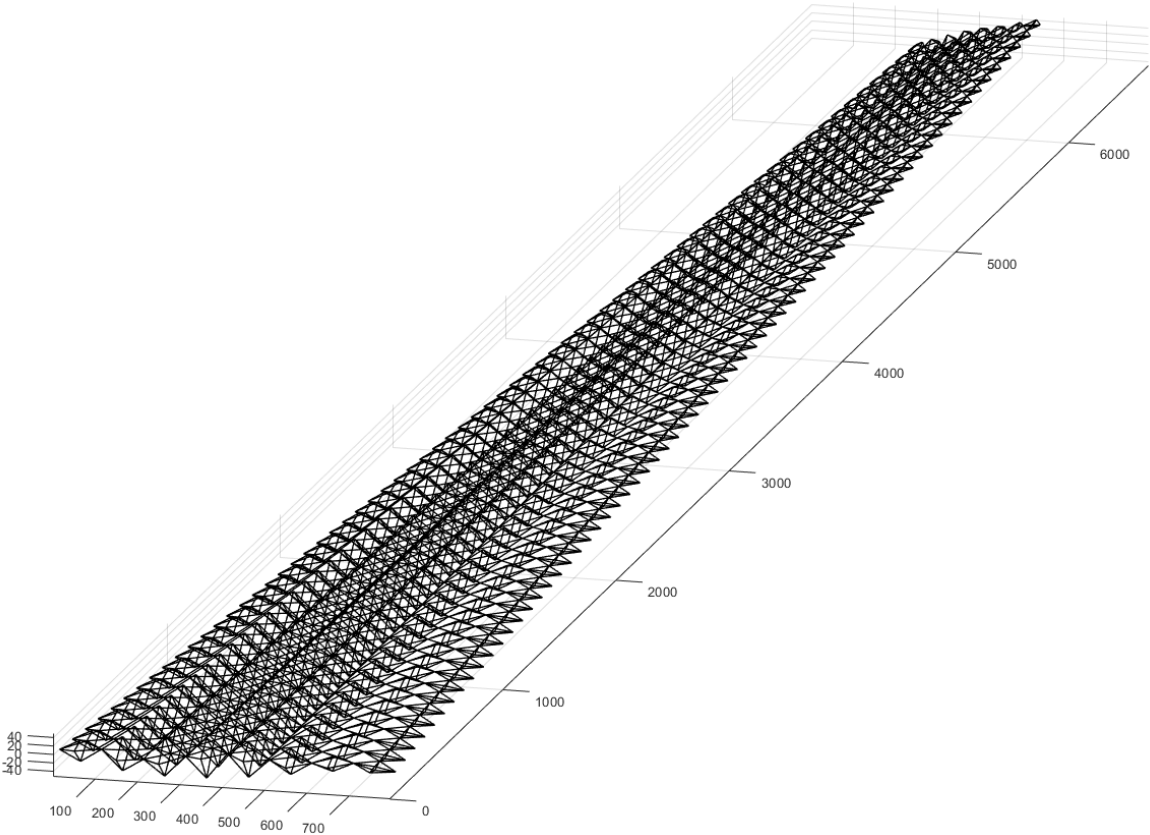


Figure 3: Illustrational Wing Design without Skin - Octahedron Internal Structure Only, $c = 800$ mm, $b = 6700$ mm Taper ratio = 0.4, Washout angle = 15°

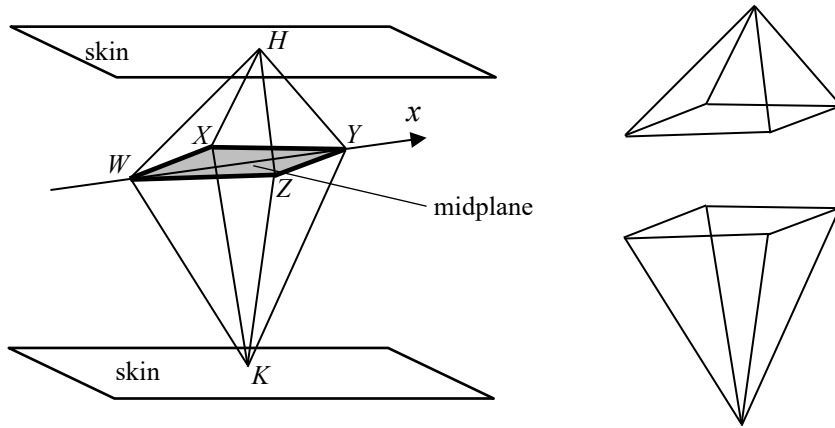


Figure 4: Octahedral Cell Terminology

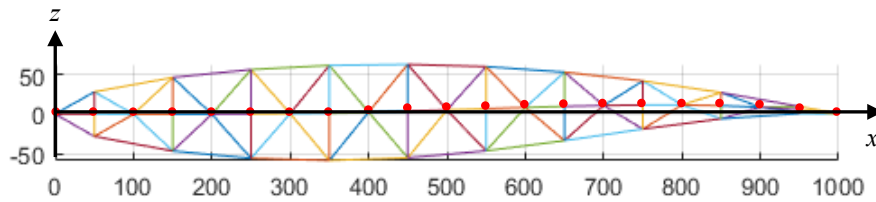


Figure 5: Cross-sectional View of Airfoil, Octahedrons and Midplane Vertices

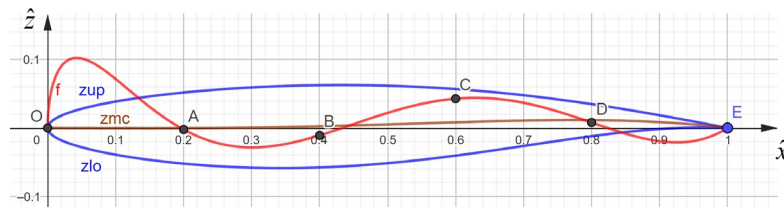

Figure 6: An Example of Unacceptable Curve - Part of the Curve Overshoots the Airfoil Boundary
zmc stands for mean camber line \hat{z}_c

Figure 5 shows the cross-sectional view of an airfoil, the octahedrons and all midplane vertices marked with red dots from a rectangular wing without washout. The curve $z = h(x)$ which satisfies $z_{lo}(x) \leq h(x) \leq z_{up}(x)$ for all $0 \leq x \leq c$ and $0 \leq y \leq b/2$ is defined such that all the midplane vertices are lying on the curve. Figure 6 shows an example of unacceptable curve which is obtained by simply performing curve fitting over several points inside the airfoil. Part of the curve overshoots beyond the airfoil, which results in unrealistic designs.

For simplicity, the airfoil is firstly normalized to a unit chord length, i.e., $c = 1$:

$$\hat{z} = \hat{h}(\hat{x}), 0 \leq \hat{x} \leq 1 \quad (2)$$

First of all, several control points, say A , B , C and D are evenly distributed in chordwise coordinates. The model has been programmed such that users can specify any number of control points they like to adjust the control over the curve. Let $\hat{z}_c(\hat{x}) = (\hat{z}_{up} + \hat{z}_{lo})/2$ be the mean camber line of the airfoil, such that:

$$\hat{t}_{1/2}(\hat{x}) = \hat{z}_{up}(\hat{x}) - \hat{z}_c(\hat{x}) = \hat{z}_c(\hat{x}) - \hat{z}_{lo}(\hat{x}) \quad (3)$$

be the half-thickness of the airfoil. Define $a_1, a_2, a_3, a_4 \in [-1, 1]$ be the design variables assigned to each of the control points A, B, C and D such that the \hat{y} -coordinates of the control points are:

$$[\hat{y}_A \ \hat{y}_B \ \hat{y}_C \ \hat{y}_D] = [a_1 \hat{t}_{1/2}(x_A) \ a_2 \hat{t}_{1/2}(x_B) \ a_3 \hat{t}_{1/2}(x_C) \ a_4 \hat{t}_{1/2}(x_D)] \quad (4)$$

Notice that the \hat{y} -coordinates of the control points in Eq. (9) does not directly control the formation of the curve. However, the \hat{y} -coordinates of the control points somewhat demonstrates how these control points are pulling the curve toward them at different chordwise locations and give variation to the octahedral midplane vertices. Denote the curve passing through $(0, 0)$, (x_A, a_1) , (x_B, a_2) , (x_C, a_3) , (x_D, a_4) and $(1, 0)$ by curve fitting as $\hat{H}_0(\hat{x})$, for example, $[a_1 \ a_2 \ a_3 \ a_4] = [-0.04 \ -0.22 \ 0.71 \ -0.15]$. The degree of polynomial is number of control points plus one. In this example, we are using 4 control points, hence there is a polynomial of degree 5 passing through all six points. With $\max\{|a_1|, |a_2|, |a_3|, |a_4|\} = 0.71$, we have $\max\left\{\left|\min \hat{H}_0(\hat{x})\right|, \left|\max \hat{H}_0(\hat{x})\right|\right\} = 0.85$. We want to bound the range of $\hat{H}_0(\hat{x})$ within $[-1, 1]$ interval. Therefore, the polynomial $\hat{H}_0(\hat{x})$ is scaled such that the maximum absolute value of $\hat{H}_0(\hat{x})$ equals to the maximum absolute value of a_1 through a_4 . The curve after scaling is given by:

$$\hat{H}(\hat{x}) = \frac{\max\{|a_1|, |a_2|, |a_3|, |a_4|\} \hat{H}_0(\hat{x})}{\max\left\{\left|\min \left(\hat{H}_0(\hat{x})\right)\right|, \left|\max \left(\hat{H}_0(\hat{x})\right)\right|\right\}} \quad (5)$$

Finally, the curve $\hat{H}(\hat{x})$ is squeezed into the airfoil without overshooting:

$$\hat{z} = \hat{h}(\hat{x}) = z_c(\hat{x}) + \frac{\hat{H}(\hat{x}) t_{1/2}(\hat{x})}{2} \quad (6)$$

Expanding Eq. (11), we have:

$$\hat{h}(\hat{x}) = \hat{z}_c + \frac{\hat{H}_0(\hat{x}) \max\{|a_1|, |a_2|, |a_3|, |a_4|\} t_{1/2}(\hat{x})}{2 \max\left\{\left|\min \left(\hat{H}_0(\hat{x})\right)\right|, \left|\max \left(\hat{H}_0(\hat{x})\right)\right|\right\}} \quad (7)$$

Notice that due to the scaling in Eq. (10), normally the curve $\hat{z} = \hat{h}(\hat{x})$ does not pass through the control points A, B, C and D because otherwise it requires either of A, B, C and D to be the global maximum or minimum of the curve obtained. However, it is clear that the curve obtained is somewhat “tracing” the control points, or the control points are somewhat “pulling” the curve to their sides. Figure 7 shows the fitted curve obtained from aforementioned example and the corresponding octahedron geometry.

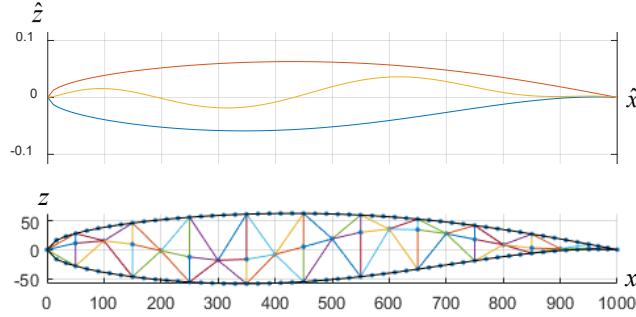


Figure 7: Curve in Airfoil and Corresponding Octahedron Cell Midplane Vertices Positioning,
 $a_1 = -0.04, a_2 = -0.22, a_3 = 0.71, a_4 = -0.15$

The introduction of the design variables so far produces a rectangular wing. That is, we would be able to create the wing model as shown in Figure 1 and Figure 5 by defining grid points and element connectivity. In order to produce wing model with taper and washout angle as shown in Figure 2 and Figure 3, coordinate transform should be applied to all grid points in the model, while the element connectivity could remain still. Consider an arbitrary grid point (x_R^j, y_R^j, z_R^j) defined in a rectangular wing without geometric twist. By introducing taper ratio λ , each grid point undergo the following transform, become (x_T^j, y_T^j, z_T^j) to make the wing model a tapered wing:

$$\begin{cases} x_T^j = \frac{c}{2} - \left(\frac{c}{2} - x_R^j\right) \left(1 - \frac{1-\lambda}{b} y_R^j\right) \\ y_T^j = y_R^j \\ z_T^j = z_m - (z_m - z_R^j) \left(1 - \frac{1-\lambda}{b} y_R^j\right) \end{cases} \quad (8)$$

where

$$z_m = \frac{z_{\max} + z_{\min}}{2} \quad (9)$$

Similarly, the wing washout is introduced by transforming all grid points from tapered grid points (x_T^j, y_T^j, z_T^j) to washout grid points (x_W^j, y_W^j, z_W^j) . The washout grid points are obtained by the following coordinate transform from the tapered grid points:

$$\begin{cases} x_W^j = x_T^j \cos \theta - z_T^j \sin \theta + \left(\frac{c}{2} - \frac{c}{2} \cos \theta + z_m \sin \theta\right) \\ y_W^j = y_T^j \\ z_W^j = x_T^j \sin \theta + z_T^j \cos \theta + \left(z_m - \frac{c}{2} \sin \theta - z_m \cos \theta\right) \end{cases} \quad (10)$$

where

$$\theta = \frac{\pi \alpha_{\text{washout}}}{180^\circ} \cdot \frac{y_T^j}{b} \quad (11)$$

is the angle of rotation in radian.

The geometry of the proposed wing is well defined so far. Next, we will define some physical properties to give the wing weight and stiffness. The proposed FEM model of the wing with single-layer octahedron internal structure is prepared with the MSC NASTRAN input file format. There are two major types of elements being applied, one is the 2-D CQUAD4 plate elements for the skin of the wing; another is the 1-D CBAR element that represents the member of the octahedrons throughout the structure. We believe that the current model setup is adequate for a preliminary evaluation of the optimization problem. Despite of observation of moderate macroscopic static aeroelastic deformation under some high-aspect-ratio (high AR) configurations, it is believed that the deformation of each element in the model is still lying in the linear range due to the rejection criteria (death penalty) set for the material no-yielding requirement in the optimization algorithm. The material properties chosen to run the FEM simulation is that of Al 6061. We assume both the CQUAD4 and CBAR elements for skin and bar are made by Al 6061. We introduce two design variables for CQUAD4 and CBAR elements, namely the thickness of the CQUAD4 plate elements t for all skin elements, as well as the radius of the CBAR elements r , assuming all CBAR elements are circular bars. Notice that since CQUAD4 element is a 2-D element while CBAR is a 1-D element, neither of these elements accurately describes the intersections at each grid point. Although a model constructed with 3D elements such as CHEXA, CPENTA and CTETRA would bring more preferable fidelity, the construction would be far more time-consuming owing to its complexity.

To summarize Section 2, we proposed the optimization problem and defined aerodynamic and structural design variables of the proposed wing model with single-layer octahedron internal structure. There are 11 parameters defining a PARSEC airfoil, in which 9 of them will be used in our optimization problem. We demonstrated an example with 4 design variables that manipulate the control points which defined a curve inside the airfoil to give the midplane vertices variations. There are also 2 more design variables t and r respectively defining the wing skin thickness and the bar elements that constitute the octahedron cells. Other design variables include the chord length c , taper ratio λ and washout angle α_{washout} . If there are P cells in chordwise directions and Q cells in spanwise direction, then the total number of CBAR elements will be $12PQ$, where 12 is the number of bar elements per octahedron cell, PQ is the total number of cells.

3. SIMULATION SETUP

In this section, we will briefly describe the aerodynamic theory we used, then describe how different software packages including the TS-EMO, MSC NASTRAN, UTCart, MATLAB and the model generation code are being integrated to deploy the optimization.

3.1. Aerodynamics: The Lifting Line Theory

The prediction of the lift and drag coefficients of the 3-D wings rely on Prandtl's Lifting Line Theory (LLT) for subsonic incompressible flow. Recall that we have a tapered wing with a washout angle. The aerodynamic properties of airfoils at different spanwise locations need to be separately evaluated. The monoplane equation of a standard four-term solution is given by:

$$\frac{c(\theta)a_0}{4b}\sin\theta[\alpha(\theta)-\alpha_{L=0}(\theta)]=\sum_{n=1,3,5,7}A_n\sin(n\theta)\left[n\frac{c(\theta)a_0}{4b}+\sin\theta\right] \quad (12)$$

In order to evaluate the lift and induced drag coefficients, four monoplane equations need to be set up at different values of θ to solve the four unknowns A_1 , A_3 , A_5 and A_7 . Each value of θ corresponds to a spanwise coordinate. For the standard four-term solution, we select $\theta = m\pi/8$ for $m=1,2,3,4$, representing 0%, 38.3%, 70.7% and 92.4% of the half-spanwise coordinates, as shown in Figure 8.

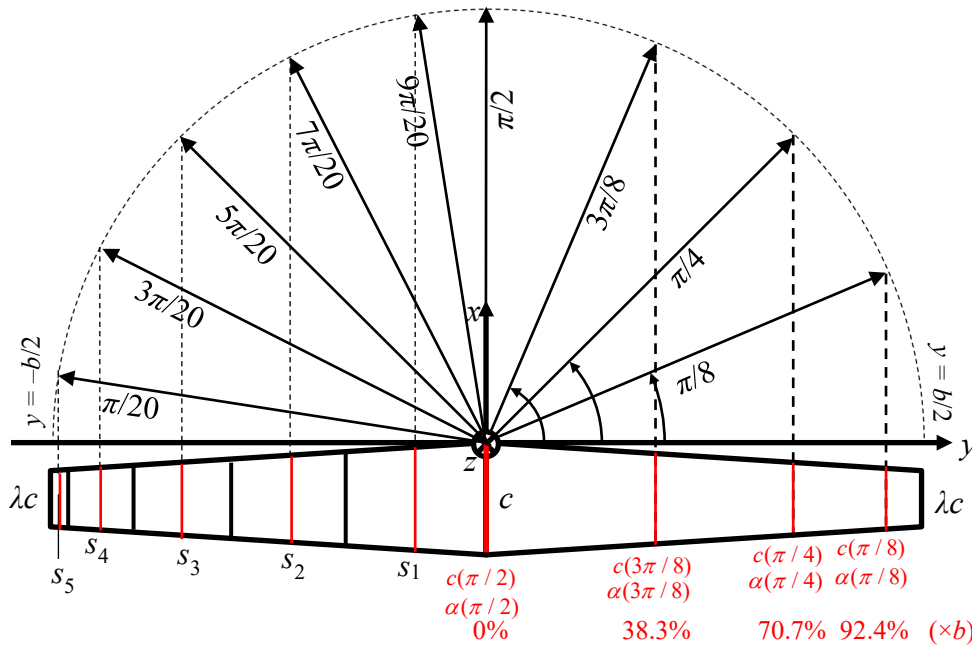


Figure 8: Wing Planform and Spanwise Coordinates where α and c are being evaluated for monoplane equation (right half wing) and parasite drag (left half wing)

The lift coefficient C_L and the induced drag $C_{D,i}$ are respectively:

$$C_L = A_1\pi AR \quad (13)$$

$$C_{D,i} = \frac{C_L^2}{\pi e AR} \quad (14)$$

where

$$e = \left(1 + \sum_{n=3,5,7} n \left(\frac{A_n}{A_1}\right)^2\right)^{-1} \quad (15)$$

is the span efficiency factor. Under steady level flight, the flying car wing would exhibit a static torsional deformation. When calculating the lift and drag coefficients, static aeroelastic torsional deformation need to be considered. We made the following simplifying assumptions:

- The airfoil at each spanwise location of the wing under the normalized coordinate system is the same.

- The torsional warping of the wing is neglectable.
- The impact of bending deformation towards the aerodynamic characteristics is neglected.

So far, the drag obtained from the LLT monoplane equation does not represent the total drag of the 3-D wing. According to Torenbeek [20], the total drag can be decomposed in terms of induced drag and parasite drag, which is suitable for subsonic incompressible flow. The parasite drag is obtained by evaluating the airfoil properties at five spanwise locations with $\theta = n\pi / 20$ for $n = 1, 3, 5, 7, 9$. The parasite drag coefficient is given by:

$$C_{D,p} = \frac{\sum_j s_j C_{d,j}}{S} \quad (16)$$

where $\{s_j\} = \{s_1, s_2, s_3, s_4, s_5\}$ are the five subdivided areas on the wing planform as shown in Figure 8, $C_{d,j}$ is the parasite drag coefficient of the 2-D airfoil evaluated at each spanwise location, and S is the total area of the wing. In another word, the parasite drag of the 3D wing is obtained by weighing the local drag coefficient with the subdivided area, then taking the average across all subdivided areas.

3.2. Optimization Setup – The Objective Function

In order to start the optimization program, the objective function with input and output clearly defined needs to be set up. In simple words, we have 18 input variables, in which 9 of them are related to aerodynamics, 4 of them are related to the internal curve controlling the midplane vertices, and 5 of them are some geometrical properties of the planform and elements. The two outputs are the negative range of the flight and the weight of the wing attached, respectively.

The objective functions are $\text{output}(1) = -\text{Range}$ as well as $\text{output}(2) = \text{Weight}$. By inputting the design variables into the objective function, it will automatically generate the model and input files for NASTRAN to perform aeroelastic analysis. From the preliminary divergence analysis, a first picture of divergence speed as well as the weight and maximum stress in the structure at an angle of attack could be obtained. When the rejection criterion, namely no-yield conditions, and no-aeroelastic-instability condition, are triggered, a death penalty to the magnitude of 10^{99} is assigned such that not only all later calculations are being skipped, but also to make the TS-EMO refrain from searching in vicinity of he rejected points.

Death penalty of different values has been assigned at different stages of rejection. This is for debugging purposes and to enable us to understand better the reasons behind the rejections. For a complete run of the objective function evaluation without being rejected, it is expected that the UTCart 2-D CFD Solver will be called 81 times, while NASTRAN will be called 11 times. This is the number of function calls for *one* evaluation – the total number of calls will be multiplied by the number of function evaluations requested in the TS-EMO program.

4. RESULTS AND DISCUSSION

In this section, we will discuss the results of the optimization problem produced by the TS-EMO program developed by Bradford et. al. [1].

4.1. Solver Setup and Input Parameters

The 18 design variables and their respective lower and upper bounds in the TS-EMO are shown in Table 1. Some important physical properties regarding the flow and material properties are summarized in Table 2. The numerical experiments over the initial design provided us important insights in the programming of the objective function $objfun$. Since the torsional displacement distribution at different angle of attacks are different, it is necessary to run aeroelastic analysis at each angle of attack so that the AOAs, hence the CFD results at different spanwise coordinates would be accurate.

Table 1: Design Variables of Optimization Problem, Meanings, Bounds and an Example Initial Design

Input	Meaning	Units	Lower Bound	Upper Bound	Initial Design
1	p_1 of PARSEC Airfoil Parameterization		0.004	0.010	0.0083
2	p_2 of PARSEC Airfoil Parameterization		0.2	0.6	0.4312
3	p_3 of PARSEC Airfoil Parameterization		0.04	0.08	0.0629
4	p_4 of PARSEC Airfoil Parameterization		-0.5	-0.3	-0.4273
5	p_5 of PARSEC Airfoil Parameterization		0.2	0.6	0.3441
6	p_6 of PARSEC Airfoil Parameterization		-0.08	-0.04	-0.0588
7	p_7 of PARSEC Airfoil Parameterization		0.5	0.8	0.7018
8	p_{10} of PARSEC Airfoil Parameterization		-10	10	-6.86
9	p_{11} of PARSEC Airfoil Parameterization		4	12	8.08
10	a_1 of curve defining midplane vertices		-1	1	-0.04
11	a_2 of curve defining midplane vertices		-1	1	-0.22
12	a_3 of curve defining midplane vertices		-1	1	0.71
13	a_4 of curve defining midplane vertices		-1	1	-0.15
14	c_r , root chord, to be rounded to the nearest 100 mm	mm	600	1000	1000
15	λ , the taper ratio		0.4	1	0.7
16	r , radius of circular bars (octahedron CBAR elements)	mm	5	15	3
17	t , thickness of skin (skin panel CQUAD4 elements)	mm	5	12	4
18	$\alpha_{washout}$, washout angle at the wing tip	°	0	5	5

Table 2: Flow and Material Properties

Flow Properties		Material Properties	
Density	1.225 kg/m ³	Mass Density	2700 kg/m ³
Temperature	297 K	Young's Modulus	69 GPa
Speed of Flow	102 m/s (Ma = 0.3)	Shear Modulus	27 GPa
Dynamic Viscosity	1.812×10 ⁻⁵ Pa s	Yield Stress	276 MPa
Flutter Safety Factor	1.2	Safety factor	1.1

In the Bayesian Optimization, an initial data set needs to be generated such that a surrogate model that guides the search of the global optimum could be obtained. After a surrogate model has been obtained, TS-EMO will search within the design space for a specified number of function evaluations. In this optimization problem, we generated an initial data set that contains 48 function evaluations. In addition, another 148 function evaluations was conducted for searching of the Pareto-optimal solutions. A total of 196 function evaluations was carried out, in which 31 of the function evaluations were assigned death penalties. The optimization program ran for 4 days on a workstation in the laboratory, in which each function evaluation takes around 30 minutes.

4.2. Pareto-Optimal Designs

After the 196 functions evaluations, the TS-EMO selected 4 data points as the Pareto-optimal designs. The number of Pareto-optimal solutions are far less than our initial expectation, probably due to the lack of trade-off relevance caused by the dependence of the objective functions, and the inadequate number of function evaluations. We note that by setting the objective functions in the way defined in Eq. (1), a decreased weight of the wing will increase the range of the flight. Figure 9 shows the distribution of all the results of function evaluations that are not assigned with death penalty. The horizontal axis is the range of the flying car without constant C being multiplied with. It is basically the maximum lift-drag ratio divided by the total weight (in kg) of the vehicle. The vertical axis is the weight of the wing alone in kilogram. It could be observed that the data points are following a general trend that a decrease in weight of the wing would increase the range of flight. The blue dots represents those function evaluations in the initial data set. 48 function evaluations were requested in the initial data set, while 26 of them were not assigned with death penalty. The red crosses are the function evaluations under guided search of the TS-EMO algorithm. 148 functions evaluations were requested, and 139 points are preserved. The four pareto optimal solutions were selected by the program and circled with yellow circles. The aerodynamic calculation covered the angle of attacks ranged from -5° to 11° . The rejection criterion, in particular, the no-yield condition, is assessed at each AOA, hence the structural integrity at each AOA was assured. Since the solutions are being rejected if the maximum stress across all elements exceeds a designated value (yield stress divided by the safety factor, Eq. (1)), all the solution points shown in Figure 9 should be safe to operate over this wide range of AOAs.

The input variables of the four Pareto-optimal results are shown in Table 3, sorted from left to right with respect to the increasing weight. A tradeoff between the weight and the range was observed at the Pareto optimal solutions. By increasing the weight, the range of the flight would be increased. The trend observed on the Pareto optimal solutions is counterintuitive as normally a decrease in the weight of designed wing will make the denominator in the range equation smaller, in term increases the range. However, readers should be reminded that the maximum lift-drag ratio is also a very important factor affecting the range of the vehicle. One possible explanation for the counterintuitive behavior of the Pareto-front is that, as the aerodynamic performance of the wing is being optimized, the structural performance of the airfoil is being deteriorated [21]. In order to maintain the structural integrity of the aerodynamically efficient but structurally vulnerable wing, extra materials should be added. A larger skin thickness decreases the maximum von Mises stress in the elements, while a larger radius of the bar elements (hence, an increased cross-sectional area) decreases the maximum axial stress.

It is extremely difficult to depict the complete cellular structure as it would be look crammed like how Figure 3 looks like. Hence, Figure 10 shows the airfoil and internal octahedron structural design for each of the four Pareto-optimal designs (a), (b), (c) and (d) as listed in Table 3, as well as their respective general shapes. Three of the pareto-optimal designs suggested a taper ratio very close to the lower bound specified in the optimization problem, i.e., 0.4. All the airfoils in the Pareto-optimal solutions somewhat demonstrated curve inflexion in either upper or lower skin.

Table 3: Input Design Variables and Outputs of Pareto Optimal Solutions

Input	(a)	(b)	(c)	(d)
1	0.0081	0.0070	0.0064	0.0081
2	0.5913	0.3089	0.4969	0.6000
3	0.0429	0.0716	0.0536	0.0561
4	-0.3252	-0.4573	-0.5000	-0.4893
5	0.3367	0.4693	0.5244	0.4715
6	-0.0753	-0.0681	-0.0724	-0.0656
7	0.5739	0.7378	0.6832	0.6267
8	-2.8456	5.8427	-9.3151	-8.5611
9	4.6946	4.6740	4.3498	4.3629
10	-0.8940	0.2598	-0.0183	-0.0021
11	0.1599	-0.5522	-0.1721	-0.1952
12	1.0000	-0.0830	-0.8996	-0.8864
13	0.1819	-0.6710	0.5279	0.8767
14	943.7515	823.7797	905.9504	803.5376
15	0.4000	0.8132	0.4328	0.4000
16	5.0548	5.1441	5.6711	5.8661
17	7.0786	7.9029	7.6498	7.3316
18	1.8415	1.7210	0.5433	0.0044
Range	0.0188	0.0189	0.0202	0.0225
Weight	226.4886	244.5417	259.0660	268.3590

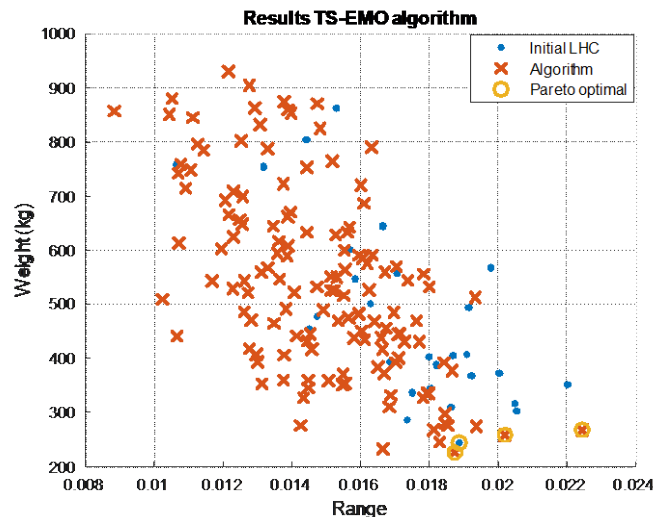


Figure 9: Pareto-Optimal Solutions Obtained by TS-EMO Algorithm

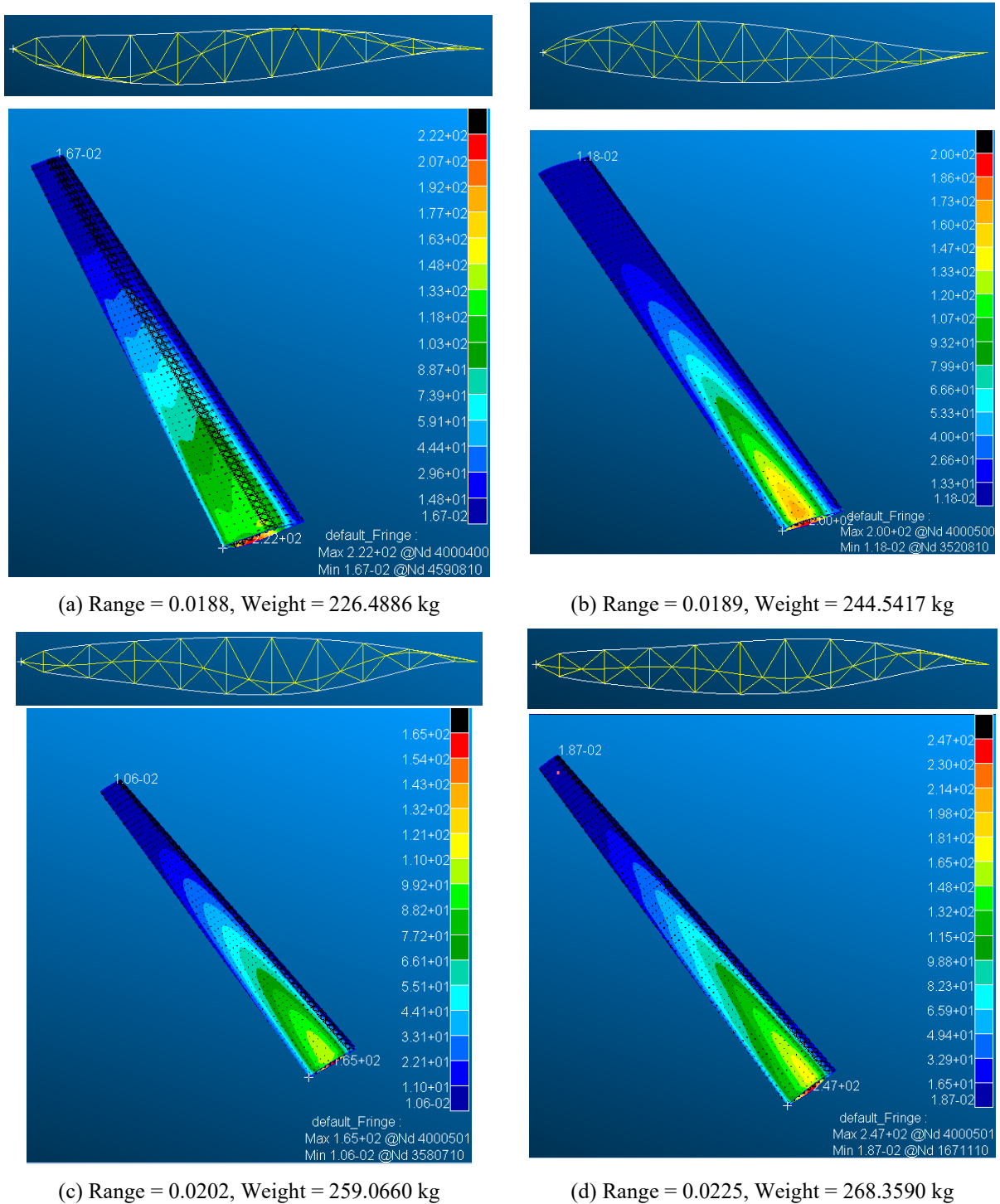


Figure 10: Airfoil and Internal Structural Design of Pareto Optimal Solutions. For each of (a), (b), (c) and (d), the airfoil with internal structure screenshots under rectangular wing configuration without washout has been obtained. The lower figure shows the look of the optimized 3-D look.

Among the four Pareto-optimal solutions, their weights ranged from 220 kg to 270 kg, which would become 17% to 20% of the total weight of the flying car. To the best of the author’s knowledge, there is very limited literature discussing the weight decomposition of UAM products like flying cars, but the weight percentage of the wing is comparable to civil airliners. Since we have no information about the C constant in the range equation so far, it is

not easy for us to provide an accurate estimate of the range of flight. By referring to Table 2 and Figure 15 in Ref [16], we roughly estimate that the SkyDrive flying car with the Pareto-optimal wing design (d), in which $(L/D)_{\max} = 30$, equipped with Li-ion battery of 150 Wh/kg specific capacity, battery occupying 30% of the total weight, would be able to glide a range of approximately 500 km. When Pareto-optimal design (a) is applied to the flying car, the range will be reduced to around 400 km. The range estimate could be greatly increased if there is further breakthrough in Li-ion battery development. The tradeoff between the manufacturing cost of the flying car and the range of flight would be an interesting topic to explore further.

5. CONCLUSION

This paper described the MDO of a wing with single-layer octahedron internal structure specifically designed for a SkyDrive flying car which weighs 1100 kg. We aim to maximize the gliding range of flight of the flying car by altering 18 design variables covering the aerodynamic and structural disciplines. We constructed an 18-input, 2-output MDO objective function code by integrating the aerodynamic, aeroelastic and structural property evaluations of the proposed wing designs combining UTCart, NASTRAN and the model generation code. The objective function is sent to TS-EMO MDO program. Four Pareto-optimal solutions were obtained after 196 function evaluations that cost around 4 days. A rough estimate of gliding range of around 400 km to 500 km has been obtained.

As a preliminary attempt of performing MDO of an unconventional octahedron cellular structured wing with TS-EMO, we believe that the current result somewhat demonstrated the potential of TS-EMO in novel aerostructural design. We believe that our optimization setup could be further improved. Firstly, we noticed that the internal geometry of Pareto-optimal design (a) seems weird. The internal curve governing the midplane vertices is almost sticking to the upper and lower skin. This could be improved by changing the lower and upper bounds of inputs 10 through 13. Secondly, the cellular structure near the trailing edge seems absurdly compressed. We are considering replacing the trailing edge cells with the corrugated panels developed by Yokozeki et. al. [2] to make the design more reasonable. Thirdly, the element properties, namely the thickness of skin and radius of circular bars, are uniform across the structure. Nonuniform element property distribution could be introduced to further reduce the weight. As illustrated in Figure 10, for all Pareto-optimal designs, the maximum stress occurred over a very tiny region near the wing root, mainly on skin elements. The absence of spanwise/chordwise skin thickness/bar radius distribution prevented the design from being further optimized. Furthermore, it is believed that more Pareto optimal solutions could be obtained if the number of function evaluations can be increased.

We believe that TS-EMO functioned satisfactorily in this optimization problem. Bayesian Optimization has high potential in the aerospace industry as a way of exploring novel structural designs. Meanwhile, the proposed cellular structured octahedron wing showed considerable capability to extend the range of flying cars much beyond its original specifications.

ACKNOWLEDGMENTS

The authors acknowledge Japan Society for the Promotion of Science for the financial support of this project (Grant-in-Aid for Scientific Research, 21H01527 and 21H01535).

REFERENCES

- [1] E. Bradford, A. M. Schweidtmann and A. Lapkin, "Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm," *Journal of Global Optimization*, no. 71, pp. 407–438. DOI: 10.1007/s10898-018-0609-2, 2018.
- [2] T. Yokozeki, S. Aya and H. Yoshiyasu, "Development of variable camber morphing airfoil using corrugated structure," *Journal of Aircraft*, vol. 51, no. 3, pp. 1023-1029, 2014.
- [3] E. Chang, L. Y. Matloff, A. K. Stowers and D. Lentink, "Soft biohybrid morphing wings with feathers underactuated by wrist and finger motion," *Science Robotics*, vol. 5, no. 38, 2020.
- [4] D. Li, Z. Shiwei, A. D. Ronch, J. Xiang, J. Drofelnik, Y. Li, L. Zhang, W. Yining, M. Kintscher, H. P. Monner, A. Rudenko, S. Guo, Y. Weilong, J. Kim, S. Storm and R. D. Breuker, "A review of modelling and analysis of morphing wings," *Progress in Aerospace Sciences*, no. 100, pp. 46-62, 2018.
- [5] N. Tsushima, R. Higuchi, H. Arizono and M. Tamayama, "Multi-Scale Aeroelastic Analysis of Wings with Lattice-Based Mechanical Metamaterials," in *AIAA SciTech 2021 Forum*, Virtual Event, 2021.
- [6] B. Jenett, S. Calisch, D. Cellucci, N. Cramer, N. Gershenfeld, S. Swei and K. C. Cheung, "Digital Morphing Wing: Active Wing Shaping Concept Using Composite Lattice-Based Cellular Structures," *Soft Robotics*, vol. 4, no. 1, pp. 33-48, 2017.
- [7] H. Zhang, Z. Zhang, C. Song and C. Yang, "A Morphing Wing with Cellular Structure of Non-uniform Density," *Smart Materials and Structures*, vol. 30, p. 105005, 2021.
- [8] N. B. Cramer, D. W. Cellucci, O. B. Formoso, C. E. Gregg, B. E. Jenett, J. H. Kim, M. Lendraitis, S. S. Swei, G. T. Trinh, K. V. Trinh and K. C. Cheung, "Elastic shape morphing of ultralight structures by programmable assembly," *Smart Materials and Structures*, vol. 28, p. 055006, 2019.
- [9] S. K. Moon, Y. E. Tan, J. Hwang and Y.-J. Yoon, "Application of 3D Printing Technology for Designing Light-weight Unmanned Aerial Vehicle Wing Structures," *International Journal of Precision Engineering and Manufacturing - Green Technology*, vol. 1, no. 3, pp. 223-228, 2014.
- [10] SkyDrive Inc., "Flying Car," [Online]. Available: <https://en.skydrive2020.com/air-mobility>. [Accessed 5 May 2023].
- [11] T. Imamura, Y. Tamaki and M. Harada, "Parallelization of a compressible flow solver (UTCart) on cell-based refinement Cartesian grid with immersed boundary method," in *29th International Conference on Parallel Computational Fluid Dynamics*, Glasgow, Scotland, UK, 2017.
- [12] Y. Tamaki, M. Harada and T. Imamura, "Near-Wall Modification of Spalart–Allmaras Turbulence Model for Immersed Boundary Method," *AIAA Journal*, vol. 55, no. 9, pp. 3027-3039, 2017.
- [13] P. I. Frazier, "arXiv," 10 July 2018. [Online]. Available: <https://arxiv.org/pdf/1807.02811.pdf>.
- [14] A. M. Schweidtmann, A. D. Clayton, N. Holmes, E. Bradford, R. A. Bourne and A. A. Lapkin, "Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives," *Chemical Engineering Journal*, vol.

- 352, pp. 277-282, 2018.
- [15] D. Helmdach, P. Yaseneva, P. K. Heer, A. M. Schweidtmann and A. A. Lapkin, "A Multiobjective Optimization Including Results of Life Cycle Assessment in Developing Biorenewables-Based Processes," *ChemSusChem*, vol. 10, no. 18, pp. 3632-3643, 2017.
- [16] M. Hepperle, "Electric Flight - Potential and Limitations," German Aerospace Center, Braunschweig, Germany, 2012.
- [17] P. D. Vecchia, E. Daniele and E. D'Amato, "An airfoil shape optimization technique coupling PARSEC parameterization and evolutionary algorithm," *Aerospace Science and Technology*, vol. 32, no. 1, pp. 103-110, 2014.
- [18] S. Wu, "PARSEC airfoil polynomial coefficient solver," 2015. [Online]. Available: <https://www.mathworks.com/matlabcentral/fileexchange/50138-parsec-airfoil-polynomial-coefficient-solver>.
- [19] M. S. Khurana, H. Winarto and A. K. Sinha, "Airfoil Optimization by Swarm Algorithm with Mutation and Artificial Neural Networks," in *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, The United States, 2009.
- [20] E. Torenbeek, "Aerodynamic Drag and Its Reduction," in *Advanced Aircraft Design - Conceptual Design, Analysis and Optimization of Subsonic Civil Airplanes*, 2013, pp. 81-91.
- [21] A. Oyama, "0.3. What is Multidisciplinary Optimization," in *Multiobjective Optimization Starting From Zero (Japanese Version), Book 0*, Tokyo, Amazon Kindle, 2018, pp. 53-64.



A PRELIMINARY LOW-FIDELITY MDO APPROACH FOR LOAD ALLEVIATION THROUGH MOVABLES ON HAR WING

Daniel Muradas^{1*}, Sylvie Marquier¹, Joseph Morlier² and Christian Gogu²

1: Loads & Aeroelastics Engineering
Airbus Operations SAS
316 route de Bayonne, Toulouse, France
daniel.muradas-odrizola@airbus.com

2: ICA, ISAE-SUPAERO, MINES ALBI, UPS, INSA, CNRS
Université de Toulouse
Toulouse, France
{joseph.morlier,christian.gogu}@isae-supero.fr

Abstract. *As part of the current efforts targeting further development on cleaner and more efficient aviation, this study proposes a new MDO approach for an optimized High Aspect Ratio (HAR) wing weight using a load alleviation. Load Alleviation Function (LAF) aims at diminishing the peak wing bending moment due to a manoeuvre or a gust by redistributing the lift to a more inboard position on the wing using movables to decrease the overall wing weight. Nowadays, movables are exclusively designed to master the control of the aircraft, and LAF takes advantage of these control surfaces characteristics despite not being designed for it. The challenge of LAF tuning on a HAR will come from the high wing deformation. So this study aims to address the handling qualities (HQ) and LAF aspects in an optimization process of the control surfaces positioning, actuators power, sizes and deflections covering both disciplines, with a wing weight reduction intent. The challenges faced within the project reside in the couplings of the numerous disciplines present in the overall aircraft design: aerodynamics, loads, static aeroelasticity (movable control reversal), HQ, stress, and mass. A highlight will be made on the aerodynamic and loads disciplines, as current MDO publications do not consider a broad number of load cases and flight points with the right coverage of the sub-loads disciplines (such as gust and its aerodynamic specificities), which are crucial in the ranking of different designs, and crucial in the assessment of its feasibility, and finally for certification. This study aims to define the MDO framework by identifying the right compromise between disciplines and fidelities whilst preserving the computational resources. In the first stages, the project focuses on the lower-fidelity approaches through open-source means to verify its feasibility and compile the requirements, targeting to include multi-fidelity approaches at future stages of the research project.*

Keywords: Load Alleviation Function, Multidisciplinary Design Optimization, Multi-fidelity, Handling Quality, Aircraft Control, High Aspect Ratio, Aeroelasticity, Loads

1 INTRODUCTION

In recent years there has been a surge in the interest in HAR aircraft in civil aviation [1]. Thanks to the increase in performance due to the reduction in induced drag. However, this effect is of no use if the wing cannot withstand the load applied, increasing the aspect ratio can only be achieved either by increasing the structural mass or allowing an increase in wing flexibility. The flexibility can be augmented without failure thanks to the use of advanced composite materials, which is already a common practice in recent aircraft designs. The problem arises with the aeroelastic effects that this flexibility implies in the control of the aircraft, as it will be discussed in section 2 and 4.1. The MDO approach to define the control surface concept could be a promising solution to this problem as it is addressing the flexibility effect, handling qualities and the load alleviation opportunity to minimize the weight increase due to the HAR concept. The control surface optimization through MDO is also relevant for classical wing, as sequential computation limits the number of configurations and disciplines that could be considered.

Accordingly, the interest of this study is to develop an MDO framework to optimize the LAF by improving the movables configurations. Some intricacies will arise from the high aspect ratio of the wing, a highly flexible wing will highlight the aeroelastic effects constraining the efficiency of the LAF. To address and illustrate this, two configurations will be assessed to support the comparison of a classical wing configuration versus high aspect ratio.

Load alleviation (LAF) or Maneuver Load Alleviation (MLA) involves the process of structural load mitigation that occurs during flight. The LAF configuration must comply to the European Union Safety Agency (EASA) and the Federal Aviation Administration (FAA) in the United States regulations. A series of specifications for the design and certification of aircraft regarding the maximum loads an aircraft must be able to endure is available in [2] and [3].

Referring now to the sections CS-25.301 Loads: *“Strength requirements are specified in terms of limit loads (the maximum loads to be expected in service) and ultimate loads (limit loads multiplied by prescribed factors of safety). Unless otherwise provided, prescribed loads are limit loads.”* For this case, we will focus on the limit loads, covered as well in section CS-25.321, where is specified the maximum and minimum load factor an aircraft must endure at several flight speeds. The load factor represents the ratio of the aerodynamic component to the weight of the aircraft [2].”

To know on which conditions the limit loads can be achieved during flight, the manoeuvre loads sections CS-25.331 and CS-25.333, specify the design flight speeds and load factors. In section 3.2 further details will be given on the flight envelope the aircraft will be subjected to.

The first load alleviation system was designed for the A320 [4], where it was shown a potential net load alleviation of around 15% at the wing root.

2 LAF/MLA WORKING PRINCIPLE

Knowing the potential benefits of LAF/MLA, it is imperative to know how it works in flight to assess the required disciplines for optimization. LAF in a nutshell intends to create a downward force in the wing to alleviate the increase in loads due to a manoeuvre by deploying the control surfaces. But deploying the control surfaces means a change in the location of the centre of pressure, which can be thought of as the position where the

total lift is applied in an aircraft. This force will generate a moment with respect to the centre of mass of the aircraft which must be counteracted by the Horizontal Tail Plane (HTP) in order to have a stable flight. This presents a challenge, as most of the previous studies that want to tackle aerostructural optimization, only consider the main wing.

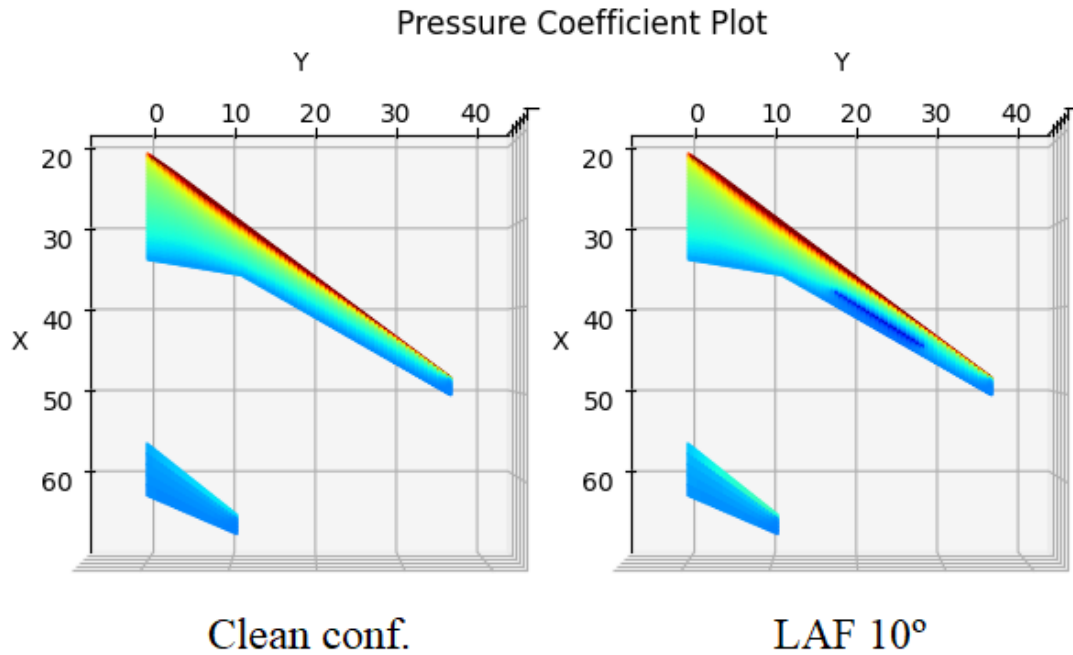


Figure 1: Comparison of pressure distribution between deployed aileron and clean configuration.

In figure 1 it has been plotted the pressure distribution of both configurations, with and without the aileron deflected. This contour plot presents the pressure difference for what would be the upper and lower surface of the wing. The slight decrease in pressure in the region where the aileron is deflected, means a local loss in the lift over the area, alleviating the load at which the wing is subjected.

As it can be seen in 2, there is already a 6.6% improvement in loads with just deploying the default aileron configuration.

3 PROBLEM DESCRIPTION

The objective of this study is to devise a preliminary MDO architecture to attain the best control surface configuration for the least wing bending at the root. The problem could be read as:

$$\begin{aligned}
 & \text{Minimize} && M_x(\text{pos}, \text{size})_i \\
 & \text{w.r.t.} && \text{pos}, \text{size} \\
 & \text{subject to} && F_z(\text{pos}, \text{size}) < \text{Limit Loads}
 \end{aligned} \tag{1}$$

Where $M_x(\text{pos}, \text{size})_i$ is the wing bending moment for a given manoeuvre_{*i*} and F_z is the vertical loads. The wing box should be subjected to an extensive flight envelope for its

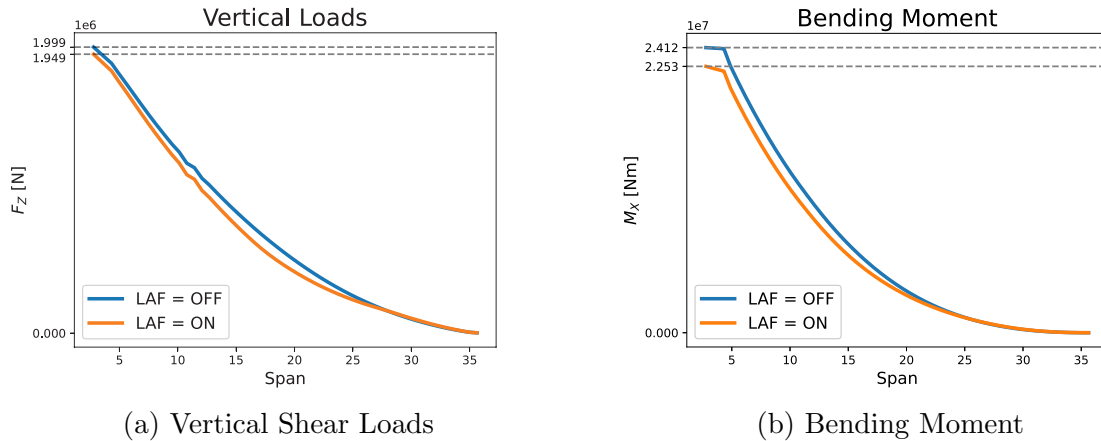


Figure 2: Effect of LAF on vertical load and bending moment (uCRM13.5 [5] with 10° aileron deflection).

correct sizing, for that a broad set of flight points will be evaluated to retrieve the worst scenario, see section 3.2. The variables *position*, *size* refer to the position and size of two adjacent ailerons present in the original design. Being this architecture just a preliminary study, many other constraints must be taken into account for it to reach certification-level requirements, such as Control Reversal (CR), Handling Qualities (HQ), Gust, Horizontal Tail Plane (HTP) Overloading and Failure Conditions. These constraints need to be assessed and this project targets their assessment as separate disciplines in the future (see section 6).

3.1 Model description

As a target for this study, the uCRM9 and uCRM13.5 were selected [5] as they perform as great examples for this test case. The uCRM9, representing a standard long-range aircraft with a wing aspect ratio of 9, will be used as the baseline for the optimization. To validate the results, [6] has a great compilation of the static aeroelastic performance of both uCRM9 and uCRM13.5.

These models were conceived for aerostructural optimization, so there are no specifications for the control surfaces. To have some reference, the model CRM-HL (Common Research Model in High Lift configuration) is practically the same as the uCRM9, and following the design specifications of the technical report of Shmilovich et al.[7] the position and size of the reference model were determined for the uCRM9. For the uCRM13.5 since there is no available information, the same nondimensional aileron size and position from the CRM-HL was used.

Both models share the same centre of mass (presented in yellow in figure 3). The relevant design criteria for the aircraft are depicted in table 1.

3.2 Flight envelope

The definition of the flight envelope is another important step in the optimization, as they will be the test cases for the model. For this reason, real sizing test cases for long-range aircraft are used, shown in table 2. At every iteration, all manoeuvres are evaluated, and the worst case will be the output to minimize, as the manoeuvring sizing

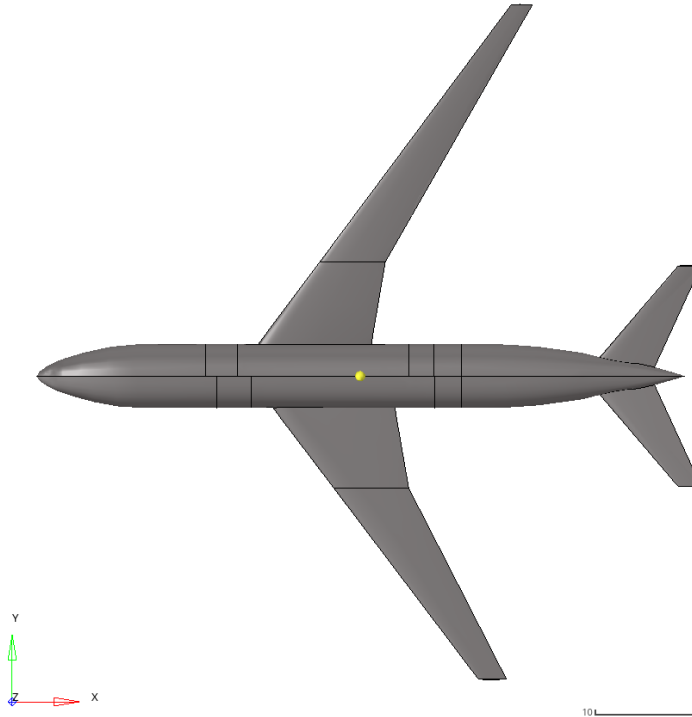


Figure 3: Aircraft model comparison, uCRM13.5 (Top) and uCRM9 (Bottom).

Table 1: Design properties for both aircraft.

Aircraft	uCRM9	uCRM13.5
AR [-]	9	13.5
Span [m]	58.76	72.00
MTOW [ton]	297.5	284.256
Mach Cruise [-]	0.85	0.85
Cruise alt. [kft]	37	37

case may not be the same for different LAF configurations.

In 2 also includes the cruise conditions as a reference. V_C and V_A stand for design cruising and manoeuvring speeds, respectively. n is the load factor.

4 COMPUTATIONAL APPROACH

Given the design criteria and constraints, an aerodynamic and structural model is required for the main Multidisciplinary Analysis (MDA) of the optimization loop.

4.1 MDA and Static Aeroelasticity

As mentioned in section 2, a simple Fluid-Structure Interaction (FSI) with just aerodynamics and structure will not suffice. It is required to take into account the contribution of the tail. The interest of the study will be just static aeroelasticity, so NASTRAN will be selected for its capability of not only solving the FSI but also solving the longitudinal stability problem.

However, it is crucial to understand beforehand how the static aeroelastic analysis

Table 2: Flight envelope.

V_A			
Mach	Alt. [kft]	Q [Pa]	n [-]
0.512	10.0	12783.5	2.5
0.780	25.0	16012.1	2.5
0.780	38.0	8815.7	2.5

V_C			
Mach	Alt. [kft]	Q [Pa]	n [-]
0.85	37.0	10984.4	1.0
0.594	10.0	17206.1	2.5
0.860	26.0	18630.9	2.5
0.860	38.0	10717.4	2.5

is performed. In this problem two disciplines take part: structures and aerodynamics. Both are heavily coupled since the deformation of the wing depends on the load applied, which, in turn, is obtained from the pressure distribution of the aerodynamic surfaces. This circles back to the structural part, as the aerodynamic shape, consequently, will vary depending on the load applied. This loop is what static aeroelasticity aims to solve.

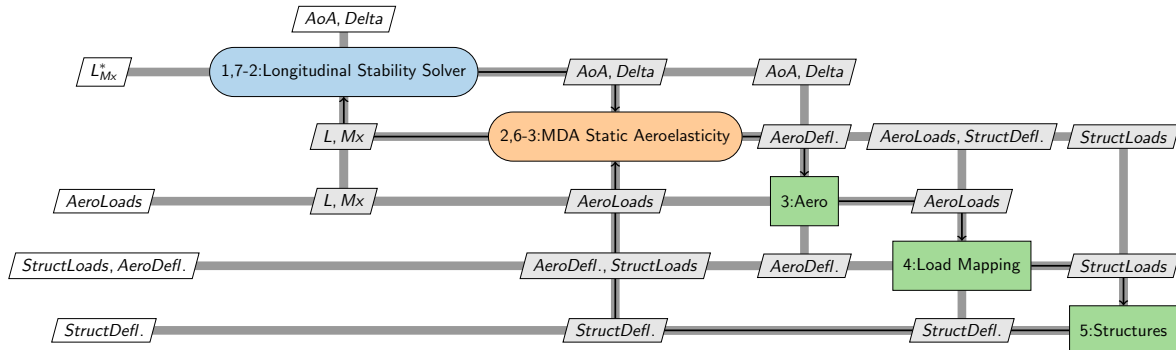


Figure 4: MDA XDSM for a static Aeroelastic problem with a longitudinal stability solver.

As shown in figure 4, the MDA Static Aeroelasticity (Sol 144 in Nastran) not only has to solve the aero-structural problem but on top of that equilibrium must be satisfied. Where the total lift given by the aerodynamic surfaces must equal the weight of the aircraft, as well as having no pitching moment at the Center of Mass (CoM).

4.2 Structural Model

The models uCRM9 and uCRM13.5 [6] NASTRAN Finite Element Model (FEM) are readily available, only for the wingbox. But with some modifications, it is possible to adapt the model to satisfy the problem needs.

First of all, it is imperative to find and specify the CoM of the aircraft. NASTRAN requires as input the centre of mass to be used as a pivot point for longitudinal stability. From the results of [6], it can be assumed that the CoM is placed near the reference point

used to obtain the pitching moment, as it is close to zero. Once this point is known, the centre of mass of the wing box plus fuel is obtained from the FEM. To force a CoM at the desired point, the approach taken was to include a third concentrated mass point as depicted in figure 5, this mass will equilibrate the wing box on the CoM and the remaining aircraft mass will be placed at the CoM. This workaround will suffice for static aeroelasticity, but it will not work for any dynamic analysis.

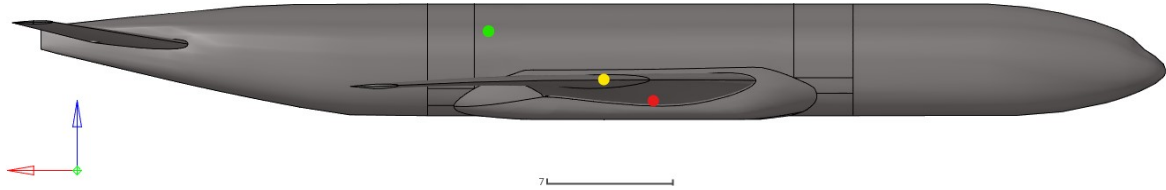


Figure 5: uCRM9 point mass locations, the aircraft CoM (yellow), aircraft trim mass point (green), and wingbox CoM (red).

4.3 Full Aerodynamic Model: Wing and HTP

The aerodynamic model will be in this case the Doublet Lattice Method (DLM) present in NASTRAN. Its simplicity will present a great advantage in the optimization process, as it will rapidly provide aerodynamic loads to the structural discipline reducing considerably the run times, compared to other flow solvers such as Computational Fluid Dynamics (CFD). The DLM model available in NASTRAN will as well account for compressibility effects, via the Prandtl–Glauert transformation [8].

To include the aerodynamics in NASTRAN, the projection of the wing was used as a reference. The mesh will consist of a series of horizontal panels representing idealised wetted surfaces, depicted in figure 6.

The element density of the DLM panels in the wings is much greater than the HTP, as it will be the focus of the optimization. As the aileron is defined as a subset of the panels that make the wing, the precision of the mesh must be high enough to avoid aliasing.

The main issue for using DLM as the aerodynamic solver is the linear behaviour, this means that there will not be any viscous effects that may lead to flow separation, which is the case in manoeuvres with a high load factor as the angle of attack is high as well. Apart from that the aerodynamic forces will not include the contribution due to camber or airfoil thickness.

4.4 Load Mapping

The last discipline in the MDA is the load mapping, often referred to as load and displacement interpolation. Both aerodynamic and structural meshes are essentially different, not only for the different physics involved but also because the nodes for both do not coincide. Within NASTRAN, the splining cards have as its purpose collecting the load applied at a given aero-node and applying it to the correct structural nodes, and, at the same time, retrieving the displacement of each structural node to be applied in the respective aero-node so the aerodynamic mesh can be deformed in parallel to match the structural mesh. Being the splining essentially an interpolation, it is important to always validate the loads introduced, as the total load from the aerodynamics must be equal to the load applied in the structure so that the interpolation is conservative. Many radial

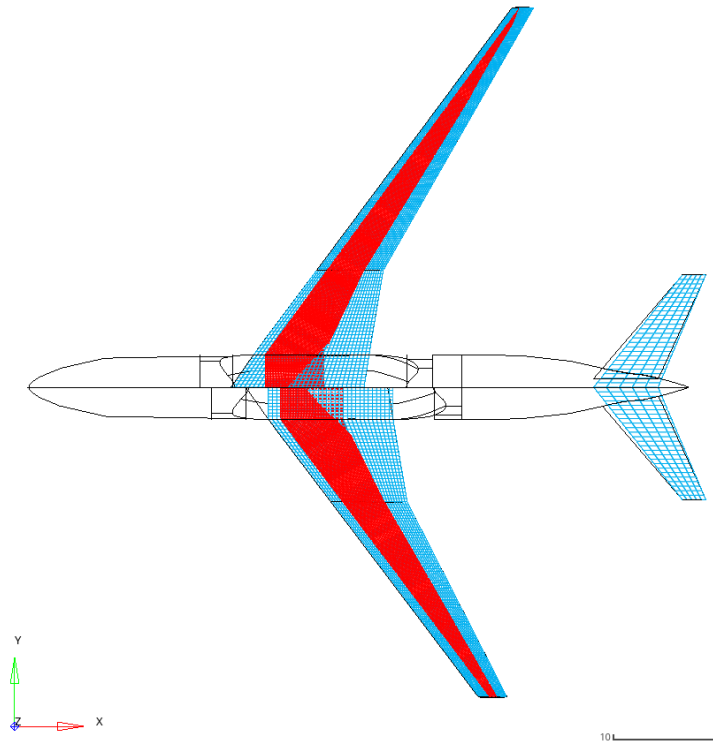


Figure 6: Aircraft FEM (Red) & DLM (Blue) model comparison, uCRM13.5 (Top) and uCRM9 (Bottom).

basis functions for load interpolation are available, some of which are discussed in [9] and will serve as a foundation for accurate load interpolation.

For NASTRAN, many options are available for the splining. SPLINE1 is the preferred solution as it will interpolate the loads in a defined surface [10], this Thin Plate Splines will transfer the load from the aerodynamic panels to the rigid connections to the structure, see figure 7. For this reason, rigid connections from the leading edge to the geometrical centre in between ribs and to the trailing edge were created, aiming at capturing the local twist of the wing and tail. The loads are distributed from the SPLINE1 surface to the structural FEM via radial interpolation through the RBE3 card [10], as shown in figure 7.

4.5 Result Validation

Once the model is ready, using the results from [6] as a reference, knowing that the model is rather low-fidelity it is not expected high accuracy regarding the CFD/CSM results obtained in [6]. The objective is to validate if the results are coherent with the study test cases. Only the maximum deflection of the wing and angle of attack for trim was used, see table 3, where the results from the static aeroelasticity from NASTRAN are compiled under Sol144.

As expected, the displacement results differ from the reference from 26.42% to 39.67%. For the same wing box model, previous studies [11] have shown large discrepancies between low-fidelity and high-fidelity aeroelastic analysis. This is to be expected, not only because of the low fidelity aerodynamics, but it can be also due to the contribution of the tail might be different. Finally, the angle of attack (AoA) results are also different. This is

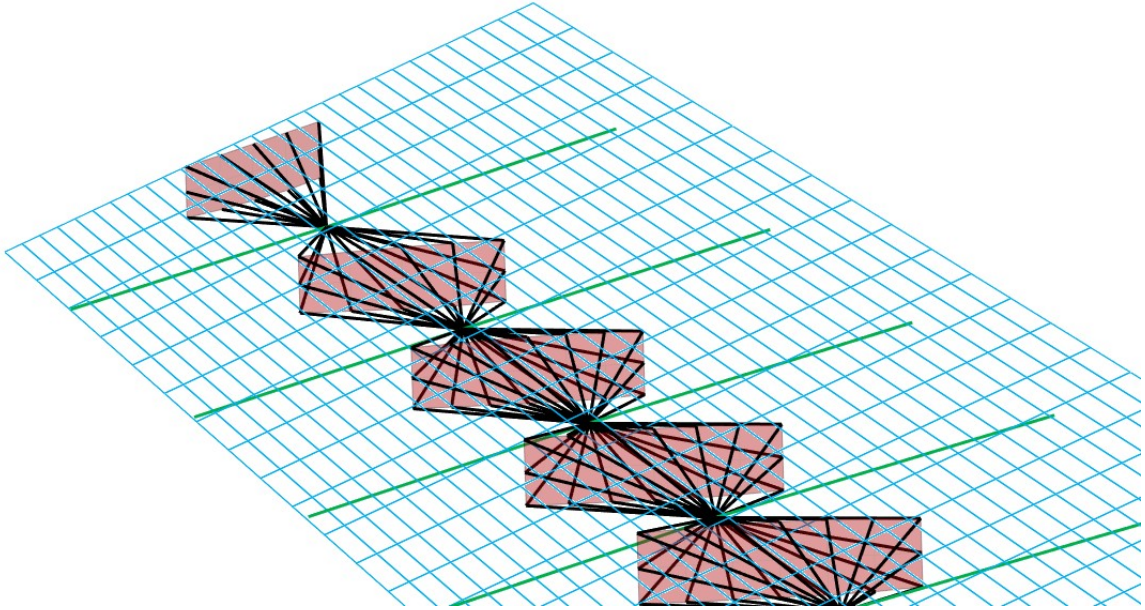


Figure 7: SPLINE1 configuration (green), ribs (red), RBE3 (black) and CAERO1 (blue) entries.

Table 3: Result validation in cruise flight conditions.

uCRM9	Brooks et al.[6]	Sol144
Max Vertical deflection [m]	2.524	1.857
AoA [deg]	2.044	5.389
uCRM13.5	Brooks et al.[6]	Sol144
Max Vertical deflection [m]	6.02	3.632
AoA [deg]	-	5.083

due to the fact that only the wing and the tail are considered.

Nevertheless, the results obtained, compared to reference single-aisle aircraft publications, are representing the right baseline for this analysis. The final deformation can be seen in figure 8.

4.6 Optimization Process

Given that the MDA is currently contained in just one discipline and handled by NASTRAN, the optimization can be achieved simply by introducing an optimizer such as Sequential Least Squares Programming (SLSQP) or Constrained Optimization By Linear Approximation (COBYLA), these optimizers have the advantage that they do not require gradient input, as the sensitivities are computed through the execution. This allows treating NASTRAN as a "black box", which is not the most optimal process, but given the simplicity of the architecture, as a first approach, it will suffice. The eXtended Design Structure Matrix (XDSM) is presented in figure 9.

For the execution of the optimization, GEMSEO [12] was utilized for its simplicity to create and define a multidisciplinary analysis and optimization process.

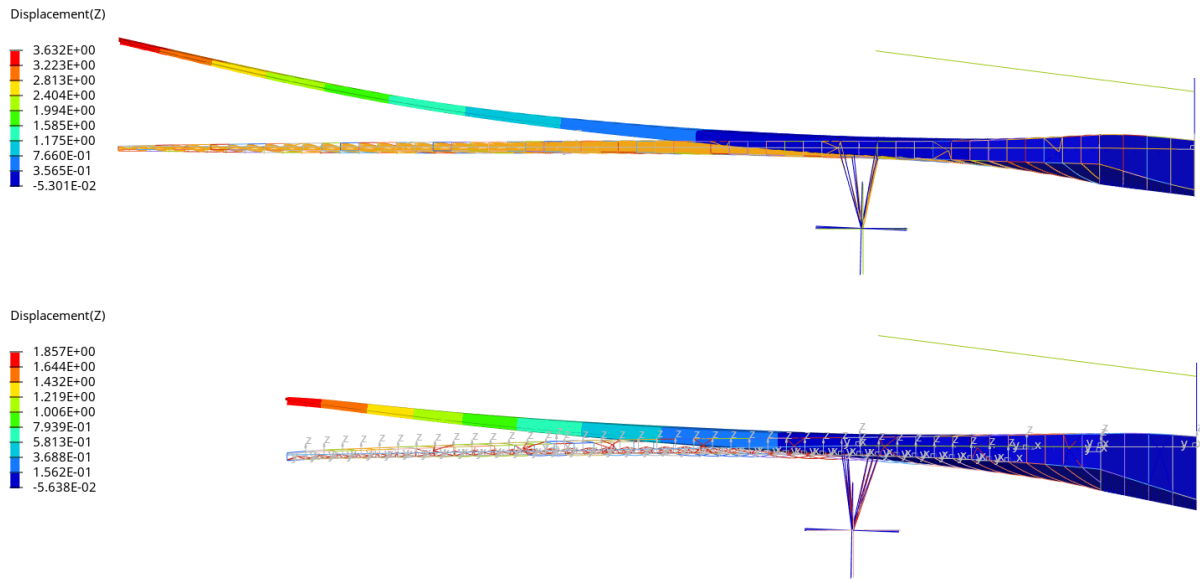


Figure 8: Wingbox cruise deflection comparison, uCRM13.5 (Top) and uCRM9 (Bottom).

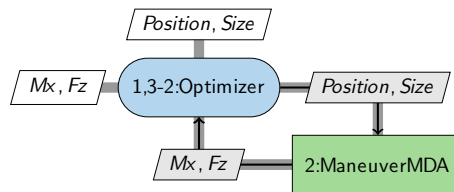


Figure 9: MDAO XDSM for aileron LAF optimization.

5 RESULTS

The first iterations of the optimization had two design variables, the position and size of the two adjacent ailerons. And it comes as no surprise that the optimizer will attempt to reach the maximum size it can for the control surface. The linear aerodynamics present in the DLM solver does not penalize the size of the aileron, so for the final results, a final size of 0.2 was set. It is for this reason as well the deflection angle was not taken into consideration and all the tests were run with 10° deflection. As a design variable for the optimization, the optimizer will always be inclined to the greatest value as no flow separation will appear.

Apart from that, the optimizer has given some interesting insight on what is the preference for the placement of the ailerons. As it can be seen in figure 10, contrary to what it may be most intuitive, the aileron is not placed near the wing tip where the lever arm respect to the root is greatest, but it tends to 60-70% respect to the wingspan. This is due to the fact that near the wingtip the wing box is more flexible, which means that the bending effect will not be as apparent in the root. So for LAF to be effective, it must be closer to the root, finding a compromise between the lever arm and structural flexibility. Less benefit with LAF is anticipated in HAR configuration, the increase in flexibility diminishes the effectiveness of the control surface. And, on top the wing deflection, the aileron induces a torque reducing the local angle of attack, and thus decreasing the intended effect by the aileron deployment. The overall weight impact should be assessed to

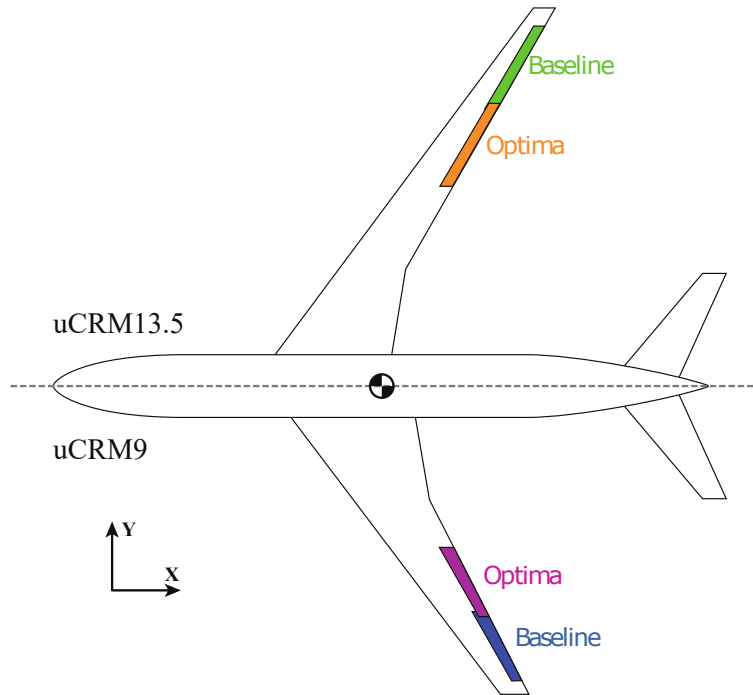


Figure 10: Aileron position optimization, baseline vs optima.

Table 4: Results, nondimensional reference and optimal geometrical aileron placement. Showing the inboard relocation for optimal load alleviation. Maximum bending moment found at $M=0.78$ at 38kft with $n=2.5$ for both cases.

uCRM13.5	Reference	Optimal
Position [-]	0.8	0.624
Size [-]	0.2	0.2
M_X Reduction [%]	-	3%

uCRM9	Reference	Optimal
Position [-]	0.8	0.708
Size [-]	0.2	0.2
M_X Reduction [%]	-	2%

identify the real ranking between HAR wings and the classical configuration (AR9).

Nonetheless, the outcome of this analysis shows an additional 3% benefit on top of the load reduction already present thanks to LAF, this can potentially reduce hundreds of kilograms in the wing structure, and it is targeted as a future part of the project to address the potential weight savings for both wing configurations.

It is worth mentioning that the final placement of the aileron will interfere with the flaps present in the current design. For this study, the optimizer was given the liberty to find the most suitable position in order to get a better insight into the problem. This result also means that it might be interesting in the future to consider as well the use of spoilers, which are closer to the optimal zone for the aileron.

6 NEXT STEPS

As this study is part of a larger project, the next steps to be taken to develop the current MDAO framework by adding other disciplines relevant for feasibility and certification. Firstly, the control reversal effect must be taken into account, by creating a separate discipline that monitors the local lift contribution in the control surface area. Indeed, the capability to control the aircraft is dependant of the control surface efficiency.

Only 2.5g manoeuvres were taken into account in this study, and following the certification specification section 25.335 [2], the aircraft must operate below limit loads under a gust encounter. The control surfaces are susceptible to failure, to fully validate the design it is necessary to validate the loads under failure conditions.

Another important aspect is that ailerons are conceived for the control of the aircraft, if the optimizer does not constrain at some level the effectiveness of the control surfaces to still be able to control the aircraft to fulfil the handling quality perspective.

As mentioned before, the linear aerodynamics present in the current implementation means there is no penalty for high angles of attack and aileron deflection. Fast solutions must be addressed such as [13] and [14], especially for gust computation. And for higher fidelity, finding a nonlinear aerodynamics solver such as CFD or derivatives that is efficient enough to run inside the optimization loop is going to be part of the upcoming challenges. Following previous works like [15], the loads can be interpolated from different aerodynamic flow solvers, which will also present a great opportunity to assess multi-fidelity in the loop.

On top of the flight points, to further enrich the test cases there is an interest to consider different mass configurations with different fuel distributions along the wing which may lead to harsher load conditions.

Finally, it is imperative to address the potential weight savings which ultimately will highlight the true objective of LAF. The MDAO architecture must have, as a final objective, to optimize the wing structure and then be coupled with stress in order to allow the convergence of the aero-loads-stress process.

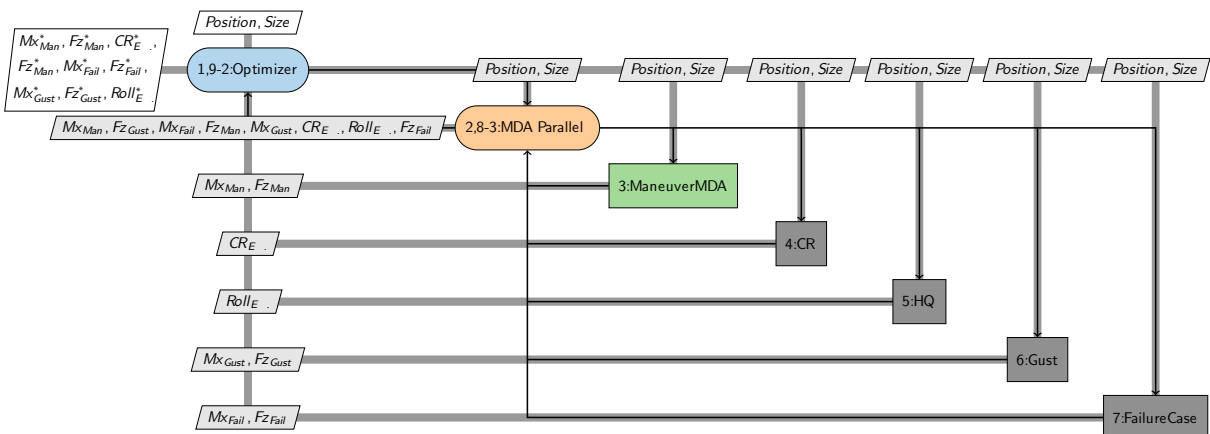


Figure 11: MDAO XDSM for aileron LAF, with future implementations (grey).

In figure 11 the target MDAO XDSM is presented, with the future disciplines to develop in grey color.

7 CONCLUSIONS

This study has demonstrated the LAF/MLA capabilities and how they can be improved by the better distribution of the control surfaces. This MDA process in spite of the fact that it is very simple differs from other wing/wing box optimizations because it requires the HTP, resembling Overall Aircraft Design (OAD). Although it is a very preliminary framework, it has made clear the requirements for the MDA required to perform such optimization, not only for the physics involved but also the requirements that need to be fulfilled as constraints for the optimizer to converge in a feasible design. This study will set the foundation for a greater and more complex framework that will intend to deepen the concept of LAF optimization with greater fidelity and feasibility.

ACKNOWLEDGEMENTS

This project has been funded by the ANRT Cifre program.

Figures containing the uCRM aircraft geometry, mesh plots or FEM results were achieved with Altair Hypermesh [16].

REFERENCES

- [1] G. G. Carrier, G. Arnoult, N. Fabbiane, J.-S. Schotte, C. David, S. Defoort, E. Bernard, and M. Delavenne. *Multidisciplinary analysis and design of strut-braced wing concept for medium range aircraft*. 2022. doi:10.2514/6.2022-0726. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-0726>.
- [2] *Certification Specifications for Large Aeroplanes (CS-25)*. EASA, Cologne, Germany, 2007.
- [3] *Airworthiness Standards: Transport Category Airplanes (CFR-25)*. FAA, 2022.
- [4] *Designing a Load Alleviation System for a Modern Civil Aircraft*, London, England, jul 1986. ICAS.
- [5] T. R. Brooks, G. G. Kenway, and J. R. R. A. Martins. Ucrm: undeflected common research model. 2019. doi:10.17632/gpk4zn73xn.1.
- [6] T. R. Brooks, G. K. W. Kenway, and J. R. R. A. Martins. Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings. *AIAA Journal*, 56(7):2840–2855, July 2018. ISSN 0001-1452, 1533-385X. doi:10.2514/1.J056603. URL <https://arc.aiaa.org/doi/10.2514/1.J056603>.
- [7] Arvin Shmilovich, Matthew Stauffer, Rene Woszidlo, and Paul Vijgen. Low-Speed Performance Enhancement using Localized Active Flow Control: Simulations, Scaling and Design of Localized Active Flow Control on the Common Research Model (4/4). Contractor or Grantee Report 20220006736, NASA, Apr. 2022.
- [8] *MSC Nastran 2021.4 Aeroelastic Analysis User's Guide*. Hexagon, 2021.4 edition, 2021.
- [9] M. Lombardi, N. Parolini, and A. Quarteroni. Radial basis functions for inter-grid interpolation and mesh motion in fsi problems. *Computer Methods*

- in Applied Mechanics and Engineering*, 256:117–131, 2013. ISSN 0045-7825. doi:<https://doi.org/10.1016/j.cma.2012.12.019>. URL <https://www.sciencedirect.com/science/article/pii/S0045782513000029>.
- [10] *MSC Nastran 2021.4 Quick Reference Guide*. Hexagon, 2021.4 edition, 2021.
- [11] Y. L. Lamer, J. Morlier, E. Benard, and P. He. Aeroelastic analysis of high aspect ratio and strut-braced wings. In *33th Congress of the International Council of the Aeronautical Sciences*, page 11 p., Stockholm, Sweden, Sept. 2022. URL <https://hal.science/hal-03888120>.
- [12] F. Gallard, C. Vanaret, D. Guénot, V. Gachelin, R. Lafage, B. Pauwels, P.-J. Barjhoux, and A. Gazaix. Gems: A python library for automation of multidisciplinary design optimization process generation. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018.
- [13] B. Barriety, J.-P. Boin, O. Chandre-Vila, and O. T.Mauermann. Fast fluid-structure computational method taking into account non-linear aerodynamic. In *2019 IFASD International Forum on Aeroelasticity and Structural Dynamics*, 2019.
- [14] O. C. Vila, Y. Nivet, J. Morlier, and N. Gourdain. *Fast Method For Dynamic Fluid-Structure Interactions Considering Unsteady Aerodynamics*. 2022. doi:10.2514/6.2022-2538. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-2538>.
- [15] J. Mas Colomer. mid2supaero/aerostructures. <https://github.com/mid2SUPAERO/aerostructures>, 2019.
- [16] *CAE Altair HyperWorks*. Altair Engineering, Inc., Troy MI, United States, 2021.2 edition, 2021. URL <https://altairhyperworks.com/product/HyperMesh>.



AN ARBITRARY LAGRANGIAN-EULARIAN ALGORITHM TO SOLVE COMPRESSIBLE FLOW PROBLEMS: HEMLAB ALE

Erol Aksoy^{1*}, Mehmet Sahin²

1: Aeronautical Engineering
Faculty of Aeronautics and Astronautics
Istanbul Technical University
aksoyer16@itu.edu.tr

2: Aeronautical Engineering
Faculty of Aeronautics and Astronautics
Istanbul Technical University
msahin@itu.edu.tr

Abstract. *Capabilities of In-house HEMLAB solver is extended by implementing an ALE approach to solve rigid grid motion and is employed to asses the hover performance of S-76 main rotor. In the study, firstly, we verified our algorithm on baseline cases proposed under HiOCFD4 workshop. Then, we performed Euler simulations of S-76 main rotor with swept-tapered tip. Since the rotor problems requires high resolution grids, adaptive mesh refinement is exploited in the study. pyAMG anisotropic metric mesh adaptation library was integrated to HEMLAB previously to do mesh refinement. In the study the capabilities of HEMLAB solver is further tested on such a complex flow physics.*

Keywords: FVM, ALE, Rotor simulation, Adaptive Mesh Refinement (AMR)

1 INTRODUCTION

Moving boundary problems attracted researchers over years since most of the complex phenomenons in nature include moving boundaries coupled with their surroundings. There are several methodologies to overcome difficulties emerging from movement of body. A very accurate method to solve the problem is based on Arbitrary-Lagrangian-Eulerian (ALE) formulation. Exploiting the ALE that coalesce the advantages of Lagrangian and Eulerian approaches, large distortions on the mesh can be handled while preserving the precise shape of interface and high-order solutions on deforming meshes. In this context, in-house HEMLAB algorithm is enhanced with the capability of solving moving mesh problems via ALE. The numerical algorithm is initially validated on heaving and pitching/heaving airfoil problem which is proposed in High-Order CFD Workshop 4 (HiOCFD4). Then, the algorithm is applied to solve flow around S-76 main rotor which is provided under Hover Prediction Workshop (HPW).

Rotor analysis still contains some difficulties despite of the improvements in CFD and computer technology. These difficulties simply rely on the vortex formation at the proximity of the rotor and its interaction with other blades, highly complex geometries of the rotor, strong vibrations and shock waves [1]. The flow around a rotating object also includes sharp velocity gradients and strong vorticities. For instance, due to this strong vorticity, the turbulent simulation of a rotating flow needs a special consideration. To overcome this issue, some researchers divided domain into two and the outer domain is solved using Euler equations to preserve the vortices because vortices can be easily diffused by viscosity. Wake breakdown phenomena is also a problem to resolve and understand [2]. The flow physics may even be more complex in forward flight because of dynamic change of pitch angle and, because, vibration effect becomes more significant. In addition, It looks like the mesh structure is highly effective on the CFD solutions [3]. For accurate predictions of performance parameters, excessive numbers of grids are needed. The mentioned difficulties force us to use high fidelity CFD tools instead of blade element theory, blade element momentum theory, free and prescribed Wake methods. The current focus of the researchers are on developing reliable CFD tools and algorithms to simulate rotor problems. However, they seems to be still very expensive. Adaptive mesh refinement technique offers an opportunity to get rid of some of the unnecessary expenses.

In the vision of overcoming these issues with the researchers all over the world, some benchmark problems are prepared under the AIAA Hover Prediction Workshop (HPW). The current study focuses on solution of flow around S-76 main rotor geometry with swept-tapered tip, which is one of the problems presented in HPW. HEMLAB solver is empowered with an Arbitrary-Lagrangian-Eulerian(ALE) technique to solve the rotor problem. pyAMG library is employed to do anisotropic mesh refinement. By doing so, the expense of the problem will be reduced significantly. The previous study in Ref. [4] showed that an-isotropic mesh adaptation may become very beneficent on the current problem.

2 METHODOLOGY

HEMLAB solver [5] uses a vertex based median-dual finite volume approach. It employs an efficient edge based data structure. The data structure was extended from an existing 2D data structure to 3D. It utilises pointers to reach a specific data lying on the facet-edge data types belonging each edge of a face of each volume. Using the pointers

org, *dest*, *sym* and *rot*, whole domain can be swept starting from an arbitrary point. Due to this unique data structure, spatial locality in physical memory is achieved. Therefore, the algorithm works cache-efficiently. HEMLAB can solve Euler and RANS equations. For turbulent flows, it utilises one equation negative Spalart-Allmaras model [6]. Implicit solutions are obtained by employing Newton's sub-iterations. Fully coupled formulation is used for Spalart-Allmaras turbulence model in Newton's method. PETSc [7] library is exploited to solve implicit equation system with FGMRES Krylov subspace algorithm. Block matrices are exploited for the evaluation of jacobian matrix to improve its computational performance. Time dependent simulations are conducted by first order dual time-stepping approach [8]. HEMLAB solver can employ HLLC [9] where the wave speed estimate is as proposed by Batten et. al. [10] and Roe's flux difference scheme [11] to discretize convective terms. The second order accuracy in spatial discretizations are achieved by using gradients computed at nodes. The gradients for flux reconstruction are calculated by unweighted second-order least square approach [12] and left and right state can be obtained using Taylor series expansion with second order accuracy. On the other hand, Green-Gauss gradients are computed at control volume faces for viscous fluxes.

Following the recent advancements in the adaptive mesh refinement techniques, pyAMG anisotropic mesh adaptation library is implemented into HEMLAB solver. The adaptive mesh refinement technique provided by pyAMG [13, 14] relies on a feature based error calculation based on Hessian matrix of a given solution parameter such as Mach, density etc. The software controls the L_p norm of interpolation error. Minimization of the interpolation error is achieved by Hessian matrix that is scaled by its determinant. This term represents a metric in Riemannian manifold. The aim of the calculation is to find a unit length edges and unit volume elements with respect to the metric. This is achieved by doing an diagonalization in orthogonal frame. This methodology employed by pyAMG also takes the direction of edges into account. The parameter determining the Hessian matrix is called *sensor*. Theoretically, the sensor may take any parameter. However, it actually depends on the physics that is desired to capture. For instance, it is more reliable to use Mach sensor to capture shocks waves and one may use summed combination of the parameters [15]. As stated by Fidkowski and Roe [16], entropy variable can become a general-purpose sensor variable. In this respect, the previous study [4] and current one attempt to utilise Mach and entropy variable as sensor functions. Mach sensor function is

$$\text{Mach Sensor} = M \times e^{-0.1r} + 0.5$$

whereas the entropy sensor is

$$\text{Entropy Sensor} = e^{-0.1r} \times S + 0.5$$

$$S = \frac{R_g}{\gamma - 1} \ln \frac{T}{T_{inf}} - R_g \ln \frac{\rho}{\rho_\infty}$$

where M is Mach number, r is the distance from wall and S is entropy. The R_g and γ terms represent gas constant and specific heat capacity ratio where T and ρ corresponds to temperature and density. The exponential term with r is added to avoid excessively large elements at the farfield.

HEMLAB code was originally developed for fixed mesh problems. In the scope of this study, an Arbitrary-Lagrangian-Eulerian(ALE) approach is implemented to solve

moving mesh problems. For now, only rigid motion is considered to solve the rotor problem. Implementing ALE kinematics, the only difference from the Eulerian form is that an additional convective term (flux due to grid motion) appears due to the grid velocity. The term brings a term to diagonal of Jacobian of convective fluxes and it just shifts the eigenvalues of the jacobian whereas the eigenvectors remains unchanged. In order to preserve uniform flow solutions in the ALE approach, the grid velocities needs a special treatment. The numerical discretizations had better to satisfy Discrete Geometric Conservation Law (DGCL). Farhat and co-workers showed that DGCL is at least sufficient condition for non-linear stability of an ALE formulation [17]. However, since the authors' previous study [4] has showed that the solution in question can be obtained without DGCL, It is not implemented in HEMLAB for now. For rotor problem, an angular velocity prescribed specifically for the problem and rigid mesh motion is determined based on this velocity. Using the angular velocity and location vector, grid velocity can be easily obtained at any point. The grid velocity are calculated at the mid points of internal edges. For the boundary surfaces, they are calculated on centroids of each facet belonging to points on an edge. By this way, second order accuracy is preserved on boundary as well. The boundary conditions are imposed as weak-Riemann boundary condition using Roe flux difference scheme, assigning right state as prescribed farfield boundary condition, on the farfield. Weak-prescribed boundary condition is imposed on slip wall whereas strong boundary condition is implied on no-slip wall on the rotor surface.

3 APPLICATION ON FLOW AROUND HEAVING/PITCHING AIRFOIL

Resolving flow physics over moving bodies with high order accuracy is a challenging goal. A workshop, High-Order CFD Workshop (HiOCFD), was established to test the behaviours of different codes under challenging cases. They determined several cases to asses behaviours of code on moving mesh problems. One of them was determined for 4th meeting of the workshop (HiOCFD4). In the case, flow around NACA0012 airfoil in heaving and pitching motion was examined. In the scope of this study, we assesed the same cases, which are named as case 1 and case 2 in BL3 test case (heaving and pitching airfoil) in HiOCFD4 [18], and compared the results with the results from HiOCFD4 [19].

The case 1 assesses pure heaving motion with a prescribed function of grid motion. In HiOCFD4, researchers used high order methods such as Discontinuos-Galerkin. The distance of farfield is 100 chord from the body in these simulations. Even if HEMLAB is second-order accurate and farfield is 10 chord above from the body, the solution has a good conformity with the codes in HiOCFD4 as seen in Fig. 3. In addition, flow field can be seen in Fig. 1. Based on the solutions of Wukie et al. [20], very similar patterns are captured in vorticity contours.

Case 2 combines pitching and heaving motion in order to align flow. There are differences between the codes in HiOCFD and HEMLAB, however it might be a consequence of that farfield is close to the body. Flow field can be seen in Fig. 2.

4 APPLICATIONS ON FLOW AROUND S76 MAIN ROTOR

Solution of flow field around S76 main rotor was proposed under AIAA Hover Prediction workshop. In this study, swept-tapered configuration is investigated. The rotor has 4 blades with main characteristics of each blade given in Tab.1. Rotor has 3 different airfoils along the span: SC1013R8, SC1095R8 and SC1095. Taper ratio and sweep angle

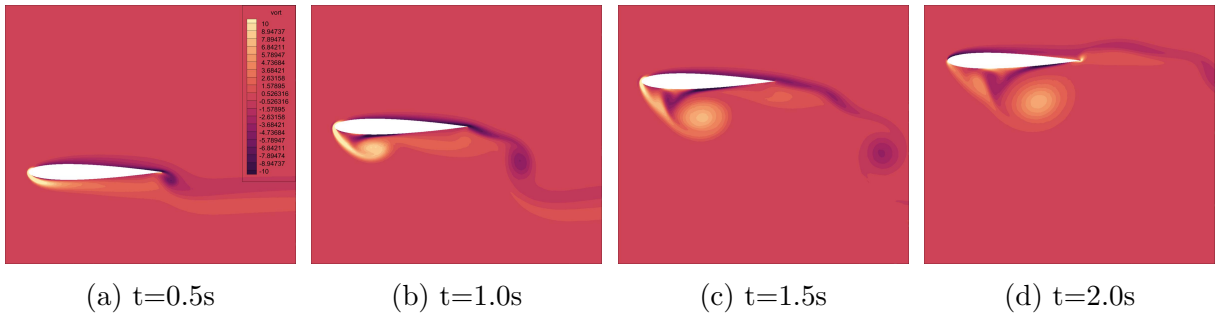


Figure 1: HiOCFD4 heaving airfoil (case 1) case vorticity contours over time.

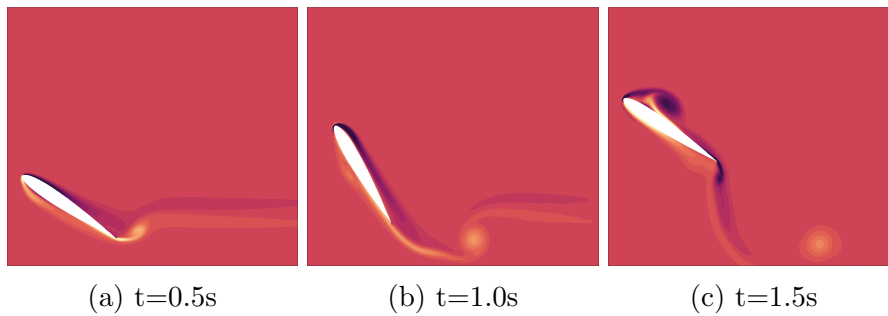


Figure 2: HiOCFD4 heaving-pitching airfoil case (case 2) vorticity contours over time.

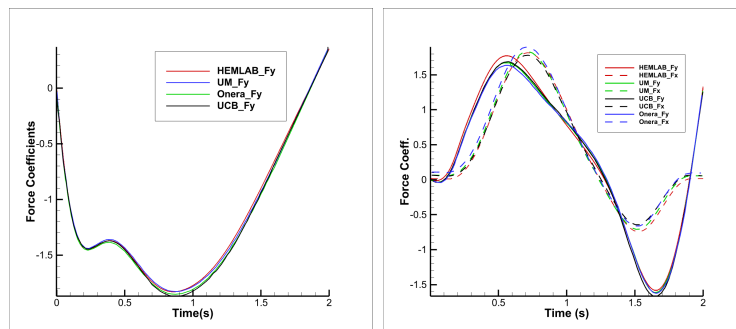


Figure 3: Performance parameter comparison for case 1 (left) and case 2 (right) .

of tip region are also shown in Tab.1. The geometry of rotor blade and twist distributions along the span are drawn in representative figure 4.

Table 1: Characteristics of the S-76 main rotor blade

Parameters	Values
Number of blades	4
Rotor solidity	0.07043
Cut-out location(%R)	19
Tip location(%R)	95
Tip taper Ratio(%)	60
Tip sweep(LE)	35
Radius(m)	6.704
Ref. chord(m)	0.371

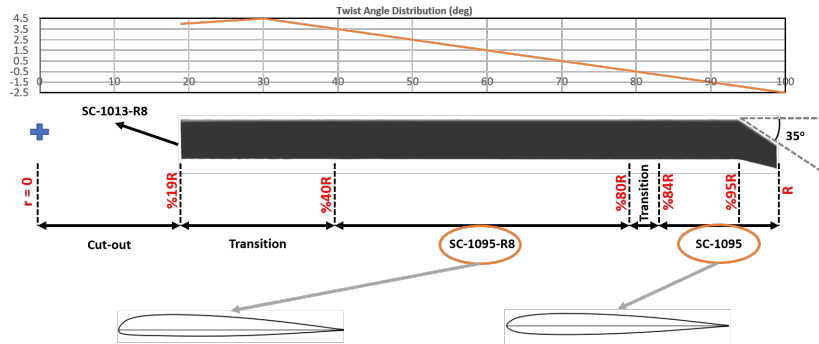


Figure 4: S76 main rotor geometry.

4.1 Computational Grids

The grid structure directly effects the results. As previously said, this effect can become obvious in such a complex flow analysis. However, the current analyses utilise anisotropic mesh adaptation. Therefore, high accuracy results can be achieved, creating relatively simple mesh topologies in the initial phase.

4.1.1 Blade Surface Mesh

Unstructured blade surface mesh is shown in Fig. 5. 235 points are used along the span where the number of chord-wise vertices is 59 at the root and 44 at the tip. Span-wise points are aligned more often at the root and tip to capture the tip and root vortices. Due to the high twist rate and different shapes across the span, surface capture is hard to achieve with unstructured grids without using extreme amount of nodes. Therefore, we created a boundary-layer like surface meshes at the leading and trailing edges. These are extended along the chord. The surface mesh is shown in Fig. 5.

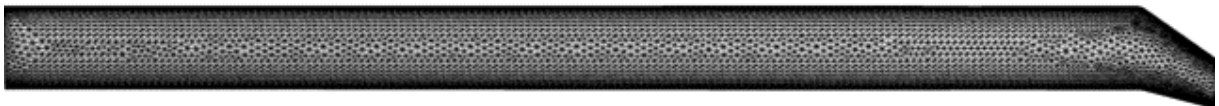


Figure 5: Blade surface mesh of the S76 main rotor.

4.1.2 Volume Mesh

The grid topology can be seen in Fig. 6. The spacings are enlarged step by step as moving away from the body. The near-body zone has a cylindrical shape with $1.3R$ height and $2R$ diameter. In this zone, a uniform spacing as $0.015R$ is used. The off-body zone with the spacing of $0.2R$ also indicates a cylindrical shape. However, in this time, the height of the cylinder is $5.4R$ while the diameter is $6R$. And the spacings at the farfield have the value of R where the farfield cylinder elongates $40R$ in longitudinal direction and $40R$ in radial direction. As a result, the total vertex number inside the computational volume becomes 1,043,679.

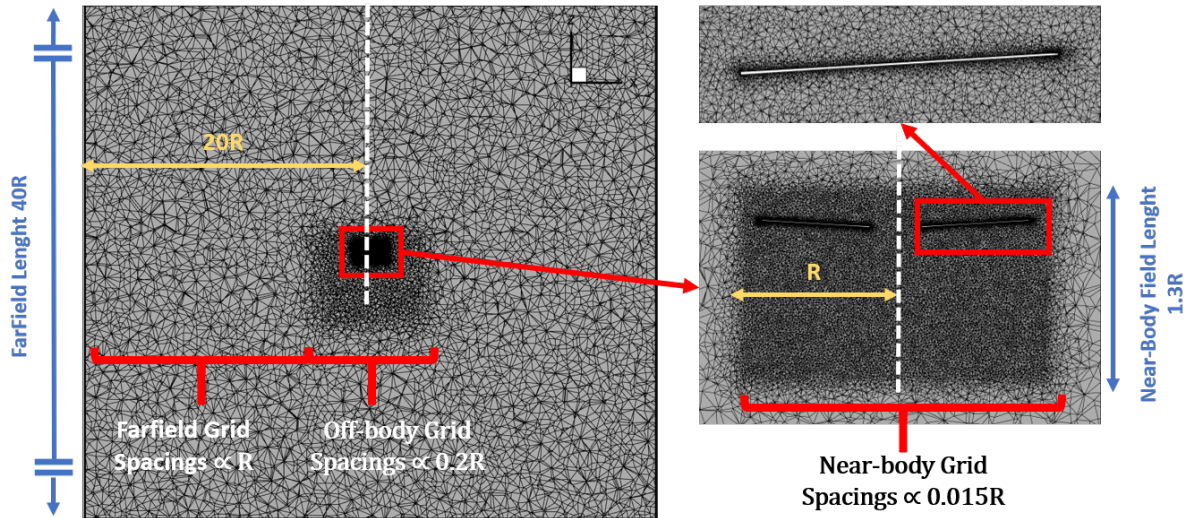


Figure 6: Computational domain of the Euler simulations.

4.2 Analysis Conditions

The main rotor of S-76 helicopter is practised with 3.5 deg cone angle and 9 deg collective pitch angle. The analysis was performed when tip Mach number is 0.65 and Reynolds number is 1.18 million based on reference chord length. The analysis conditions can be summarized at Tab.2.

Table 2: Hover conditions

Parameters	Value
Reynolds number	1.18
Tip Mach number	0.65
Cone angle	3.5°
Collective angle	9°

Table 3: Simulation conditions for cases.

Parameters	Simulation 1	Simulation 2	Simulation 3
Sensor Function	Mach	Mach	Entropy
Flux Scheme	Roe	HLLC	Roe/HLLC
Solved equation	Euler	Euler	Euler
Time integration scheme	1st Order Dual-Time Stepping		
Baseline solution	10 Revolution		

Three different simulations were conducted in total. These are shown at Tab.3. In all of them, Euler equations are solved. First two simulations are comparing flux schemes Roe and HLLC. The third simulation assess the entropy sensor function. It must be noted that a baseline solution was applied for 10 revolution before the refinement and, then, refinement was applied at each revolution. Therefore, adaptation sizes in Tab.3 indicates only the refinements starting from 11th revolution. The baseline solution also will be

examined in detail in the following sections.

4.3 Baseline Solution

The baseline solution corresponds to first 10 revolution (10000 time step). We did not do any refinement during the baseline solution. In previous works, the effect of early refinement was observed and stated. So that, the rotor was rotated for 10 revolutions using baseline mesh. Fig. 7 shows the vortex structures. Initially developed vortex ring moves along the wake and then disappears where the off-body grid starts.

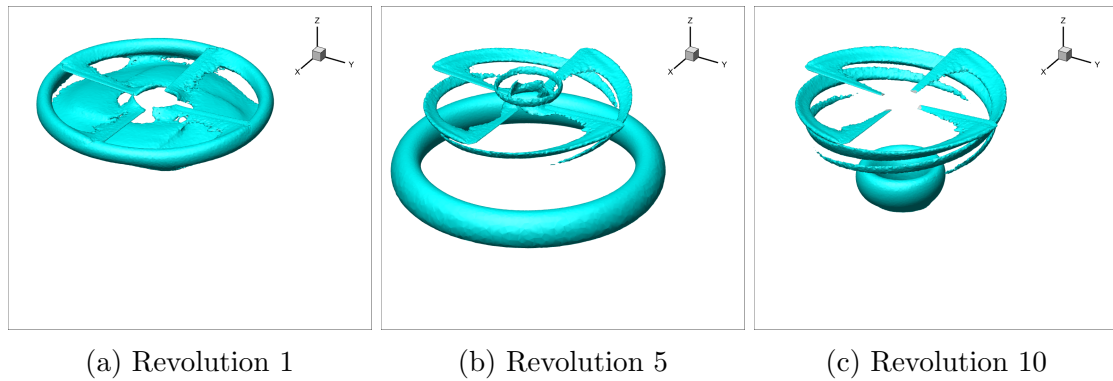


Figure 7: Baseline solution vortex structures.

4.4 Simulation Results

The first simulation was done using Roe flux scheme. Mach sensor function is used for mesh refinement process. For the first 4 refinement level, the complexity (adaptation size) of 400000 is used. Then, the next 4 refinement level is done by using the complexity of 800000 whereas last two level is 2400000. At the last iteration, the number of nodes was 8,249,524. As expected, tip vortices are captured by Mach sensor, however wake region is not refined. The vortex structures and wake layer can be seen in Fig. 8. In addition, resulted mesh structures are shown in Fig. 10.

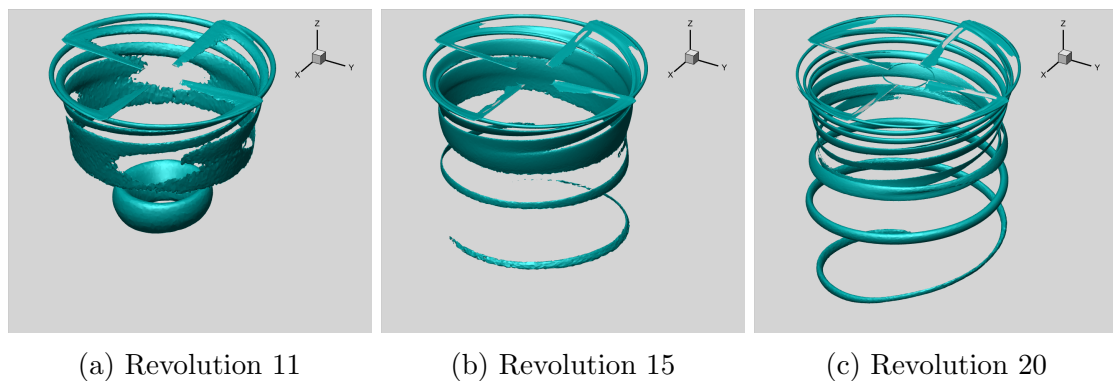
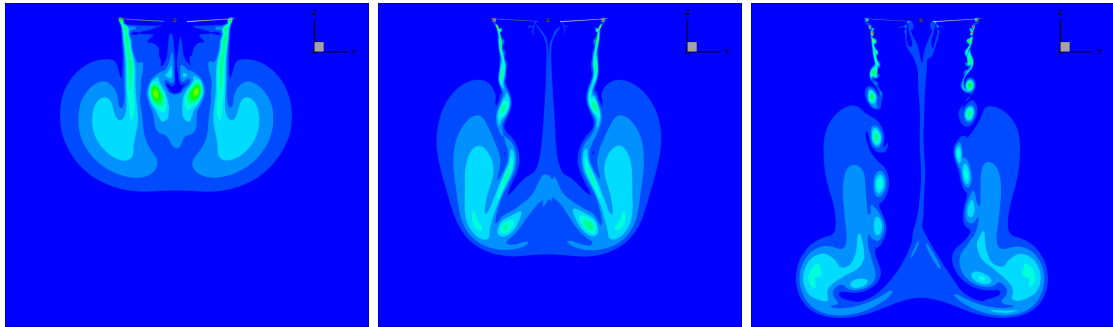


Figure 8: Simulation 1 (Mach sensor and Roe scheme) vortex structures.

In the Simulation 1, solution is diverged to negative density at the tip of the rotor where vortex formation occurs. To overcome this issue, effect of the Harten entropy correction had to be increased and the simulation could be completed.

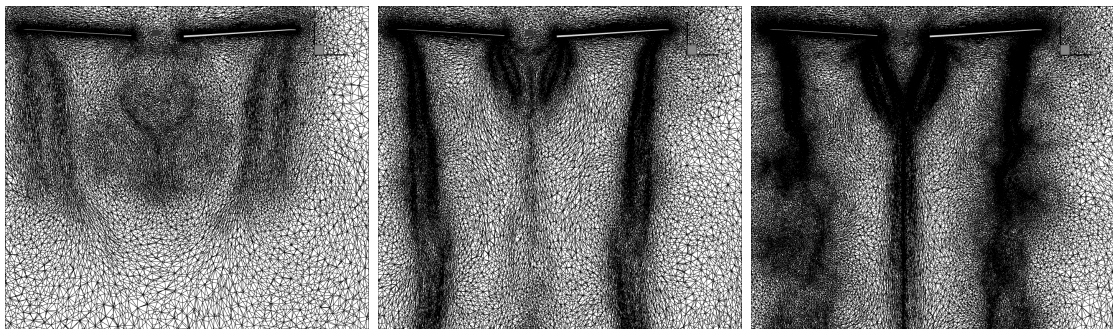


(a) Revolution 11

(b) Revolution 15

(c) Revolution 20

Figure 9: Simulation 1 (Mach sensor and Roe scheme) wake region.

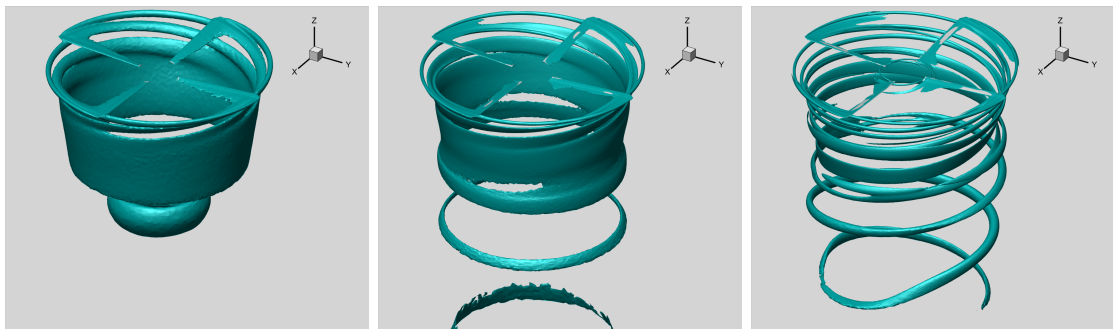


(a) Revolution 11

(b) Revolution 15

(c) Revolution 20

Figure 10: Simulation 1 (Mach sensor and Roe scheme) mesh topology.



(a) Revolution 11

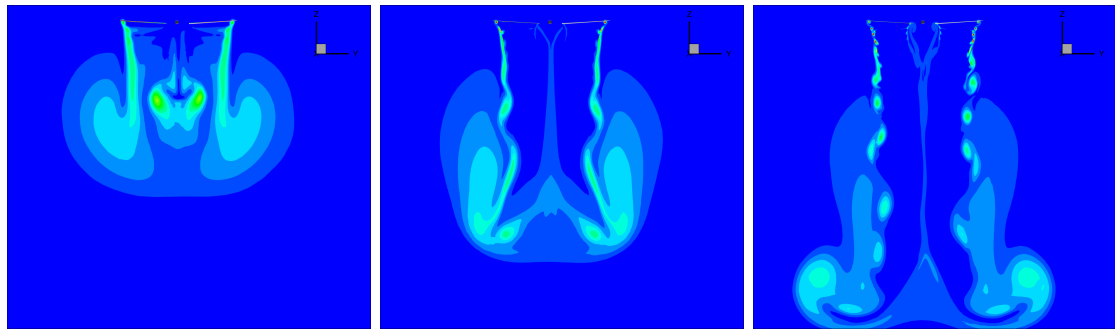
(b) Revolution 15

(c) Revolution 21

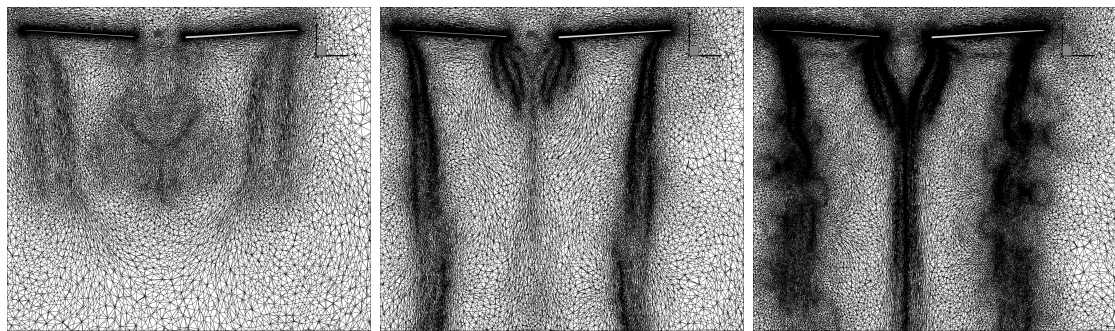
Figure 11: Simulation 2 (Mach sensor and HLLC scheme) vortex structures.

In the second simulation we utilised HLLC scheme. Since HLLC scheme is proven to be positivity preserving [10, 21], the problem with negative density did not occur for HLLC scheme. The solutions and mesh structures for simulation 2 are shown in Fig. 11 and 13. The same adaptation sizes as in the simulation 1 were used in this simulation. However, last three levels had the adaptation size of 2400000. At the last iteration, the number of nodes was 9,870,963.

The third simulation was started using Roe scheme and entropy sensor function. However, due to the negative density problem, Roe scheme gave non-physical solutions and, at this time, Harten entropy correction did not work. Therefore, after revolution 15, we had to switch HLLC scheme. The resulted mesh structures and vortex structures can

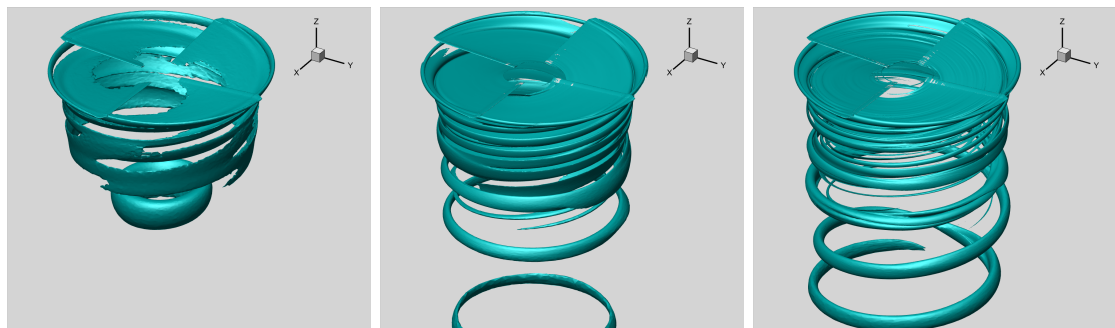


(a) Revolution 11 (b) Revolution 15 (c) Revolution 21
 Figure 12: Simulation 2 (Mach sensor and HLLC scheme) wake region.



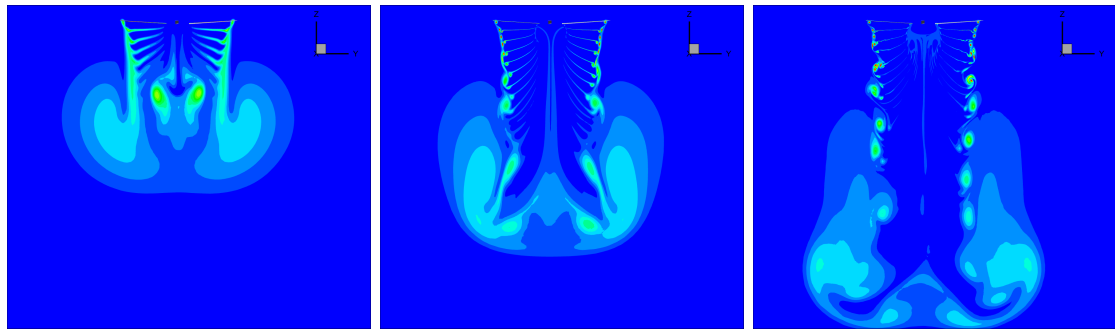
(a) Revolution 11 (b) Revolution 15 (c) Revolution 21
 Figure 13: Simulation 2 (Mach sensor and HLLC scheme) mesh structures.

be seen in Fig. 14 and 16. The refinement levels are designed same as the simulation 2 and there were 23,127,097 number of nodes in the domain at the last time step. Since entropy sensor captured the trailing edge wake sheet as well, node number is excessively larger than other two simulations. To the authors knowledge, there is no solution in the literature that captures trailing edge wake sheet and tip vortices using adaptive mesh refinement. However, this study and previous studies [4] showed that the entropy sensor can capture the wake layer (trailing edge wake sheet).



(a) Revolution 11 (b) Revolution 15 (c) Revolution 21
 Figure 14: Simulation 3 (entropy sensor and Roe/HLLC scheme) vortex structures.

Fig. 17 shows the development of performance parameters(thrust and torque coefficients) over time step. As seen, thrust coefficients of Mach sensor simulations show better agreement with the experimental solution where the experimental solutions are obtained

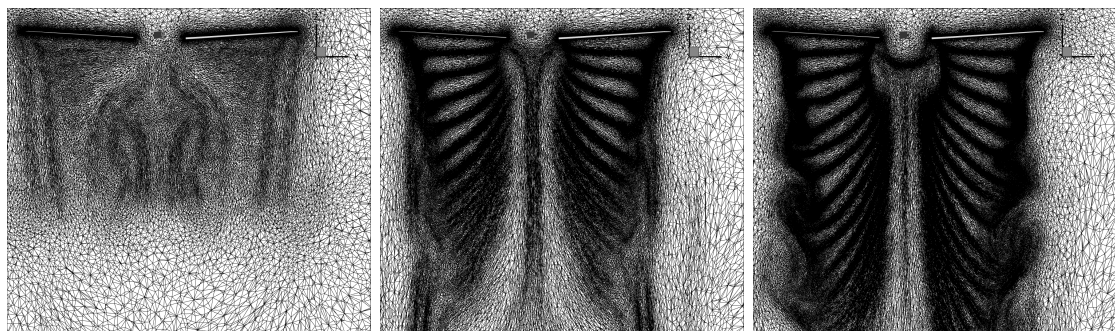


(a) Revolution 11

(b) Revolution 15

(c) Revolution 21

Figure 15: Simulation 3 (entropy sensor and Roe/HLLC scheme) wake region.



(a) Revolution 11

(b) Revolution 15

(c) Revolution 21

Figure 16: Simulation 3 (entropy sensor and Roe/HLLC scheme) mesh structures.

from Balch and Lombardi’s study [22]. However, they do not show the same behaviour for torque coefficient. Vice versa is true for entropy sensor simulation. However, since RANS simulations are not performed, it is early to say which one is the best. The best solution might be the combination of entropy and Mach sensors. Entropy sensor has an advantage to capture the wake region whereas it has lack of ability to capture velocity gradients the farther away from the body. So that, it creates coarse meshes over potential flow regions, whereas Mach sensor behaves better.

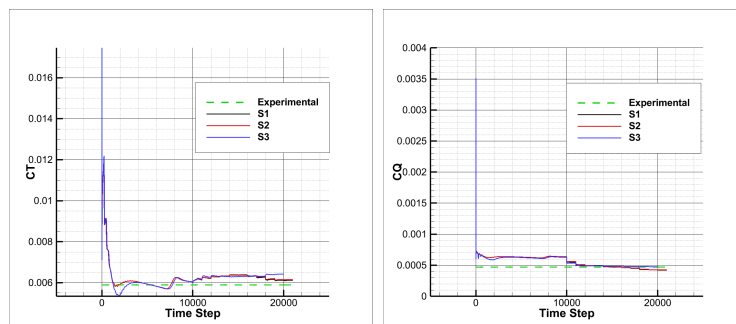


Figure 17: Performance parameter comparison of simulations.

In addition, there can be observed highly non-linear behaviour in the wake region plots above. In these regions, flow lost its periodicity and becomes highly time-dependent. So that, asymmetric mesh formations appears in these regions and, maybe, asymmetry grows the lost of periodicity, which creates problems for wake capturing since the mesh

adaptation is performed after each revolution.

5 CONCLUSION

We implemented a strategy, employing ALE kinematics on Navier-Stokes Equations, to deal with moving boundaries in compressible flow problems. First, we validated the methodology on 2D flow around heaving and pitching airfoil which is proposed in HiOCFD4 workshop. Then, the algorithm is employed to solve flow around S-76 main rotor.

The results of S-76 main rotor analysis gave very similar results to the previous study [4] with SU2. The simulations proved again that Euler simulations around a rotor can be performed using an-isotropic adaptive mesh refinement and the refinement can provide advantages in the speed of the simulation by avoiding unnecessary computational load. However, until RANS simulations are performed, we can not say that the performance parameters are accurately estimated by AMR. Another key outcome of this paper is wake layer under rotor can be captured and refined by entropy sensor function unlike the other simulations in literature. Entropy sensor, in addition, estimated more accurate figure of merit than Mach sensor, however, thrust coefficient estimation of Mach sensor is better than Entropy sensor. To compare them precisely, we need to wait RANS simulation results. Another point is that Roe scheme does not preserve the positivity when strong expansion occurs. Since this is the case at the tip of the rotor, Roe scheme failed. Entropy correction can improve the behaviour of Roe in the case of expansion with the expense of increased dissipation, however HLLC is found to be better in this study.

ACKNOWLEDGEMENTS

Authors gratefully acknowledge that this project is supported by Scientific and Technological Research Council of Turkey (TUBITAK) as a TUBITAK 1001 project with the project number of 122M598.

REFERENCES

- [1] N. Hariharan and L. Sankar. A review of computational techniques for rotor wake modeling. In *38th Aerospace Sciences Meeting and Exhibit*, page 114, 2000.
- [2] N. S. Hariharan. An overview of wake-breakdown in high-fidelity simulations of rotor in hover. In *AIAA Scitech 2020 Forum*, page 0530, 2020.
- [3] N. M. Chaderjian. Advances in rotor performance and turbulent wake simulation using des and adaptive mesh refinement. *Seventh International Conference on Computational Fluid Dynamics (ICCFD7)*, 2012.
- [4] E. Aksoy and M. Sahin. An application of anisotropic mesh refinement to solve flow around the s-76 main rotor with swept-tapered tip. In *AIAA SciTech 2022 Forum*, page 1319, 2022.
- [5] S. Akkurt and M. Sahin. An efficient edge based data structure for the compressible reynolds-averaged navier–stokes equations on hybrid unstructured meshes. *International Journal for Numerical Methods in Fluids*, 2021.

- [6] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.
- [7] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. Gropp, et al. *Petsc users manual*. 2019.
- [8] A. Jameson and S. Shankaran. An assessment of dual-time stepping, time spectral and artificial compressibility based numerical algorithms for unsteady flow with applications to flapping wings. In *19th AIAA Computational Fluid Dynamics*, page 4273. 2009.
- [9] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the hllc-riemann solver. *Shock waves*, 4:25–34, 1994.
- [10] P. Batten, N. Clarke, C. Lambert, and D. M. Causon. On the choice of wavespeeds for the hllc-riemann solver. *SIAM Journal on Scientific Computing*, 18(6):1553–1570, 1997.
- [11] P. L. Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.
- [12] W. K. Anderson and D. L. Bonhaus. An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers & Fluids*, 23(1):1–21, 1994.
- [13] pyamg, 2018. URL <https://pyamg.saclay.inria.fr/pyamg.html>.
- [14] A. Loseille. Unstructured mesh generation and adaptation. In *Handbook of Numerical Analysis*, volume 18, pages 263–302. Elsevier, 2017.
- [15] W. G. Habashi, J. Dompierre, Y. Bourgault, D. Ait-Ali-Yahia, M. Fortin, and M.-G. Vallet. Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent cfd. part i: general principles. *International Journal for Numerical Methods in Fluids*, 32(6):725–744, 2000.
- [16] K. J. Fidkowski and P. L. Roe. An entropy adjoint approach to mesh refinement. *SIAM Journal on Scientific Computing*, 32(3):1261–1287, 2010.
- [17] C. Farhat, P. Geuzaine, and C. Grandmont. The discrete geometric conservation law and its effects on nonlinear stability and accuracy. In *15th AIAA Computational Fluid Dynamics Conference*, page 2607, 2001.
- [18] Bl3 - heaving and pitching airfoil, 2016. URL <https://how4.cenaero.be/content/bl3-heaving-and-pitching-airfoil>.
- [19] P.-O. Persson. Bl3 - heaving and pitching airfoil, introduction and summary by p.-o. persson (ucberkeley, usa). Retrieved from: <https://how4.cenaero.be/content/baseline-test-case-summaries>.
- [20] N. A. Wukie. A no-slip, moving-wall boundary condition for the navier-stokes equations. In *AIAA Aviation 2019 Forum*, page 3318, 2019.

- [21] B. Einfeldt, C.-D. Munz, P. L. Roe, and B. Sjögren. On godunov-type methods near low densities. *Journal of computational physics*, 92(2):273–295, 1991.
- [22] D. Balch and J. Lombardi. Experimental study of main rotor tip geometry and tail rotor interactions in hover. volume 1. text and figures. 1985.



INTEGRATION OF LIFE CYCLE ASSESSMENT METHODOLOGY AS AN ENVIRONMENT DISCIPLINE MODULE IN MULTIDISCIPLINARY ANALYSIS AND OPTIMIZATION FRAMEWORK

Thomas Bellier^{12*}, Joseph Morlier¹, Cees Bil², Annafederica Urbano¹ and
Adrian Pudsey²

1: Institut Clément Ader
ISAE-SUPAERO, Mines Albi, UPS, INSA, CNRS
Université de Toulouse
Toulouse, France
thomas.bellier@isae-supero.fr, <https://ica.cnrs.fr/home/>

2: School of Engineering
Royal Melbourne Institute of Technology
Melbourne, Victoria, Australia
<https://www.rmit.edu.au/>

Abstract. *As sustainability becomes one of the main challenges of the aerospace industry, we need to find new ways to integrate it into the design phase of aerospace systems. The Multidisciplinary Analysis and Optimization (MDAO) framework is a great host for an environmental discipline thanks to its modularity. However current Life Cycle Assessment (LCA) software do not integrate well with other computation tools, and are designed for low amounts of simulations on massive databases, while MDAO requires often many iterations with only slight variations. This work presents newly developed tools aiming at bridging those two valuable methods to enable better use of LCA within MDAO. Symbolic links between classic design variables and associated LCA parameter, python-based tools compatible with **OpenMDAO**, and optimization of the LCA algorithms to allow for multiple runs on variations of the same overall system makes it possible to integrate LCA considerations inside an MDAO model at reduced performance cost. The resulting framework makes ecodesign more accessible as environmental impacts can be used inside the design process, possibly as main objectives or constraints.*

Keywords: environment, life cycle, sustainability, multidisciplinary, ecodesign

1 INTRODUCTION

Integrating environmental concerns in a Multidisciplinary Analysis and Optimisation (MDAO) framework is challenging, for multiple reasons. The software, the language, and the people are different between the technical design and Life Cycle Assessment (LCA) of a project, which means that both tasks are performed at best with slow communication, at worst completely independently and the environmental analysis is carried out only on the already finished design. In that context, reducing environmental impact can be done during an MDAO, but that requires identifying beforehand which variables will impact the most relevant impacts, and the lack of integrated LCA can lead to important bias with many variables impacting the environmental aspect of a design being ignored.

Integrating LCA as an environmental discipline inside an MDAO would solve those problems, as it would give access to all relevant variables to the LCA module avoiding biases and errors, and also give access to all relevant environmental impact to the rest of the MDAO model, including as main constraint or objective, to directly optimise using them and not just analyse the result. This however presents multiple challenges: an LCA requires access to a lot of data and variables, which increases the links between the disciplines in the model. Also, as mentioned previously, those two processes are usually carried out by different departments, and there is an extremely limited choice of tools that uses the same programming language to be used together. This work presents the specificity of LCA in an MDAO context, the developed structure to integrate into python, and two simple examples to illustrate some of the potential benefits of this integration.

2 LIFE CYCLE ASSESSMENT

2.1 Principles

Life Cycle Assessment (LCA, also called Life Cycle Analysis) is a standardized method to evaluate the environmental consequences of a product or activity across its entire life. It's defined by a family of ISO norms [1], from ISO 14040 to ISO 14044, that can be extended by different standard guidelines, such as the European ILCD [2]. It's a very powerful tool to establish impact assessment, compare different products, and create environmental certifications as the defined methodology ensures the results are as complete as possible, and that every product tested for comparison or certification will go through the same fare analysis. LCA methodology is split into four main parts, as described in Figure 1. All of those need to be carried out properly. [3, 4]

2.1.1 Goal and Scope Definition

The first thing when conducting an LCA is to define correctly the problem. [5] Why is the study performed? What is the product studied and its purpose? Which impact are studied? Will it be compared to something else? Answering those questions requires looking at:

- Choosing a functional unit: the functional unit is a quantitative description of the service or utility for which the LCA is performed, and will be used as the reference to scale the output of the impacts and inventory analysis. It will be also the scaling definition for comparison. For example, when studying a wind turbine or any other

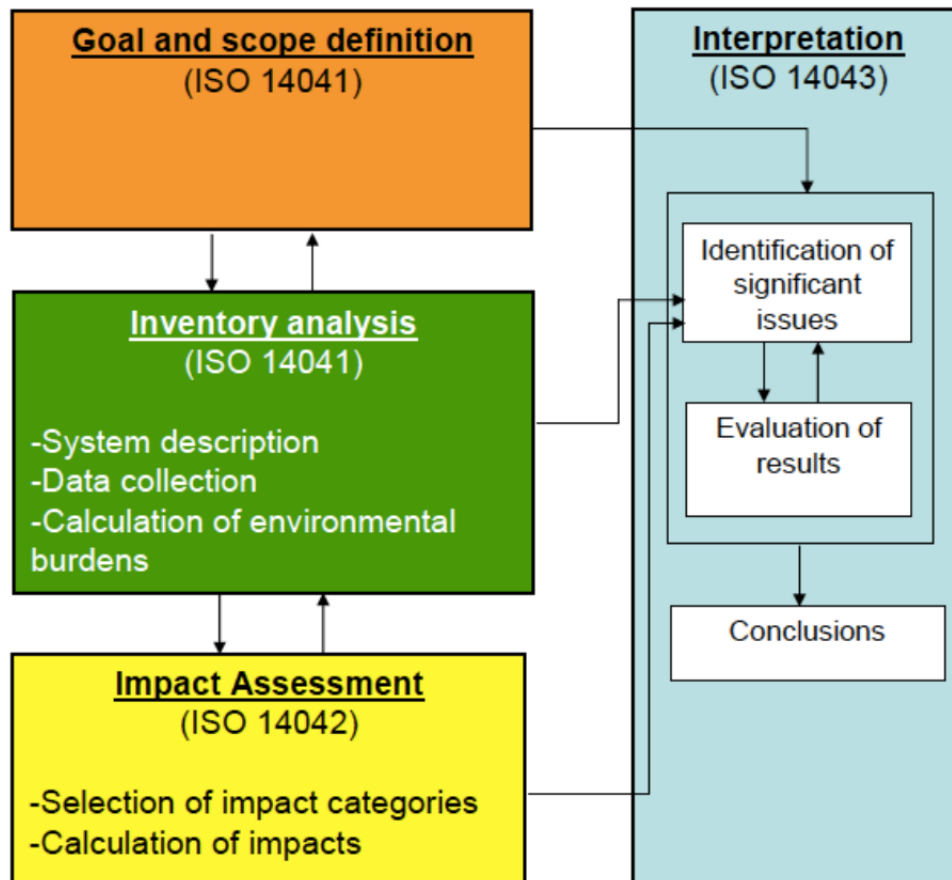


Figure 1: LCA methodology and ISO norms [1]

source of electricity, a common choice for the functional unit would be 1 kWh of electricity sent to the network, as it's the primary product of that wind turbine. Therefore, every impact would be listed "per kWh produced" and if compared to another source of electricity, each output would be scaled to that 1 kWh. [6]

- Defining the scope of the LCA: defining which activities and processes belong to the life cycle of the product or service that is studied. The life cycle is core to the LCA methodology, and in this phase, it's important to consider every step that will be relevant for the impact assessment, like the conception, extraction of raw materials, transformations, manufacturing, logistics, utilisation, disposal, recycling, and others.
- Choosing the studied impacts: selecting and defining the impacts that would be studied as the output of the LCA. [7] Those can be local ones like local pollution with micro particles, global ones like the global warming potential (GWP), or even very application-specific ones, like the space debris generation for an orbital activity.
- Selecting the geographical and time boundaries of the study: most of the impacts will be time and space dependants, so it's necessary to define for each aspect in the scope a time window and a physical location when relevant.
- Identifying the additional requirements for a certification or an official public release: if the LCA is carried out in order to certify a product or to prove that it fits

environmental standards defined by an external entity, those can require additional steps or requirements. If the LCA is intended to be publicly disclosed, the results need to be reviewed externally.

The goal and scope definition is very important and will influence every other step that comes after. It will determine the validity of the conclusions based on the LCA results.

2.1.2 Inventory Analysis

After the definition of goals and scope, every flow of any resource within the scope needs to be collected in an inventory analysis and scaled based on their contribution to the final functional unit. The analysis often relies on external databases of either generic flow and products, like Ecoinvent [8], or more specialized ones like the SSSD [9] for the space sector. The inventory analysis outputs a life cycle inventory, which contains a list of physical flows for the product or service described by the functional unit.

2.1.3 Impact Assessment

The impact assessment is the translation of the elementary flows of the inventory analysis into relevant environmental quantities or impacts. The three main steps described in the ISO 14040 norm are:

- Selection of the impact categories relevant to the choices made during the goal definition. Those impact categories need to have an environmental model and an associated indicator that can be quantified from the elementary flows of the inventory analysis.
- Classification of the elementary flows, by selecting the relevant flows for the selected impact categories and assigning them to the relevant impact.
- Characterisation of the impacts, by converting the classified elementary flows into values used in the impact categories indicators and environmental models, and then measuring the value of those impact indicators.

2.1.4 Interpretation

The results of the LCA and the values of the impact indicators are used as described after the goal definition. In most cases, the previous steps and especially impact assessment will add uncertainties due to the lack of information about an activity or the difficult correlation between raw outputs and environmental models. [10] Those uncertainties will also be interpreted. [11]

2.2 Challenges for aerospace

LCA is a generic technique and can be used for any project or object, including aerospace systems. However, it is mostly used for systems whose environmental impact is important for the general public. This includes the past years of civilian aviation, especially commercial aviation, as it has a significant contribution to climate change. [12, 13] Various LCA had been made for different aspects of civilian aviation. [14, 15] LCA on a

commercial aircraft reveals that most of the impact, especially on climate change, is due to the usage of kerosene during flight. [16] For that reason, further studies focus on the LCA of the fuel itself [17] or the usage of completely different technologies like hydrogen [18] or electric [19] plane engines.

There is far less available data for space and military projects. Military activities are too secret to share enough data for an LCA study, and space activities used to be considered too niche to have a significant impact, thus explaining the lack of extensive LCA. While an aircraft is not a launcher, the technologies used in aviation are often related to the ones used in space and this extensive research is a useful starting point for the LCA of a launcher. A state of the art of LCA in the space sector [20] shows that LCA for launchers is recent but expanding rapidly. Some reports have been made in the United States [21] or Russia [22], but ESA [23] is the major agency with the most involvement in LCA.

Space is a growing sector, on its influence on the environment is not negligible anymore [24, 25], on different impacts like the GWP [26] or the ozone layer destruction [27, 28]. Knowing the impacts of launchers in the high atmosphere has a lot of uncertainties [20], due to the difficulty of simulating or measuring gas propagation there. [29] This makes it difficult to know the impact of high atmosphere burning of kerosene [30, 31], the influence of black carbon [32] or other chemical residues, and the impacts of solid boosters [33–35] with non-carbon-based fuels.

3 INTEGRATION

3.1 Environment as a discipline

In all previously mentioned studies, and in most LCA studies in general, the environmental assessment is carried out once, on a finished design. This makes it possible to compare existing products or check for example the compliance of a proposed design with environmental requirements. However, this means that the environmental parameters are not optimised directly, as objectives. If we consider an MDAO problem, this out-of-loop LCA would be modeled as shown in Fig.2. In this example, the LCA is only done after optimisation and its outputs are entirely decoupled from the design process.

This common approach is generally used because LCA software doesn't allow integration of a project in some external development or optimisation tool. This means that in order to optimise the environmental impact using this structure, one must either do a very slow iterative optimisation process or extract from the LCA process the main contributing factors and add them manually in the optimisation loop.

The goal of this work is to integrate directly the entire LCA model inside the optimisation loop, to allow for quick and easy access to all of the environmental output of the LCA and use them as objectives, constraints, or simply internal variables. In that scheme, an MDAO problem with coupled LCA would have the structure presented in Fig.3. There, the LCA is fully integrated into the MDAO framework and is usable as any normal discipline module, which means that its outputs can be used in the optimiser or another function.

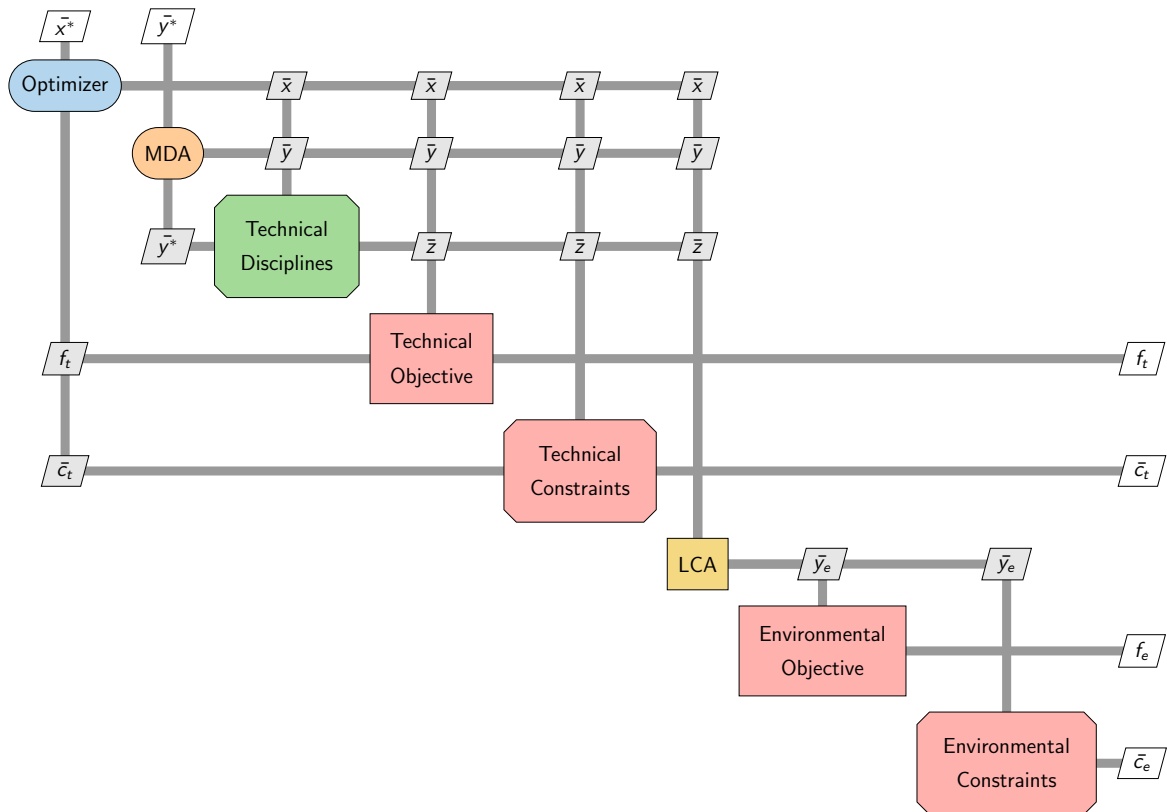


Figure 2: Generic XDSM diagram with decoupled LCA

To manage both aspects in one single environment, and stay compatible with a lot of existing projects, this development uses the *Brightway 2* [36] and *OpenMDAO* [37] python modules. Those are currently the only compatible tools for MDAO and LCA respectively.

3.2 Database linking

Linking LCA with a technical MDAO model still requires following the LCA principles described in Sec.2. Any database in *Brightway 2* [36] format would work.

3.2.1 Goal and Scope Definition

The Goal and Scope Definition has to be carried out like a normal LCA study. It's especially important to think of the intended coupling between the environment and other disciplines, and thus considering the following points:

- the functional unit should be directly related to the output of the overall MDAO model;
- the scope should integrate at least everything modeled in the MDAO;
- the studied impact may be used as constraints, objectives, or other variables within the MDAO framework.

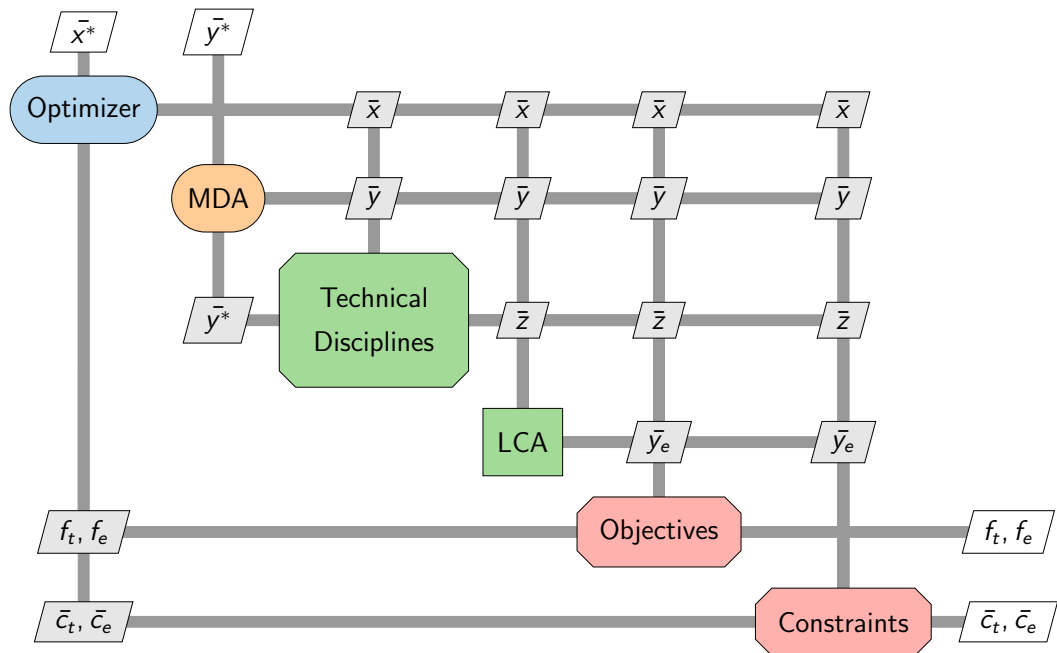


Figure 3: Generic XDSM diagram with integrated LCA

3.2.2 Inventory Analysis

The Inventory Analysis is the most important part of the coupling between MDAO and LCA. After completing the inventory analysis like a normal LCA, all of the inventory values that link with a MDAO variable need to be identified. In their declaration inside the MDAO framework, they need to be associated with the corresponding environmental database entry from *Brightway 2*. [36] That way, every time the LCA computation is done, those values are updated with their current value in the other MDAO discipline, hence creating the link between the environmental discipline and the others.

3.2.3 Impact Assessment

The impact categories are selected based on the goal definition and should be fixed throughout the optimisation process. The classification and characterisation is done using *Brightway 2*. [36]

3.2.4 Interpretation

The results of the LCA and the values of the impact indicators are used as planned in the scope definition, which means in most case converted into variables usable in and MDAO framework.

4 EXAMPLES

4.1 Sellar problem

4.1.1 Definition of the normal problem

The Sellar problem is a common example used to demonstrate and test the *OpenMDAO* library [37]. It can be mathematically described as the following optimisation problem:

$$\min : \quad x_1^2 + z_2 + y_1 + e^{-y_2} \quad (1)$$

$$\text{w.r.t. :} \quad x_1, z_1, z_2 \quad (2)$$

$$\text{subject to :} \quad (3)$$

$$3.16 - y_1 \leq 0 \quad (4)$$

$$y_2 - 24.0 \leq 0 \quad (5)$$

4.1.2 Multi-objective environmental variant

We assume arbitrarily that the Sellar problem describes an actual physical product, where y_1 would describe a quantity of steel and y_2 a quantity of wood. These two variables will now be linked to their respective entries in the environmental database, here *ecoinvent v3.8* [8]. The new environment module is configured to output the GWP impact of that system, only made of these masses of wood and steel, using the *ReCiPe* midpoint GWP100 method [38]. This integration gives us the new MDAO model highlighted on Fig. 4.

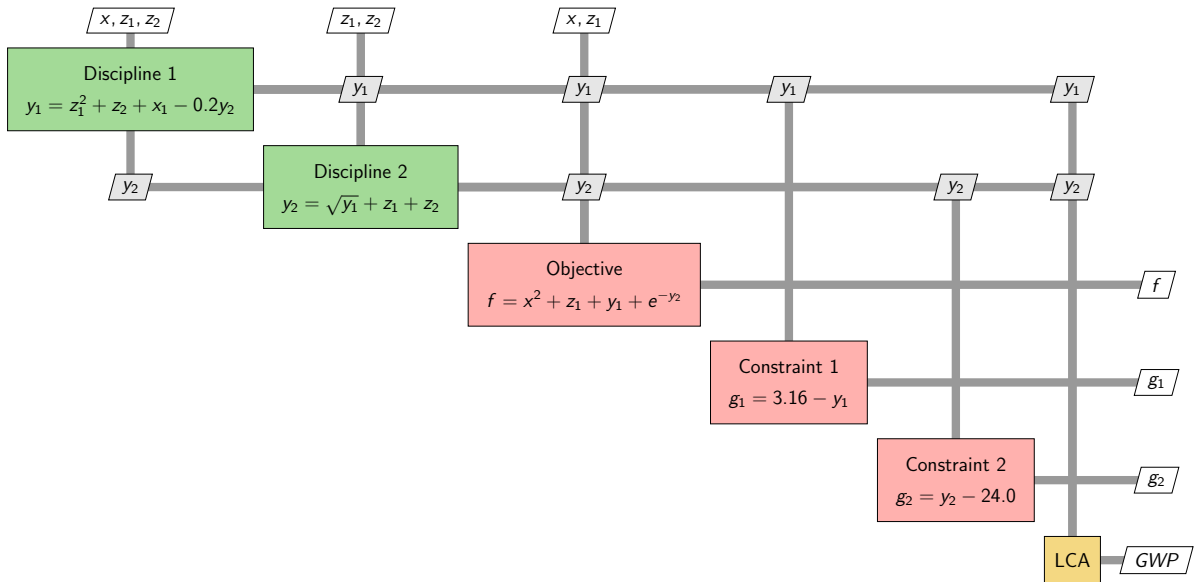


Figure 4: XDSM diagram of the Sellar problem augmented with an environment module

A multi-objective optimisation adding the newly computed GWP as the second objective is carried out. Using the NSGA-2 [39] algorithm implementation from the library *Pymoo* [40], with 50 individuals in the population and using a tolerance of 10^{-2} on the objectives as a termination criterion, it gives the Pareto front on Fig.5. The 2 extreme

points of the front are the result of a single objective optimisation using COBYLA [41] and are obtained with the values in Table 1.

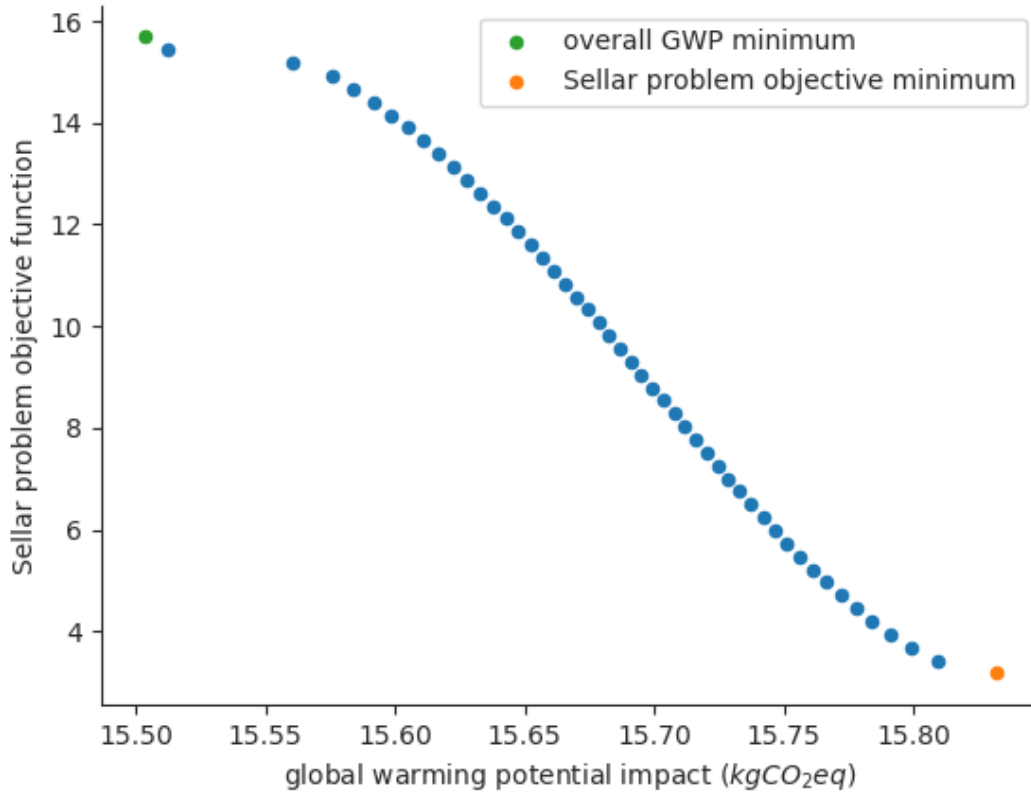


Figure 5: Pareto front of the Sellar optimisation problem augmented with an environment module

Table 1: Design variable and objective values for the extreme points of the Pareto front

x_1	z_1	z_2	GWP ($kgCO_2eq$)	Sellar objective
3.52	7.88×10^{-9} (≈ 0)	3.04×10^{-8} (≈ 0)	15.50	15.69
1.06×10^{-8} (≈ 0)	1.98	2.59×10^{-9} (≈ 0)	15.83	3.18

Using this technique, we can immediately assess the compromise on our original objective function if choosing a solution better for our chosen environmental parameter. It's important to note here that the Sellar problem doesn't represent any real problem and that any entry in any environmental database could have been assigned to any accessible variable from the *OpenMDAO* model. The next example presents an instance where the database and entries used actually match the technical problem.

4.2 Hybridised twin-turboprop aircraft

4.2.1 Hybrid Beechcraft King Air

The Beechcraft King Air is a line of twin-turboprop utility aircraft produced at more than 3000 units over 60 years and is therefore a good base for a hybrid aircraft. The library *Openconcept* [42, 43] proposes an architecture based on the King Air C90GT model, with a 3.75t MTOW and originally two 550shp turboprop engines. This aircraft is modified with the addition of an electric motor on each propeller, with a rating of at most 500hp and alimented with a battery of at most 2250kg and a specific energy of 450Wh/kg. The MTOW can be increased up to 5.70t.

4.2.2 Definition of the problem

The aircraft model of the King Air C90GT is modified with a hybrid system, which adds 7 design variables to the problem: engine power, motor power, battery weight, fuel capacity, cruise hybridisation, climb hybridisation, and descent hybridisation. To ensure the aircraft still flies, the model adapted in *Openconcept* [42, 43] adds 2 more design variables: MTOW and wing surface. The minimum and maximum values of these are presented in Table 3, with the initial value in the optimiser.

Our study focuses on two main objectives, the range and the global warming potential impact (GWP). The range is verified by simulating a trajectory. For the GWP, we use a simple LCA analysis using the *ecoinvent v3.8* [8] database. Like the previous example, the output used is the *ReCiPe* midpoint GWP100 method [38]. The database links are as shown in Table 2. When available, the location of the entry is set to Europe.

Table 2: Environmental coupling for the hybrid aircraft model

model parameter	Ecoinvent v3.8 entry
battery weight	'battery cell production, Li-ion'
motor weight	'electric motor production, vehicle'
engine weight	'internal combustion engine production, passenger car' ¹
empty weight ²	'aluminium production, primary, ingot'
fuel used	'market for kerosene'
electricity used	'market group for electricity, low voltage'

It is assumed that the aircraft will fly a thousand cycles at maximum range after being built. This model is obviously simplified here; but the idea of this example is that any design team with a more precise model and a better-suited database, or with access to a dedicated environmental study, can use those in a similar table to easily improve the environmental model being computed.

¹Ecoinvent does not include an entry for aircraft engine.

²Excludes motors, engines, and batteries.

4.2.3 Single objective optimisation

Table 3 presents the design variables values and results after optimisation for this problem, with the range fixed at 400NM and using the GWP as the sole objective, using COBYLA [41]. Figure 6 presents the resulting trajectory and energy consumption for this 400 nautical miles range solution.

Table 3: Example of hybrid aircraft optimisation for a range of 400NM

variable	min	init	max	value	units
MTOW	4000	5000	5700	5700	<i>kg</i>
wing surface	15	25	40	34	<i>m²</i>
engine power	0	1000	3000	298	<i>kW</i>
motor power	450	1000	3000	652	<i>kW</i>
battery weight	20	1000	3000	1607	<i>kg</i>
fuel capacity	500	1000	3000	500	<i>kg</i>
cruise hybridisation	0	0.5	1	0.71	
climb hybridisation	0	1	1	0.785	
descent hybridisation	0	0.5	1	0.337	
GWP				0.712	<i>kgCO₂eq/km</i>

As expected, the electrification reduces both the fuel consumption and the range of the aircraft. In order to minimise the size of the engine and stay in the best efficiency window, the engine will run almost always at the same power setting while the differences in throttle setting will be absorbed by the electric system, which is always close to 100% efficiency.

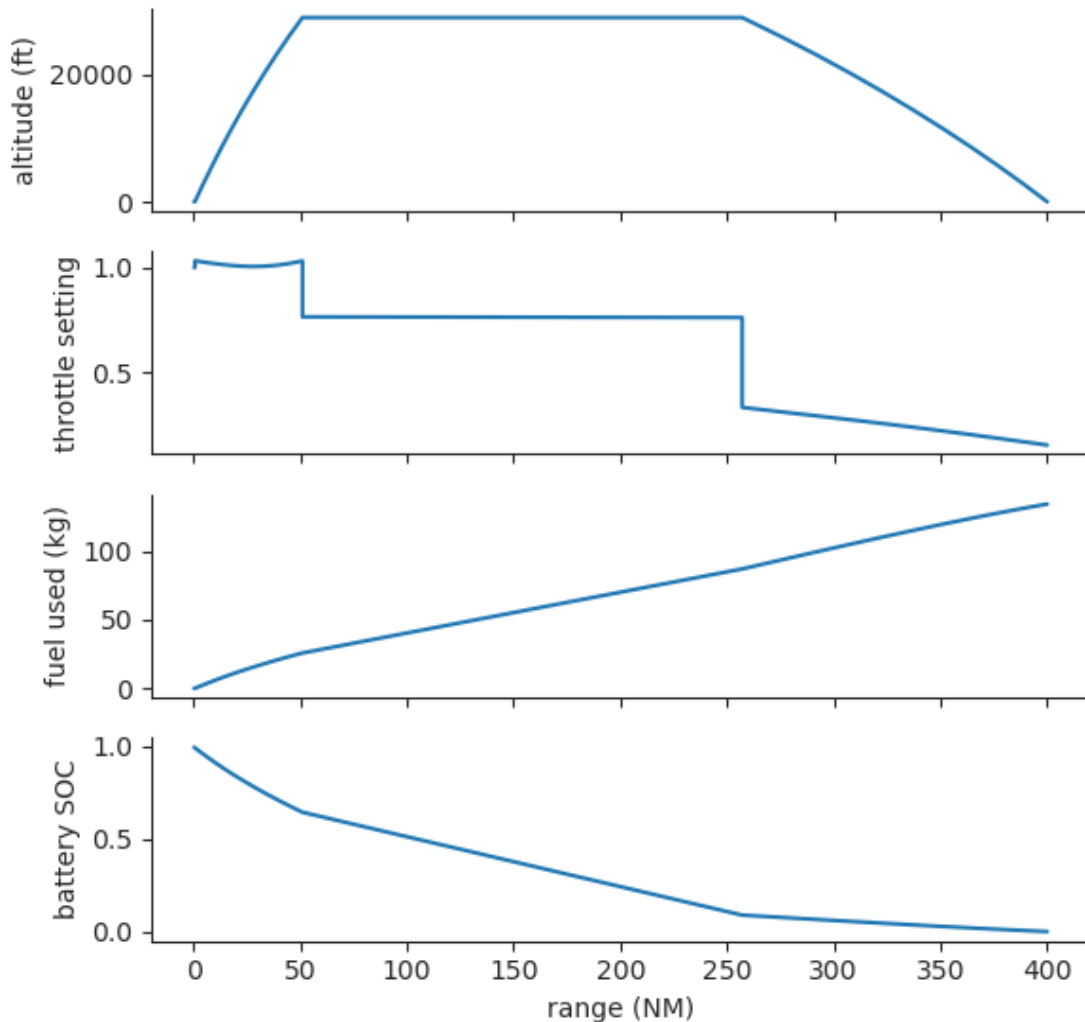


Figure 6: Optimal trajectory and energy utilisation for a hybrid aircraft with 400 nautical miles range

4.2.4 Multi-objective optimisation results

A multi-objective optimisation of the same aircraft design problem adding the range as the second objective instead of a fixed parameter is carried out. Using the NSGA-2 [39] algorithm implementation from the library *Pymoo* [40], with 12 individuals in the population and using a tolerance of 10^{-2} on the objectives as a termination criterion, it gives the Pareto front on Fig.7.

As expected, to increase the range of the aircraft the electric part of the hybrid system has to decrease, which leads to higher fuel consumption and in the end higher impact on the climate. With a range of 300NM or less, the cruise and descent are fully electric and the engine is only used for takeoff. For a range of more than 500NM, the engine produces more than half of the power during cruise.

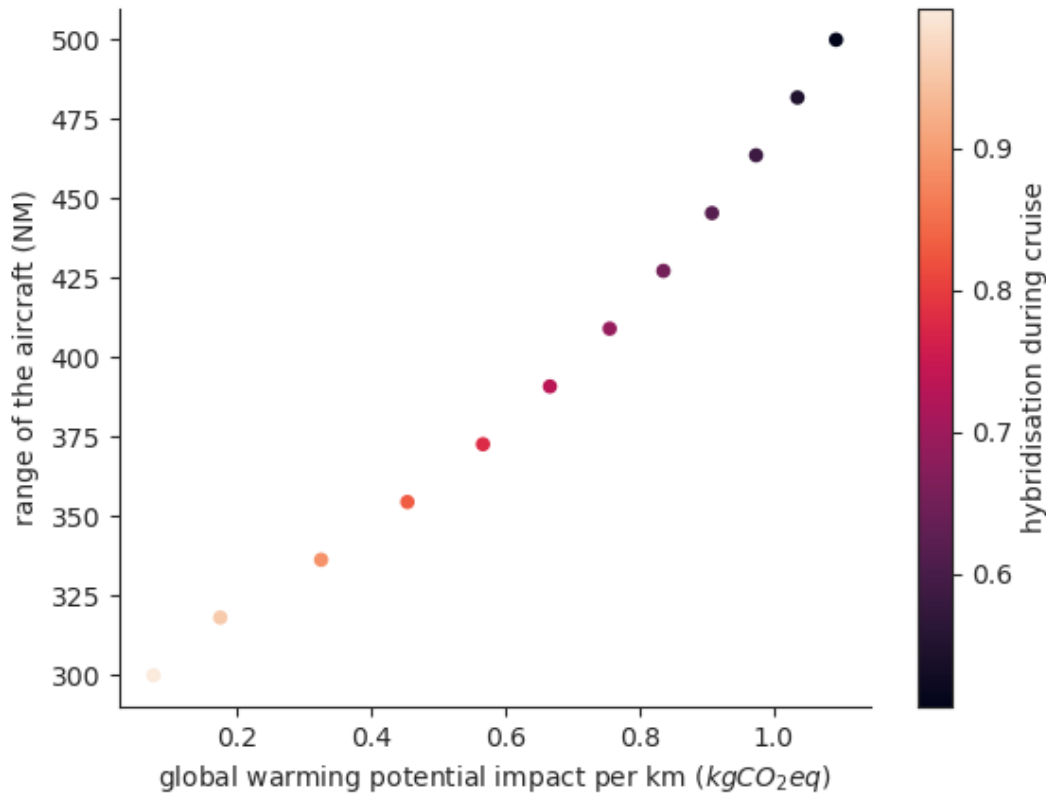


Figure 7: Pareto front for the multi-objective hybrid aircraft optimisation

A reduction of the range from 500NM ($1.1kgCO_2eq/km$, 50% hybridisation) to 370NM ($0.56kgCO_2eq/km$, 78% hybridisation) means a 26% reduction in range implies a 50% reduction in CO_2 emissions. And because our model uses life cycle assessment and not just fuel consumption, this includes other effects that penalise the hybrid aircraft, such as the production of the aircraft itself, with costly batteries and a shorter lifespan for the shorter range design.

5 CONCLUSION

The two simple examples presented highlight the potential of linking directly LCA to a MDAO model. By having access to the environmental impact in the main optimiser, we can use them as the main objective, like the GWP here, but given that they behave like other variables, they could have also been used as constraints or within another discipline and propagated further into the model. This approach also eliminates many biases in ecodesign, as the environment-related variables are computed using a proper methodology. In the aircraft example, that means including manufacturing, electricity, and other greenhouse gases instead of focusing on simply the carbon dioxide emitted by burning fuel in-flight.

Some limitations still appear: the added complexity may impair the computational performance of the MDAO model, and LCA result can be given with an associated un-

certainty, for which the amplitudes can vary enormously depending on the given problem; this uncertainty would be ignored when used as MDAO variables. Furthermore, the limited choice of common tools limits the application to only work with *OpenMDAO* and *Brightway 2*, which also limits the number of potential users. But the test results are promising, and further development may reduce uncertainties and performance penalties, and simplify the process to make it more accessible to inexperienced users. This idea will be applied in the future to complete aerospace models, especially realistic launchers, with extensive LCA to promote this ecodesign methodology for ambitious projects.

ACKNOWLEDGEMENTS

This work was supported by the Royal Melbourne Institute of Technology (RMIT) and the Toulouse graduate School of Aerospace Engineering EUR TSAE grant, 17-EURE-0005.

REFERENCES

- [1] ISO. *Environmental management. Life cycle assessment. Principles and framework*. 2020. ISBN 978-0-539-01348-1. OCLC: 9521571302.
- [2] I. for Environment and Sustainability (Joint Research Centre). *International Reference Life Cycle Data System (ILCD) Handbook :general guide for life cycle assessment : detailed guidance*. Publications Office of the European Union, 2010. ISBN 978-92-79-19092-6. URL <https://data.europa.eu/doi/10.2788/38479>.
- [3] M. Z. Hauschild. Introduction to LCA Methodology. In M. Z. Hauschild, R. K. Rosenbaum, and S. I. Olsen, editors, *Life Cycle Assessment: Theory and Practice*, pages 59–66. Springer International Publishing, Cham, 2018. ISBN 978-3-319-56475-3. doi:10.1007/978-3-319-56475-3_6. URL https://doi.org/10.1007/978-3-319-56475-3_6.
- [4] O. Jolliet, M. Saade-Sbeih, S. Shaked, A. Jolliet, and P. Crettaz. *Environmental Life Cycle Assessment*. CRC Press, 2015. ISBN 978-1-4398-8770-7. URL <https://books.google.ae/books?id=LYvwCgAAQBAJ>.
- [5] M. A. Curran. Overview of Goal and Scope Definition in Life Cycle Assessment. In M. A. Curran, editor, *Goal and Scope Definition in Life Cycle Assessment*, LCA Compendium – The Complete World of Life Cycle Assessment, pages 1–62. Springer Netherlands, Dordrecht, 2017. ISBN 978-94-024-0855-3. doi:10.1007/978-94-024-0855-3_1. URL https://doi.org/10.1007/978-94-024-0855-3_1.
- [6] K. R. Haapala and P. Prempreeda. Comparative life cycle assessment of 2.0 MW wind turbines. *International Journal of Sustainable Manufacturing*, 3(2): 170, 2014. ISSN 1742-7223, 1742-7231. doi:10.1504/IJSM.2014.062496. URL <http://www.inderscience.com/link.php?id=62496>.
- [7] R. K. Rosenbaum. Selection of Impact Categories, Category Indicators and Characterization Models in Goal and Scope Definition. In M. A. Curran, editor, *Goal*

- and Scope Definition in Life Cycle Assessment*, LCA Compendium – The Complete World of Life Cycle Assessment, pages 63–122. Springer Netherlands, Dordrecht, 2017. ISBN 978-94-024-0855-3. doi:10.1007/978-94-024-0855-3_2. URL https://doi.org/10.1007/978-94-024-0855-3_2.
- [8]ecoinvent Association. ecoinvent Database, 2021. URL <https://ecoinvent.org/the-ecoinvent-database/>.
- [9] A. Wilson, M. Vasile, C. Maddock, and K. Baker. The Strathclyde Space Systems Database: A New Life Cycle Sustainability Assessment Tool for the Design of Next Generation Green Space Systems. Sept. 2018.
- [10] E. A. Groen, E. A. M. Bokkers, R. Heijungs, and I. J. M. de Boer. Methods for global sensitivity analysis in life cycle assessment. *The International Journal of Life Cycle Assessment*, 22(7):1125–1137, July 2017. ISSN 1614-7502. doi:10.1007/s11367-016-1217-3. URL <https://doi.org/10.1007/s11367-016-1217-3>.
- [11] E. A. Groen and R. Heijungs. Ignoring correlation in uncertainty and sensitivity analysis in life cycle assessment: what is the risk? *Environmental Impact Assessment Review*, 62:98–109, Jan. 2017. ISSN 0195-9255. doi:10.1016/j.eiar.2016.10.006. URL <https://www.sciencedirect.com/science/article/pii/S0195925516300907>.
- [12] R. K. Pachauri, L. Mayer, and I. P. on Climate Change, editors. *Climate change 2014: synthesis report*. Intergovernmental Panel on Climate Change, Geneva, Switzerland, 2015. ISBN 978-92-9169-143-2.
- [13] D. S. Lee, D. W. Fahey, A. Skowron, M. R. Allen, U. Burkhardt, Q. Chen, S. J. Doherty, S. Freeman, P. M. Forster, J. Fuglestvedt, A. Gettelman, R. R. De León, L. L. Lim, M. T. Lund, R. J. Millar, B. Owen, J. E. Penner, G. Pitari, M. J. Prather, R. Sausen, and L. J. Wilcox. The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018. *Atmospheric Environment*, 244:117834, Jan. 2021. ISSN 1352-2310. doi:10.1016/j.atmosenv.2020.117834. URL <https://www.sciencedirect.com/science/article/pii/S1352231020305689>.
- [14] A. Johanning and D. Scholz. A First Step Towards the Integration of Life Cycle Assessment into Conceptual Aircraft Design. 2013. URL <https://reposit.haw-hamburg.de/handle/20.500.12738/1372>. Accepted: 2020-08-26T09:17:21Z.
- [15] A. Johanning and D. Scholz. Conceptual Aircraft Design based on Life Cycle Assessment. In *ICAS 2014 Proceedings-29th Congress of the International Council of the Aeronautical Sciences*, 2014.
- [16] João Lopes. Life Cycle Assessment of the Airbus A330-200 Aircraft. Master’s thesis, Universidade Técnica de Lisboa, Nov. 2010.
- [17] C. Koroneos, A. Dompros, G. Roumbas, and N. Moussiopoulos. Life Cycle Assessment of Kerosene Used in Aviation (8 pp). *The International Journal of Life Cycle Assessment*, 10(6):417–424, Nov. 2005. ISSN 1614-7502. doi:10.1065/lca2004.12.191. URL <https://doi.org/10.1065/lca2004.12.191>.

- [18] H. Nojoumi, I. Dincer, and G. F. Naterer. Greenhouse gas emissions assessment of hydrogen and kerosene-fueled aircraft propulsion. *International Journal of Hydrogen Energy*, 34(3):1363–1369, Feb. 2009. ISSN 0360-3199. doi:10.1016/j.ijhydene.2008.11.017. URL <https://www.sciencedirect.com/science/article/pii/S0360319908015048>.
- [19] J. Ribeiro, F. Afonso, I. Ribeiro, B. Ferreira, H. Policarpo, P. Peças, and F. Lau. Environmental assessment of hybrid-electric propulsion in conceptual aircraft design. *Journal of Cleaner Production*, 247:119477, Feb. 2020. ISSN 0959-6526. doi:10.1016/j.jclepro.2019.119477. URL <https://www.sciencedirect.com/science/article/pii/S0959652619343471>.
- [20] T. Maury, P. Loubet, S. M. Serrano, A. Gallice, and G. Sonnemann. Application of environmental life cycle assessment (LCA) within the space sector: A state of the art. *Acta Astronautica*, 170:122–135, May 2020. ISSN 00945765. doi:10.1016/j.actaastro.2020.01.035. URL <https://linkinghub.elsevier.com/retrieve/pii/S0094576520300552>.
- [21] S. S. Neumann. *Environmental Life Cycle Assessment of Commercial Space Transportation Activities in the United States*. Thesis, Apr. 2018. URL <https://rc.library.uta.edu/uta-ir/handle/10106/27352>. Accepted: 2018-06-05T15:57:18Z.
- [22] T. V. Koroleva, P. P. Krechetov, I. N. Semenov, A. V. Sharapova, S. A. Lednev, A. M. Karpachevskiy, A. D. Kondratyev, and N. S. Kasimov. The environmental impact of space transport. *Transportation Research Part D: Transport and Environment*, 58:54–69, Jan. 2018. ISSN 1361-9209. doi:10.1016/j.trd.2017.10.013. URL <https://www.sciencedirect.com/science/article/pii/S1361920917303115>.
- [23] ESA LCA Working Group. Space system Life Cycle Assessment (LCA) guidelines. Handbook ESSB-HB-U-005, ESA, Oct. 2016.
- [24] A. R. Wilson, M. Vasile, C. A. Maddock, and K. J. Baker. Ecospheric life cycle impacts of annual global space activities. *Science of The Total Environment*, 834:155305, Aug. 2022. ISSN 0048-9697. doi:10.1016/j.scitotenv.2022.155305. URL <https://www.sciencedirect.com/science/article/pii/S0048969722023981>.
- [25] J. D. DeSain, B. B. Brady, and A. F. S. Command. Potential Atmospheric Impact Generated by Space Launches Worldwide—Update for Emission Estimates from 1985 to 2013. 2014.
- [26] A. R. Wilson. Estimating the CO₂ intensity of the space sector. *Nature Astronomy*, 6(4):417–418, Apr. 2022. ISSN 2397-3366. doi:10.1038/s41550-022-01639-6. URL <https://www.nature.com/articles/s41550-022-01639-6>.
- [27] M. Ross, D. Toohey, M. Peinemann, and P. Ross. Limits on the Space Launch Market Related to Stratospheric Ozone Depletion. *Astropolitics*, 7(1): 50–82, Mar. 2009. ISSN 1477-7622. doi:10.1080/14777620902768867. URL <https://doi.org/10.1080/14777620902768867>. Publisher: Routledge eprint: <https://doi.org/10.1080/14777620902768867>.

-
- [28] J. Dallas, S. Raval, J. Alvarez Gaitan, S. Saydam, and A. Dempster. The environmental impact of emissions from space launches: A comprehensive review. *Journal of Cleaner Production*, 255:120209, May 2020. ISSN 09596526. doi:10.1016/j.jclepro.2020.120209. URL <https://linkinghub.elsevier.com/retrieve/pii/S0959652620302560>.
- [29] I. W. Kokkinakis and D. Drikakis. Atmospheric pollution from rockets. *Physics of Fluids*, 34(5):056107, May 2022. ISSN 1070-6631. doi:10.1063/5.0090017. URL <https://aip.scitation.org/doi/10.1063/5.0090017>.
- [30] E. J. L. Larson, R. W. Portmann, K. H. Rosenlof, D. W. Fahey, J. S. Daniel, and M. N. Ross. Global atmospheric response to emissions from a proposed reusable space launch system. *Earth's Future*, 5(1):37–48, 2017. ISSN 2328-4277. doi:10.1002/2016EF000399. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/2016EF000399>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/2016EF000399>.
- [31] M. N. Ross and P. M. Sheaffer. Radiative forcing caused by rocket engine emissions. *Earth's Future*, 2(4):177–196, 2014. ISSN 2328-4277. doi:10.1002/2013EF000160. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/2013EF000160>.
- [32] M. Ross, M. Mills, and D. Toohey. Potential climate impact of black carbon emitted by rockets. *Geophysical Research Letters*, 37(24), 2010. ISSN 1944-8007. doi:10.1029/2010GL044548. URL <https://onlinelibrary.wiley.com/doi/abs/10.1029/2010GL044548>.
- [33] A. Poubeau. *Simulation des émissions d'un moteur à propergol solide : vers une modélisation multi-échelle de l'impact atmosphérique des lanceurs*. phd, Université de Toulouse, Université Toulouse III - Paul Sabatier, Feb. 2015. URL <http://thesesups.ups-tlse.fr/2742/>.
- [34] L. DeLuca, L. Galfetti, F. Maggi, G. Colombo, L. Merotto, M. Boiocchi, C. Paravan, A. Reina, P. Tadini, and L. Fanton. Characterization of HTPB-based solid fuel formulations: Performance, mechanical properties, and pollution. *Acta Astronautica*, 92:150–162, Dec. 2013. doi:10.1016/j.actaastro.2012.05.002.
- [35] M. R. Denison, J. J. Lamb, W. D. Bjorndahl, E. Y. Wong, and P. D. Lohn. Solid rocket exhaust in the stratosphere - Plume diffusion and chemical reactions. *Journal of Spacecraft and Rockets*, 31(3):435–442, May 1994. ISSN 0022-4650. doi:10.2514/3.26457. URL <https://arc.aiaa.org/doi/10.2514/3.26457>. Publisher: American Institute of Aeronautics and Astronautics.
- [36] C. Mutel. Brightway: An open source framework for Life Cycle Assessment. *Journal of Open Source Software*, 2(12):236, Apr. 2017. ISSN 2475-9066. doi:10.21105/joss.00236. URL <https://joss.theoj.org/papers/10.21105/joss.00236>.
- [37] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: an open-source framework for multidisciplinary design, analysis, and

- optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, Apr. 2019. ISSN 1615-1488. doi:10.1007/s00158-019-02211-z. URL <https://doi.org/10.1007/s00158-019-02211-z>.
- [38] M. A. J. Huijbregts, Z. J. N. Steinmann, P. M. F. Elshout, G. Stam, F. Verones, M. D. M. Vieira, A. Hollander, M. Zijp, and R. van Zelm. ReCiPe2016: a harmonised life cycle impact assessment method at midpoint and endpoint level. 2016.
- [39] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi:10.1109/4235.996017.
- [40] J. Blank and K. Deb. pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8:89497–89509, 2020.
- [41] M. J. D. Powell. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. In S. Gomez and J.-P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, pages 51–67. Springer Netherlands, Dordrecht, 1994. ISBN 978-94-015-8330-5. doi:10.1007/978-94-015-8330-5_4. URL https://doi.org/10.1007/978-94-015-8330-5_4.
- [42] E. J. Adler and J. R. R. A. Martins. Efficient Aerostructural Wing Optimization Considering Mission Analysis. *Journal of Aircraft*, Dec. 2022. ISSN 1533-3868. doi:10.2514/1.c037096. Publisher: American Institute of Aeronautics and Astronautics.
- [43] B. J. Brelje and J. R. R. A. Martins. Development of a Conceptual Design Model for Aircraft Electric Propulsion with Efficient Gradients. In *Proceedings of the AIAA/IEEE Electric Aircraft Technologies Symposium*, Cincinnati, OH, July 2018. doi:10.2514/6.2018-4979.



FIXED-WING UAV MODEL IDENTIFICATION FOR LONGITUDINAL MOTION USING FIRST-ORDER MODELS AND LIMITED FLIGHT TESTING

Nuno M. B. Matos^{1,2*} and André C. Marta²

1: Research and Development
Tekever UAS
Tekever

2500-750 Caldas da Rainha
nuno.matos@tekever.com, www.tekever.com

2: IDMEC

Instituto Superior Técnico
Universidade de Lisboa
1049-001 Lisboa, Portugal

nuno.matos, andre.marta@tecnico.ulisboa.pt

Abstract. *System identification plays an important role in the determination of an aircraft behaviour that helps predict and simulate different scenarios crucial for control, mission or safety assurance analysis. This work describes the system identification process of a medium sized UAV through the usage of limited flight test data and a non-linear model dynamic simulator. The proposed solution uses parameter based first-order models to describe the various aerodynamic properties of the UAV. The parameter estimation is based on a least square error optimization algorithm in a time-domain formulation starting from a low-fidelity aerodynamic analysis solution. The work focuses on the longitudinal motion by using routine flight test data of pitch down and pitch up manoeuvres to excite the longitudinal dynamics. The optimization geared towards parameter tuning used a combination of pitch and altitude UAV model response as measure of accuracy. Very significant improvements in the UAV model response are obtained with the resulting optimizer found relevant longitudinal aerodynamic and control derivatives. The pitching moment derivatives proved to be the most important parameters, as expected. The process hereby described is meant to be usable on any fixed-wing UAV with limited planned flight test data achieving reasonable accuracy.*

Keywords: aircraft design, optimization, aerospace, system identification, aircraft dynamics

1 INTRODUCTION

System identification is a field of study that encompasses the search for the optimized characterization of a system, from which, a simulated system can be created to not only explore simulation environments but also gather important information about the system qualities. In aircraft design, it can be used to determine its dynamic characteristics. By using different identification schemes, one can develop a model to either simulate flight conditions or evaluate the overall static and dynamic stability of the aircraft.

Work has been done in this field and several different approaches have been used to properly set a system identification methodology. This includes pure geometric based system identification [1], classical flight testing analysis used to do comparison between model and flight data using non-linear dynamic models [2–6], studies on the usage of big data and artificial neural networks [7, 8], and machine learning techniques [9] that train a model to behave as the real one.

Furthermore, extensive knowledge and studies have been produced on the best ways to conduct flight testing experiments for a simpler and direct construction of a classical aircraft dynamic model [10–12] where parameter estimation plays an important role. In a similar manner, the data acquisition implications also play an important role to build a complete system identification routine with a high degree of accuracy.

This work focuses on the longitudinal dynamic model characterization and optimization through the usage of limited flight data for a medium sized fixed-wing UAV (Tekever AR5 [13]). Due to the fast growing market and demands, little can be known about medium-sized UAV's, especially when flight testing is incompatible with product demands due to its slow paced data acquisition and treatment.

Here, a solution is presented to cover the gaps in the system identification of an aircraft with the usage of only routine factory flight verification data. Using consecutive pitch up and pitch down manoeuvres (used in structural flight testing), a dynamic model constructed is used to optimize and minimize the error between observable data and simulated data.

Due to data sensitivity of the studies conducted to Tekever, the results are shown in non-dimensional form. However, they are believed to be reproducible on any aircraft with similar low flight database.

2 BACKGROUND

For a contextualization of the work, a brief review of the most prominent subjects is presented.

2.1 Fixed Wing Aircraft Equations of Motion

The equations of motion that describe the behaviour of a rigid body motion in a 6 DOF system in time can be expressed as

$$\dot{X} = f(X, U) \quad , \quad (1)$$

where X , U and f are, respectively, the state vector, control vector and the vector-valued nonlinear function of the individual states and controls [14].

The foundation for the formulation in flight mechanics is Newton's second law [15]. It states, for translational motion, that the sum of external forces \vec{F} is equal to the change

of linear momentum and, for rotational motion, that the sum of external moments \vec{M} is equal to the rate of change of angular momentum,

$$\vec{F} = \frac{d}{dt}(m\vec{U})_E = \frac{d}{dt}(m\vec{U})_B + \vec{\Omega} \times (m\vec{U})_B \quad (2)$$

$$\vec{M} = \frac{d}{dt}([I]\vec{\Omega})_E = \frac{d}{dt}([I]\vec{\Omega})_B + \vec{\Omega} \times ([I]\vec{\Omega})_B \quad (3)$$

where m , \vec{U} , $[I]$ and $\vec{\Omega}$ represent the mass, velocity vector, inertia tensor and angular velocity of the rigid body, respectively. The subscripts E and B refer to the Earth and body (aircraft) references frames.

The notation used for the position, aircraft velocity, aircraft angular velocity, forces and moments, is respectively,

$$R^E = \begin{bmatrix} x^E \\ y^E \\ z^E \end{bmatrix}, \quad U^B = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \quad \Omega^B = \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \quad F^B = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad M^B = \begin{bmatrix} L \\ M \\ N \end{bmatrix}$$

An inertial Earth-fixed frame is used where the x , y and z axes are oriented towards North, East and Down, respectively. For the aircraft, a body-fixed frame is used where the body x , y and z axis point towards the nose, right wing and the aircraft bottom, respectively. The model used neglects the structural dynamic behaviours (rigid-body assumption) and uses the complete 6 degrees of freedom equations - the attitude or kinematic equations, which provide the aircraft rotation rate equations as

$$\begin{cases} \dot{\phi} = p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} = q \cos \phi - r \sin \phi \\ \dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta} \end{cases}, \quad (4)$$

where ψ , θ and ϕ are the Euler angles that correspond to yaw, pitch and roll, the force equations, which describe the translational motion as

$$\begin{cases} X - mg \sin \theta = m(\dot{u} + qw - rv) \\ Y + mg \sin \phi \cos \theta = m(\dot{v} + ru - pw) \\ Z + mg \cos \phi \cos \theta = m(\dot{w} + pv - qu) \end{cases}, \quad (5)$$

and the moment equations, which describe the rotational motion assuming a symmetric aircraft (xz-plane), as

$$\begin{cases} L = I_{xx}\dot{p} + I_{xz}\dot{r} + (I_{zz} - I_{yy})qr + I_{xz}pq + h_zq - h_yr \\ M = I_{yy}\dot{q} + (I_{xx} - I_{zz})rp + I_{xz}(r^2 - p^2) + h_xr - h_zp \\ N = I_{xz}\dot{p} + I_{zz}\dot{r} + (I_{zz} - I_{xx})pq + -I_{xz}rq + h_y p - h_x q \end{cases} \quad (6)$$

The aircraft model is constructed through the description of the forces and moments due to the state variables, which is the description of the force and moment coefficients. Generally, the coefficients are a non-dimensional characterization of the forces and moments, as

$$C_X = \frac{X}{q_d S}, \quad C_Y = \frac{Y}{q_d S}, \quad C_Z = \frac{Z}{q_d S}, \quad C_l = \frac{L}{q_d S b}, \quad C_m = \frac{M}{q_d S c}, \quad C_n = \frac{N}{q_d S b}, \quad (7)$$

where $q_d = \frac{1}{2}\rho V^2$, S , c and b represent the dynamic pressure, reference area, reference chord and reference span, respectively. The pursuit of this work is the identification of the derivatives of these coefficients with respect to the state variables.

2.2 System Identification

System identification is the process of determining the mathematical model of a real system that best predicts the outputs of a system given certain inputs [10]. Generally, this process is composed by four elements [16]: data collection, selection of the model structure, selection of the system Identification method and optimization of the model.

For the data collection, different types of sensors and data acquisition techniques can be developed and used to better retrieve the needed dataset for the best possible validation and/or creation of a model. Generally, the data collection aims to retrieve the data in a time domain or frequency domain. Knowing *a priori* an approximation of the dynamic characteristics of the UAV can help excite the needed frequencies to determine the natural dynamic response of the system, enabling a better estimation of the model. A time domain analysis is, however, simpler and more direct to use.

The model structure is the mathematical principles used to build the computational model aimed to predict the behaviour of the real system. It is one of the most important decisions as choosing an incorrect model can lead to a partially described system [16]. Several works have been done with different model approaches for the description of the system: linear force and moment coefficient models with non-linear dynamic models [2–6], inclusion of structural deformation due to flexible wings [17] and a physics-based neural network approach [8]. This work focuses on the usage of a non-linear dynamic model with polynomial based equations to describe the influence of the state variables in the forces and moments.

The system identification method will determine the parameter estimation error usage. Common methods for aircraft estimation include the output-error method and the equation-error method [10]. The equation-error typically uses a least-squares regression to minimize the difference between forces and moments computed through the data acquired and the ones predicted by the model. On the other hand, the output-error method minimizes the weighted sum of squared differences between measured outputs and model outputs. This work uses a version of the latter with a choice on the observable variables as well as the weight given to each state variable output-error.

The final step is the optimization and validation model. Using the chosen model and identification method, the model is incrementally refined until the desired response accuracy is achieved. This process can involve the usage of different flight test data and optimization routines for an automatic correction of the model parameters. Figure 1 shows a summarized overview of the system identification process used in this work.

3 PRELIMINARY AIRCRAFT IDENTIFICATION

An initial model is created using low-fidelity aerodynamic tools, propulsion gathered data treatment and point mass inertia assessment. The aerodynamic and control co-

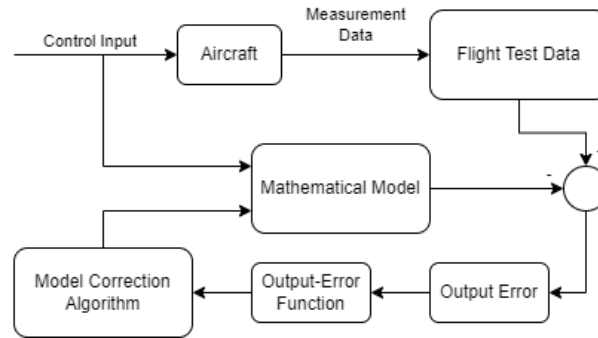
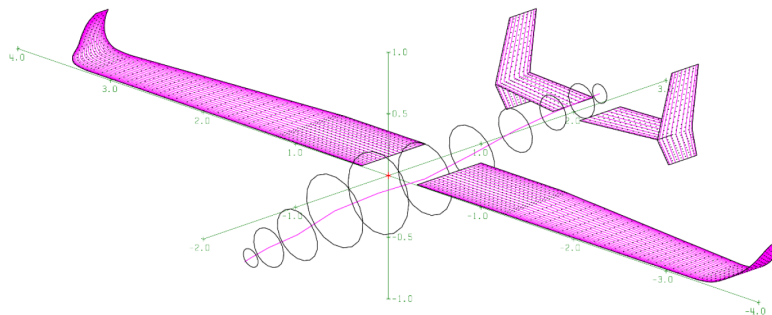


Figure 1: System identification methodology.

efficients in this model will then be optimized to achieve a better identification of the UAV.

3.1 Aerodynamics

The first aerodynamic estimate is provided using Athena Vortex Lattice (*AVL*) [18] analysis through the usage of vortice lattice method (*VLM*). This aerodynamic model is based on potential flow theory and, thus, is an initial low-fidelity estimation of the aircraft properties. Figure 2 previews the aircraft in the *AVL* software. A gapped wing was used to better represent the lift reduction zone due to the fuselage presence.


Figure 2: Tekever AR5 model in *AVL*.

The purpose is to build a longitudinal (and lateral) model of the aircraft to characterize the dependence of aerodynamic forces and moments with the longitudinal (and lateral) dynamic and control variables.

From an aerodynamic perspective, it is convenient to use the stability frame, in which the x axis points towards the longitudinal velocity, the y axis towards the right wing and the z axis following the right hand rule. To transform the lift C_L and drag C_D coefficients from the Earth-fixed frame to the stability frame, assuming zero heading ψ and roll ϕ ,

$$\begin{bmatrix} C_X \\ C_Z \end{bmatrix} = \begin{bmatrix} -\cos(\theta) & -\sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix} \begin{bmatrix} C_D \\ C_L \end{bmatrix}, \quad (8)$$

In this work, each force and moment coefficient is described as a sum of functions for each state variable and control position. For the longitudinal motion, a generic aerodynamic force or moment coefficient is written as

$$C_A = f_0(\alpha) + f_1(\dot{\alpha}) + f_2(q) + f_3(\delta_e) + f_4(\delta_t) \quad , \quad (9)$$

where each function f_i can be presented as a linear model for simplicity, or more complex, such as higher order polynomial or non-polynomial functions.

3.2 Propulsion

The Tekever AR5 has a twin piston engine configuration where both engines rotate in the clockwise direction when seen from the aircraft tail. To model these, the engine and propeller modules in *JSBSim* were used along with a correction of the propeller curves for both power and thrust. This model includes the induced moments generated by the engine and propeller rotations.

The lack of wind tunnel data created a gap in the understanding of the engine and propeller behaviour under a free stream velocity. Thus, little can be known other than the static thrust. However, using initial flight data for different flight regimes (cruise, climb and descent), different working points for the propulsion unit can be particularly analysed. This, coupled with the aerodynamic forces characterization (drag) can be sufficient to build a propulsion model that, for the same conditions of flight, produces the same outputs.

Using simple trimmed longitudinal flight conditions, and the coefficients for thrust force $C_T = \frac{T}{q_d S}$ and weight $C_W = \frac{W}{q_d S}$, the equations of motion for the translation in the longitudinal x and vertical z aircraft axes lead to

$$\begin{cases} C_X \sin(\theta) - C_Z \cos(\theta) + C_T \sin(\theta) - C_W = 0 \\ C_X \cos(\theta) - C_Z \sin(\theta) + C_T \cos(\theta) = 0 \end{cases} \quad (10)$$

By using the aerodynamic model, one can determine the lift C_L and drag C_D coefficients, from which, the C_X and C_Z force coefficients can be determined using Eq.(8). Knowing the weight W of the UAV, the thrust coefficient C_T generated can then be calculated.

For each data point available and knowing the amount of fuel consumed, this approach can lead to an accurate propulsion model deeply connected to the UAV aerodynamic model. However, the propulsion model is only correct if the aerodynamic assumptions are exact. Nonetheless, this assures that the thrust response will be the same in terms of power consumption and RPM. Furthermore, it assures that, for the same data points used, the same equilibrium would be found with the model.

This model is ultimately encapsulated in two propeller curves - the power C_P and thrust C_T coefficient curves, used by the *JSBSim* model to calculate both thrust and power as well as all the consequent forces and moments generated by each engine unit.

3.3 Inertia

The inertia of the aircraft is built with two different types of mass data: fixed point masses and variable point masses. The first describe all static payload and the airframe weight, the latter describe those that are variable, such as fuel, or whose position change. The description of the aircraft in point masses assumes that the mass body inertia I_C about its centre of mass is negligible when compared to the parallel theorem component of the inertia, resulting for the moments of inertia

$$I_{ii} = I_{C_{ii}} + md_i^2 \approx md_i^2, \quad (11)$$

and for the products of inertia

$$I_{ij} = I_{C_{ij}} - md_i d_j \approx -md_i d_j \quad , \quad (12)$$

where m is the point mass and d_i and d_j represent the distance of the point mass to the i and j axis, respectively.

3.4 Dynamic Model

The dynamic model is derived from the *JSBSim* capabilities. This software allows the choice of the force and moment axes, in which any dependency can be formulated, from constant contributions to n-variable functions as described in Eq.(9).

Using the concept formulated in Section 3.1, each longitudinal moment and forces are described by setting a function for each state and control variable. Furthermore, the model allows for the definition of the inertia of the body. Meaning, each mass or inertia contribution can be fed into the model during optimization cycles. The lateral contributions of the system are neglected for this case study, as the interest lays on the longitudinal motion.

The model is effectively a non-linear 6 degree of freedom construction of the equations presented in Section 2.1. The way the software proceeds with each step calculation is shown in Figure 3, where, from the current state variable solution, each force and moment contribution is calculated and fed into the non-linear equations scheme, from which the state variables are retrieved and used once again to begin a new step calculation cycle.

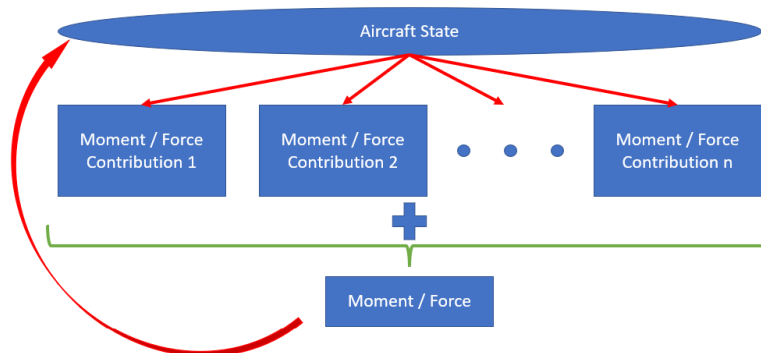


Figure 3: Force and moment calculations overview on *JSBSim*.

The *JSBSim* [19] software package includes Runge-kutta (RK) and Linear Multistep Methods (LMMs) to solve the dynamic system of equations. The default methods depend on the equation set being numerically integrated in time. Translational and rotational rate equations (Eqs.(5) and (6)) use a second-order Adams-Bashforth-Moulton (ABM) method

$$X_{ABM}(t + \Delta t) = X(t) + \Delta t \left[\frac{3}{2} \dot{X}(t) - \frac{1}{2} \dot{X}(t - \Delta t) \right] \quad , \quad (13)$$

and the kinematic equation (Eq.(4)) uses a second order RK method

$$\begin{cases} X_{RK}(t + \Delta t) = X(t) + \frac{1}{2}(k_1 + k_2) \\ k_1 = \Delta t f(X, t) \\ k_2 = \Delta t f(X + k_1, t + \Delta t) \end{cases} \quad . \quad (14)$$

The time step Δ_t was set to 0.0083 seconds, which corresponds to a frequency of $120Hz$, believed to capture sufficient detail of the dynamics of the system.

4 EXPERIMENT AND SIMULATION DESIGN

An important part of the problem is the experimental data available. In this work, routine structural test flight data is used as flight database for optimization purposes. The experiment consists of trimmed flight followed by pitch up and pitch down manoeuvres where the longitudinal motion is highly predominant. An example test layout is shown in Figure 4, where six plots are presented: on the top side, from left to right, the pitch θ , pitch rate q and estimated angle of attack α are presented; and on the bottom side, the elevator deflection delta from the initial trim state $\Delta\delta_e$, calibrated airspeed V_{CAS} and altitude h are presented.

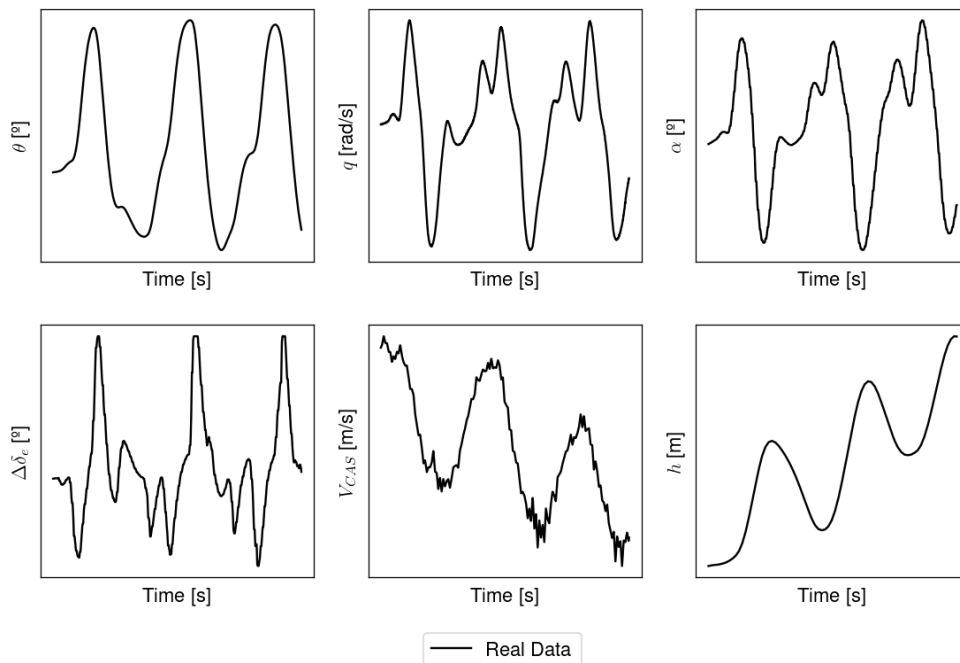


Figure 4: Structural test layout manoeuvre example for a generic Tekever AR5 aircraft.

4.1 Aircraft Data Collection

In this work, the data collection is heavily based on gyroscopic and pitot tube data to determine the angular position and speed of the aircraft, respectively. The UAV carries a data collection sensor system capable of retrieving longitudinal velocity, angular position, rates and accelerations. Angle of attack and sideslip angle estimators exist but their data is only used for data comparison and not error definition. The control data is also registered as output commands from the auto-pilot system, from which, the throttle command and elevator position can be deduced.

The autopilot works at $400Hz$, however, the control actuation is limited to $50Hz$. Thus, the control output of the autopilot is only evaluated at the control actuation frequency. In this work, the *JSBSim* simulation has a higher frequency than the control actuation, both to accurately represent it and capture more precisely the fast dynamic behaviours of the UAV. Thus, a linear interpolation is used between each control data point to evaluate,

for each simulation step, both the control positions as well as the state of the aircraft. These are used for error computation and control equivalence between simulation and real log data.

Although prone to error, the observable data is considered exact. The analysis of the degree of uncertainty in relation to a specific dataset is out of the scope of this work. Nonetheless, proven the method, an inclusion of certainty degree for each observable data enhances the system identification capabilities.

4.2 Simulation Development

This work focuses on the longitudinal simulation of the UAV based on the dynamic model created. For this, the control commands are fed into the model and the behaviour of the system is compared with the flight data available. Two important aspects arise from this analysis: the initial conditions set-up and the control input commands.

For the first, an approximation is used assuming the aircraft starts from a trimmed levelled flight condition. Using a simulated auto-pilot system, the simulated UAV is levelled and, after meeting certain convergence criteria, the experiment is started. The convergence criteria are based on the evaluation of the first derivative of a group of state variables, namely calibrated airspeed V_{CAS} , vertical velocity \dot{h} , pitch rate q , roll rate p and yaw rate r .

For the second, the available control input data consists of distributed time points of throttle or elevator commands logged by the system in a specific frequency that does not match the control output frequency. For this reason, as mentioned, a linear interpolation method is used to evaluate the value of each control in each simulation time step. For the throttle command, the important data is the RPM of the engines, from which the linear interpolation is used to calculate the simulated propeller RPM, thus producing, according to the proposed model, the same thrust and power consumption connected to the model. As for the elevator command, the important data is the elevator deflection difference from a trimmed elevator position.

To assure equal dynamic excitation on both the real and the simulated model, delta (Δ) control inputs are used instead of absolute control inputs. This methodology minimizes the trimmed state control input error and focuses on the delta impact on the overall dynamics of the UAV. Thus, the elevator command fed is the difference between the previous trimmed state condition. The throttle command, since it is validated with prior experimental data, will have similar responses for both delta and absolute implementations.

Since the position of the elevator is only known through an approximation - notice that the position is calculated by knowing the auto-pilot command and the mechanism itself (there is no sensor for the angle deflection) - most relevant data is obtained relative to the trim state, thus discarding the trim point convergence optimization for the pitch moment (C_{m_0} is constant throughout).

5 PARAMETER ESTIMATION RESULTS

In this section the parameter estimation results are discussed. First, the problem at hand is formulated and the algorithm used is validated using a test-case scenario with the convergence between a generic model the modified version of it. Then, the error function is formulated based on different definitions and the optimization problem is finally defined.

Single-manoeuvre and multi-manoeuvre optimizations are conducted where, in the latter, training and testing data sets are used to both fit the model and evaluate its generalized error based on a maximum likelihood error estimation.

5.1 Problem Formulation

This work uses the *JSBSim* software package to simulate a manoeuvre scenario with throttle and elevator control inputs equivalent to a real log data manoeuvre. The purpose is to verify the error between the constructed model and the real log data to better identify the system at hand.

For this purpose, a python-based framework was constructed to handle both the simulation calling as well as the data analysis between model and real data. This framework allows to additionally compare two *JSBSim* models and do optimization cycles to converge one into the other. This allowed for the validation of the method, which will be seen in Section 5.1.1.

The problem is the evaluation of the error between the calculated model data and the collected real log data. The error in this work is handled with four different variables - calibrated airspeed (V_{CAS}), altitude (h), pitch angle (θ) and pitch rate (q). These are used to estimate, through maximum likelihood methods, the similarities between the model and the real flight data. This work used the root mean squared error (RMSE) to calculate, for each variable, the associated error. This is further explored in Section 5.1.2.

Additionally, for the optimization, the *openMDAO* [20] tool was implemented within the python framework to efficiently handle the optimization process. The *SLSQP* (Sequential Least Squares Programming) optimizer was chosen along with first order finite differences method to address the objective function (RMSE) vector gradient with respect to the design variables.

The design variables considered can be found in Table 1. Note that throughout the parameter estimation analysis, some design variables were not included due to the results found in Section 5.2. Additionally, two variables are left out from the analysis, the zero angle of attack pitch moment coefficient (C_{m_0}) and all drag contributions. The former is left out since it only affects the initial trim point and the latter is not included since modifying the drag contributions would make the propulsion component unreliable.

The optimization loop is as follows: a first run is executed to determine the initial objective function value; the vector gradient is calculated through a first order finite difference method; the *SLSQP* calculates a first step towards modifying the design variables; the model is updated according to the new set of variables; the process is repeated until a certain convergence criterion is met. The optimization process can also be visualized in Figure 5, where τ represents the vector of design variables.

5.1.1 Validation

To prove the effectiveness of the method formulated, a *JSBSim* model identification using another *JSBSim* model was done for a modified set of initial design variables. The initial model is a generic model (the first iteration model from the aerodynamic, propulsion and inertia analyses data) from which modifications were made to verify if the optimizer could successfully return the model to its original state, both in design variable values, as well as dynamic behaviour. Figure 7 provides data on the error for the conducted analysis

Table 1: Design variables considered for the optimization problem.

Variable	Definition
C_{m_α}	Pitching moment coefficient partial derivative in relation to the angle of attack α
C_{m_q}	Pitching moment coefficient partial derivative in relation to the pitch rate q
$C_{m_{\dot{\alpha}}}$	Pitching moment coefficient partial derivative in relation to the derivative in time of the angle of attack $\dot{\alpha}$
$C_{m_{\delta_e}}$	Pitching moment coefficient contribution of the elevator control surfaces. In this work it is described as either a linear function or a quadratic function. Positive and negative deflections have different terms.
$C_{L_{\delta_e}}$	Lift contribution of the elevator control surfaces. In this work it is described as either a linear function or a quadratic function. Positive and negative deflections have different terms.
C_{L_α}	Lift coefficient partial derivative in relation to the angle of attack α
C_{L_0}	Zero angle of attack α lift coefficient
$C_{L_{\dot{\alpha}}}$	Lift coefficient partial derivative in relation to the derivative in time of the angle of attack $\dot{\alpha}$
C_{L_q}	Lift coefficient partial derivative in relation to the pitch rate q

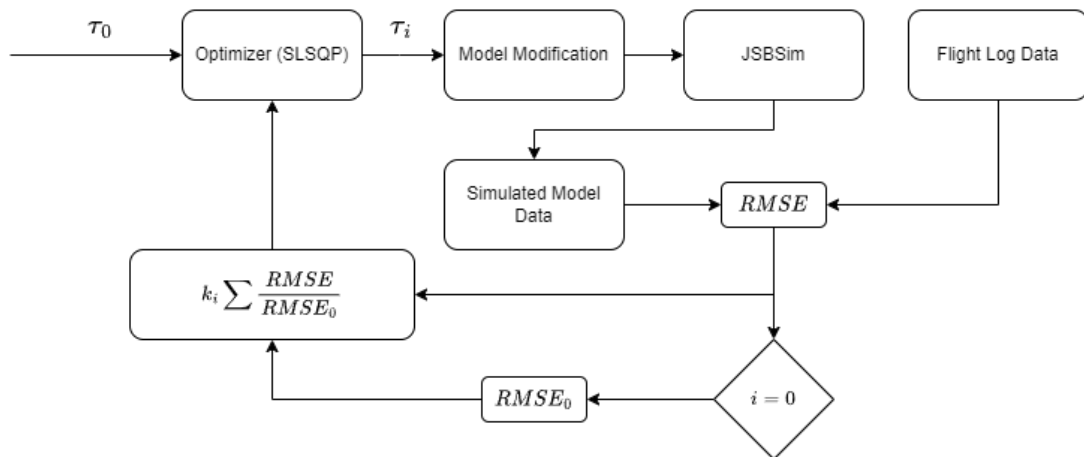


Figure 5: System identification optimization algorithm.

and Figure 6 provides a graphical overview of the conducted optimization obtained using a combined pitch, pitch rate, calibrated airspeed and altitude error formulation (see Section 5.1.2).

As seen, the optimized results converges to the same dynamic behaviour as the desired solution. The improvement is also clear as, for all shown curves, the real data and final solution data are equivalent.

5.1.2 Error Formulation

This work uses two different formulations to calculate the error between real and model data. For a single variable error, the formulation is the standard root mean square error (RMSE) normalized with the initial RMSE given as

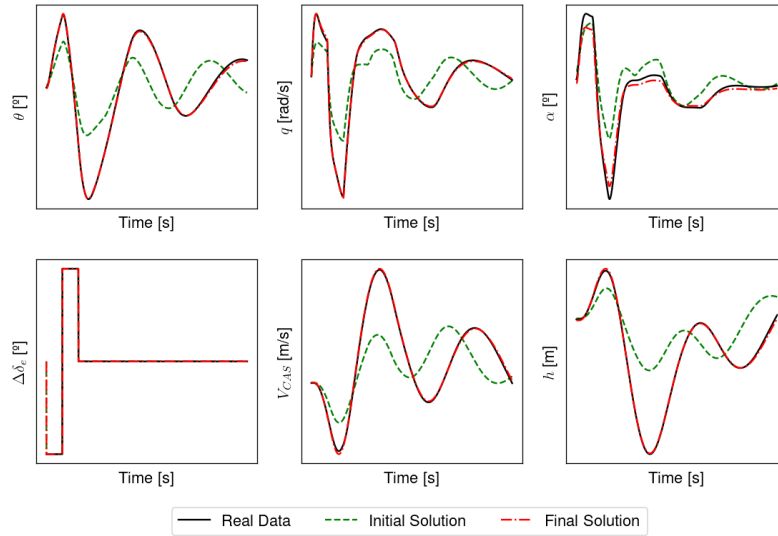


Figure 6: Methodology validation using two *JSBSim* models to prove the convergence of one towards the other.

$$E = \frac{RMSE_i}{RMSE_0} = \frac{\sqrt{\sum_{j=1}^N (y_j - \hat{y}_j)^2}}{\sqrt{\sum_{j=1}^N (y_{j_0} - \hat{y}_{j_0})^2}}, \quad (15)$$

where N is the number of data points in the manoeuvre, y_i the real values and \hat{y}_i the simulated model values, both for a desired error variable.

For a multiple variable error, this work uses an equal based weight formulation, where the error is the normalized sum of all errors (formulated as stated previously) for all variables intended,

$$E = \sum_i \frac{RMSE_{\tau_i}}{RMSE_{\tau_{i_0}}}, \quad (16)$$

where τ_i represents a generic designated error variable.

Additionally, two different approaches to the determination of y_i and \hat{y}_i were used. One classical method was to use the absolute value of the variable chosen for the error calculation

$$\delta_{y_i} = y_i - \hat{y}_i = \tau_i - \hat{\tau}_i \quad . \quad (17)$$

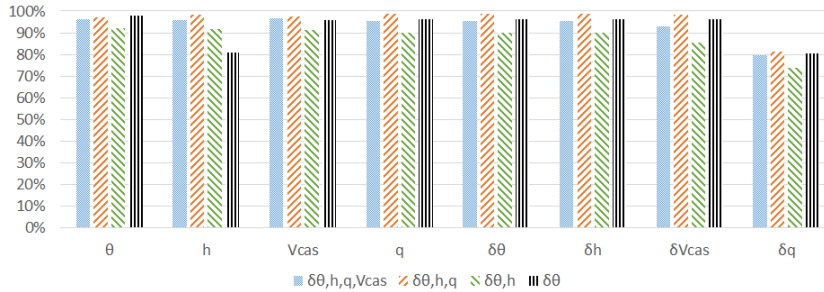
The other was to use the step difference between each state during the flight manoeuvre

$$\delta_{y_i} = y_i - \hat{y}_i = (\tau_{j+1} - \tau_j) - (\hat{\tau}_{j+1} - \hat{\tau}_j) \quad , \quad (18)$$

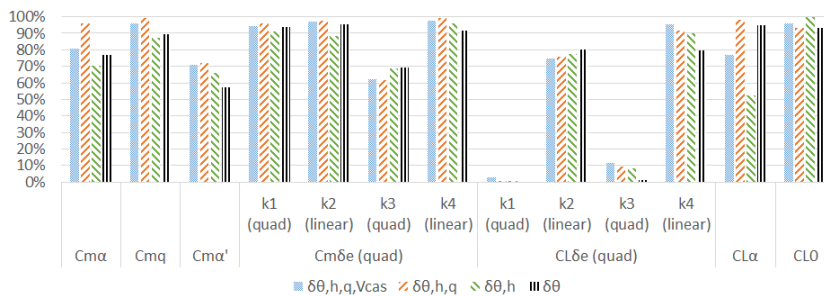
where, in this case, i represents the step index and j the state variable index. The first formulation, combined with the formulated error expressions, will evaluate the similarity between the absolute values of the state variables. The latter will evaluate the similarities between shapes.

To address which error formulation to use, different approaches were conducted using the aforementioned *JSBSim* model validation. Four error formulations were tested where each one was obtained by incorporating an additional variable, starting with the pitch step error δ_θ and then including altitude δ_h , pitch rate δ_q and calibrated airspeed $\delta_{V_{CAS}}$

errors. The results are compiled in Figure 7, where, for each optimization analyses, the results for all single variable errors are shown (Fig. 7a) and the error for each variable is also shown (Fig. 7b).



(a) Improvement in each single error score.



(b) Similarity between validation model variables and optimized model design variables.

Figure 7: Results for the different error formulations with the *JSBSim* validation scheme.

Results show that increasing the variables in the error formulation helps by averaging the improvement score in all variables, whilst having better single score in some cases. Note the improvement in the pitch step error score ($\delta\theta$) when the optimization error variables are the pitch step, altitude and pitch rate. Additionally, the similarity between real values and optimized values also improves due to the usage of a higher number of error sources. The only exception to improvement are the quadratic terms for the elevator surface lift coefficient. This can be due to their small influence in the overall longitudinal dynamics of the system being modelled, where all the other variables seem to have higher influence in the error minimization.

For these reasons and, since the real optimization problem is more complex, this work opted for the use of all error variables in the objective function formulation, where the pitch is used with the step error formulation and all the other variables (altitude, pitch rate and calibrated airspeed) are used with the absolute formulation. This augments the capture of all needed data for a best longitudinal model fit.

5.1.3 Problem Definition

Using the previous results as basis, the optimization problem is formulated as

$$\min \sum \frac{RMSE_{\tau_i}}{RMSE_{\tau_{i0}}}, \quad (19)$$

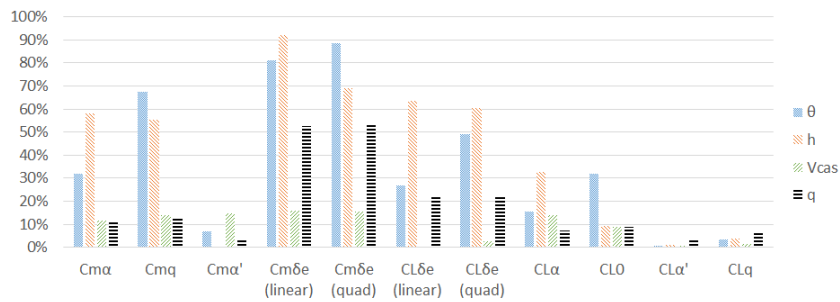
w.r.t f_{τ_i}

where the τ and f_{τ_i} are the error variables (θ , h , V_{CAS} and q) and the functions f for each design variable τ considered (as presented in Eq.(9)), respectively. The optimization algorithm is the *SLSQP* and the optimization parameters included a tolerance of $1e - 3$ and a maximum number of iterations of 200 for convergence criteria.

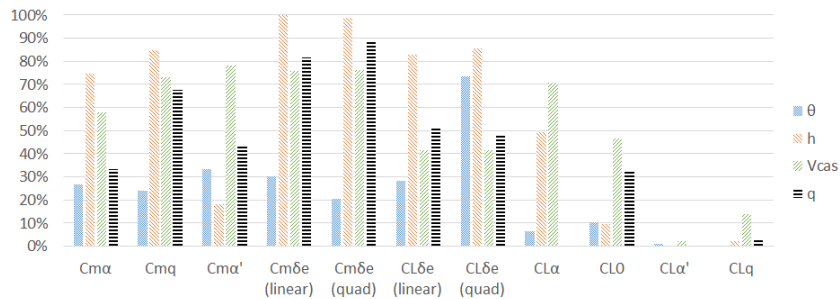
The following results use these convergence parameters and the same error formulation. The only exception is the sensitivity analysis of the design variables (Section 5.2), where the selection of the design variables for the final optimization problem is done using a single variable error approach.

5.2 Sensitivity Analysis

To assess the impact of the design variables on the optimization problem and system identification procedure, a sensitivity analysis was conducted. This analysis also allowed for the selection of the design variables for the parameter estimation optimizations conducted. For each variable and each type of single variable error considered, an optimization was conducted. The compilation of the results are presented in Figure 8, including the error improvements for each data variable using absolute (Fig. 8a) and step (Fig. 8b) error formulations in single variable optimizations.



(a) Using the absolute error formulation.



(b) Using the step error formulation.

Figure 8: Improvement in each error variable for all single variable optimization cases.

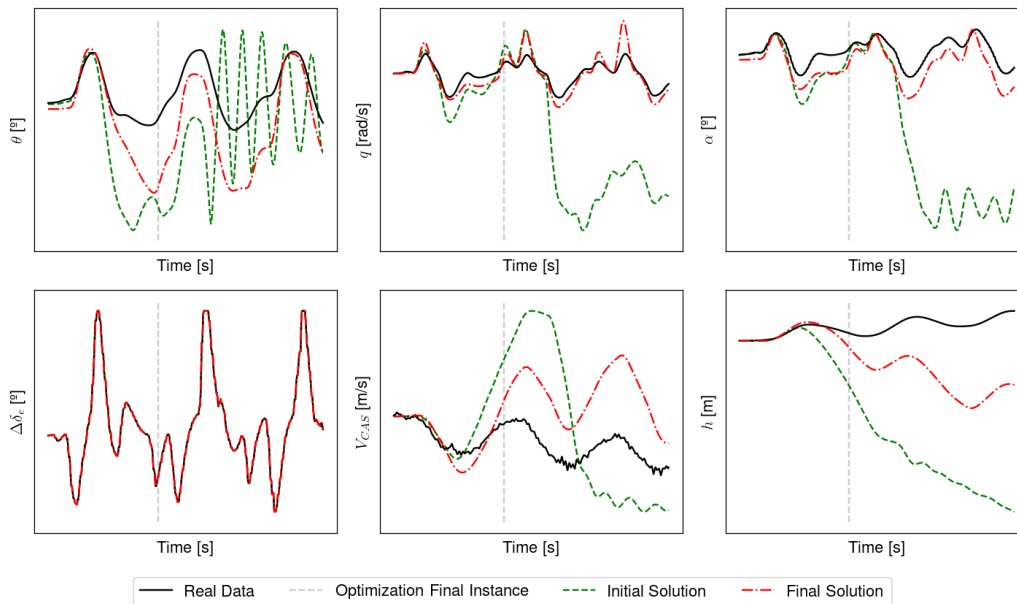
As seen, the optimization results depend not only on the designated variable but also on the type of objective (error) function used to evaluate the performance of each solution. In both cases, the variables $C_{m\alpha}$, C_{mq} , $C_{L\delta_e}$ and $C_{m\delta_e}$ seem to have more impact on the final results. Variables $C_{m\alpha'}$, $C_{L\alpha}$ and C_{L0} have sufficient impact on the optimization results to be considered in the optimization ($> 20\%$). As for the variables $C_{m\alpha}$ and C_{Lq} , their impact on the overall results is minimal, thus, these variables will be discarded.

These results also present the comparison of linear and quadratic representation of the

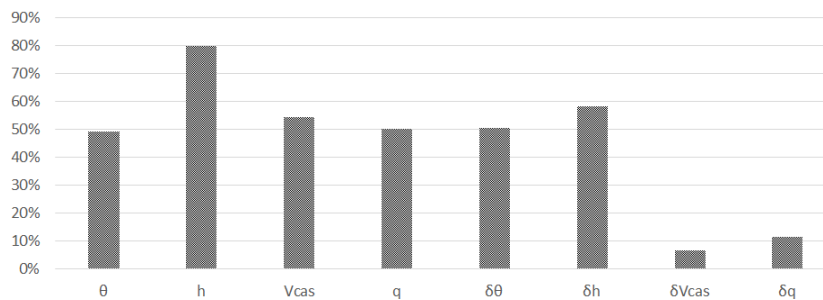
elevator δ_e influence on the aerodynamic forces. Since the quadratic option, in some cases, benefits the optimizer with increased performance, these coefficients will be used.

5.3 Single-Manoeuvre Optimization

Using the problem definition presented in Section 5.1.3 and the design variable choice from Section 5.2 the initially built model was optimized using a single flight test manoeuvre. Figure 9 shows the solution (Fig. 9a) and the error improvement (Fig. 9b).



(a) Single manoeuvre optimization graphical results.



(b) Single manoeuvre optimization error improvement score.

Figure 9: System identification using a single manoeuvre for the Tekever AR5.

Both the response and improvement scores show that the optimization was successful in modifying the model to reduce the overall error between log and simulated data. Clearly, the initial model was divergent and after a certain period of time it evolves into an unstable condition (high pitch oscillations, very low angle of attack and high altitude loss). On the other hand, the optimized solution, even if only optimized during a single cycle of the manoeuvre, is capable of not only approximate the dynamics, but also maintain the aircraft within acceptable margins of the real values found. The largest improvements were found for pitch and altitude where the error was also maximum on the initial solution.

5.3.1 Solution Application

A single manoeuvre optimization is prone to error due to the lack of variability and different condition manoeuvres. For this reason, first, the solution found is tested with two example flight test manoeuvres with the same UAV. Figure 10 yield the results for the manoeuvres tested (10a and 10b), where the initial solution provided to the optimizer is also presented to verify the improvements visually.

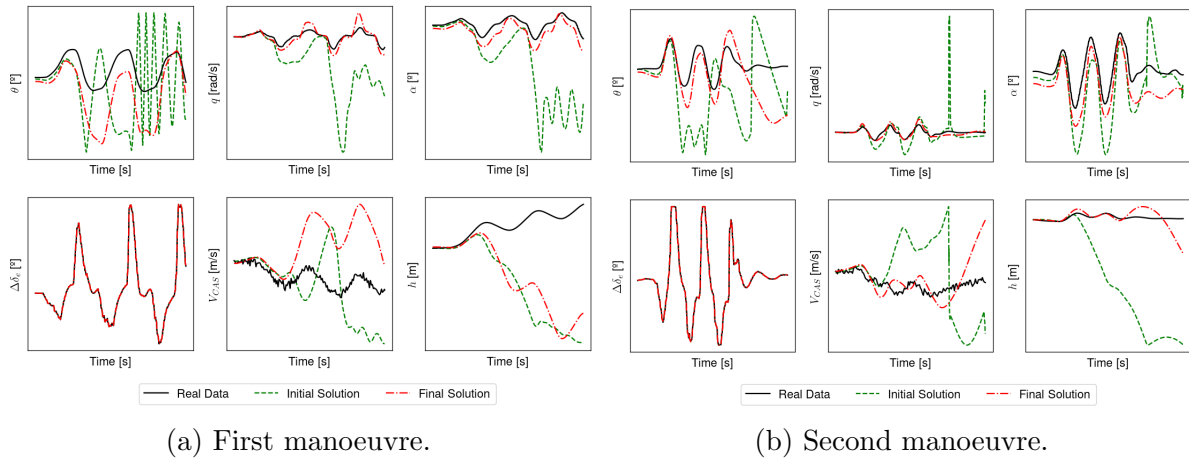


Figure 10: Single manoeuvre optimization validation with two separate independent manoeuvres of the same Tekever AR5 aircraft.

Similarly to the optimized manoeuvre case, the initial and final solution have different behaviours. The former quickly diverges for the real solution whereas the latter is capable of maintaining the dynamic of the real UAV. The second manoeuvre seems to be well fitted, however, for the first manoeuvre, both altitude and calibrated airspeed results have larger differences and seem to be more prone to error as the simulation time increases.

Nonetheless, the optimization was capable of modifying the model to be more coherent with the real log data and also maintain stability in multiple cycles of the same manoeuvre, reducing the gap between the modelled and real systems.

For a statistical analysis, the optimized solution was simulated with eight additional different similar manoeuvres and the mean and standard deviation of the error was calculated, as shown in Figure 11, for the used and observable error data: pitch (Fig. 11a), pitch rate (Fig. 11b), altitude (Fig. 11c) and calibrated airspeed (Fig. 11d).

As shown, the optimized solution improves the modelled dynamics by not only having a lower mean value for the majority of the errors, but also lower standard deviations, meaning the output of the optimized model is more accurately representing the real log data found. Additionally, some error cases are better fitted than the optimized manoeuvre scenario (for example comparing manoeuvre #3 with the used in optimization manoeuvre #1). This means that the optimized solution is capturing the general longitudinal dynamic qualities of the real aircraft. However, the mean error and standard deviations are not ideal, where room for improvement still exists.

Expectedly, overall results vary from the single manoeuvre optimization shown in Section 5.3. To further increase the performance of the method, an increased number of observable data is expected to augment the system identification capabilities by introducing new dynamics of the same aircraft and reducing the case specific elements of the

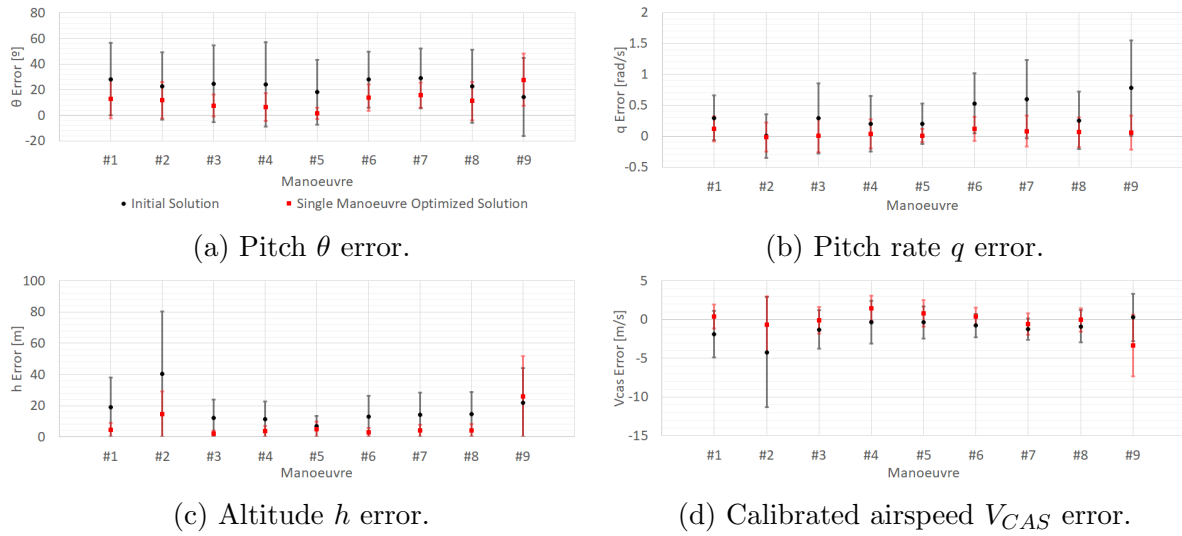


Figure 11: Average error and standard deviation of the error results for nine manoeuvres for the initial and final solution of the single manoeuvre optimization. Manoeuvre #1 was the used manoeuvre for the optimization.

single manoeuvre optimization.

5.4 Multi-Manoeuvre Optimization

To mitigate the single manoeuvre optimization case specific nature, a multi-manoeuve approach to the problem was adopted. To achieve this, multiple flight test manoeuvres were collected and used to better fit the parameter estimated model.

5.4.1 Model Formulation

The combined manoeuvre optimization needs two types of data subsets for the proper validation of the model. Using as reference the surrogate modelling theory [21], from all available manoeuvres, 75% are used as training data and 25% are used as validation data. The former is fed into the optimization problem and is used to calculate the error between the simulated model and the real flight data. For this, the error function was modified to

$$E = \sum_{m=1}^M \sum_i \frac{RMSE_{\tau_i}}{RMSE_{\tau_{i_0}}} , \quad (20)$$

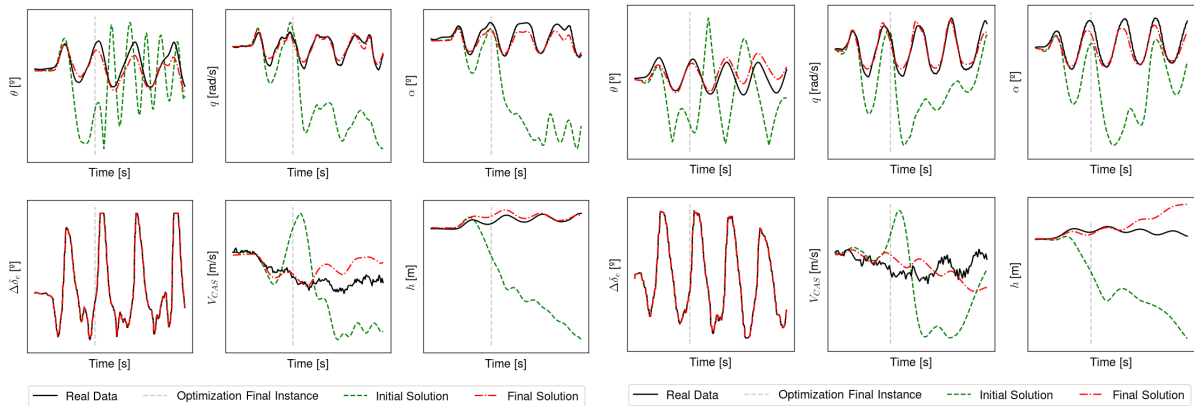
where m is the manoeuvre counter, to account for all manoeuvres in an equal weighted manner. The latter is used to prove the new found optimized model, where the model is tested with the validation data to check whether the optimization was successful.

All the manoeuvres collected correspond to equivalent flight test procedures with the same aircraft. The training and validation datasets division was done at random, where of a total of nine manoeuvres, six were selected for training and the remaining three were used for validation.

5.4.2 Optimization Results

Following the aforementioned optimization scheme, the initial model solution was optimized using six manoeuvres from a total of nine. Figure 12 presents two of these manoeeu-

vres (#4 in (12a) and #7 in (12b)) with both the initial and final solutions presented.



(a) Manoeuvre #4.

(b) Manoeuvre #7.

Figure 12: Multi-manoevre optimization results plots for two example manoeuvres used in the optimization of the same Tekever AR5 aircraft.

Clearly, the final solution found by the optimizer is not only closer to the expected dynamics of the real flight data, but also has more stability and capability to diverge less when above the used optimization time for the manoeuvre. In the first example shown (12a), the final solution correctly models both pitch and altitude response for the shown four cycles of the nose up and nose down manoeuvres. However, the second example previews an altitude deviation as the flight time increases.

Comparing to the single-manoevre optimization solution, these results show that the increased number of manoeuvres inside the optimization loop revealed useful data to the algorithm that allowed for a better dynamic response convergence. The algorithm used was successful in reducing the gap between modelled and real UAV responses, diminishing the error for the system identification, for all the used manoeuvres inside the optimization process.

5.4.3 Model Validation

As seen, the final solution obtained was capable of suitably predicting the real dynamics of the aircraft, specially in the optimization flight time period used, where above this time the response of the model can deviate. However, this analysis was only for the used manoeuvres on the optimization process. To finally validate the final solution, the three left out manoeuvres for the validation process are analysed.

First, an error analysis was conducted. Figure 13 presents the mean error and standard deviations of all found solutions - initial, single manoeuvre and multi-manoevre - for all used observable error data: pitch (13a), pitch rate (13b), altitude (13c) and calibrated airspeed (13d). The marked manoeuvres #2, #5 and #6 are the ones used for validation purposes and the remaining were used as error source for the optimization.

Compared to Fig. 11, there is a decrease in both the mean and standard deviation errors for the final solution of the multi-manoevre optimization cases, where, for the used observable data, the final solution model fits well with the real flight log data. The only exception to the results found was the manoeuvre #9, where both addressed error

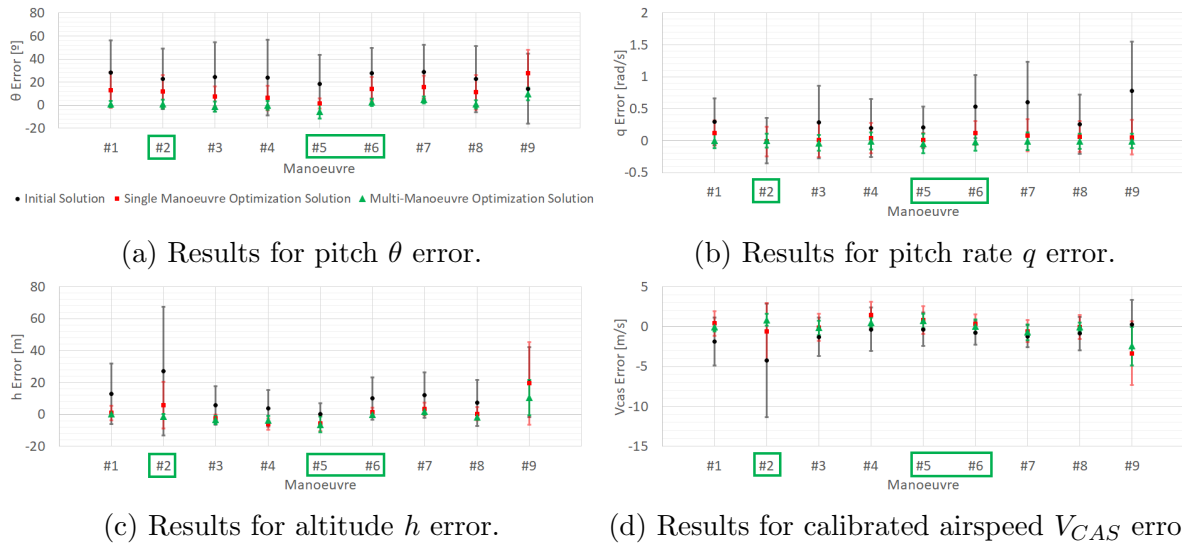


Figure 13: Average error and standard deviation of the error results for nine manoeuvres for the initial and final solutions of the multi-manoevre optimization.

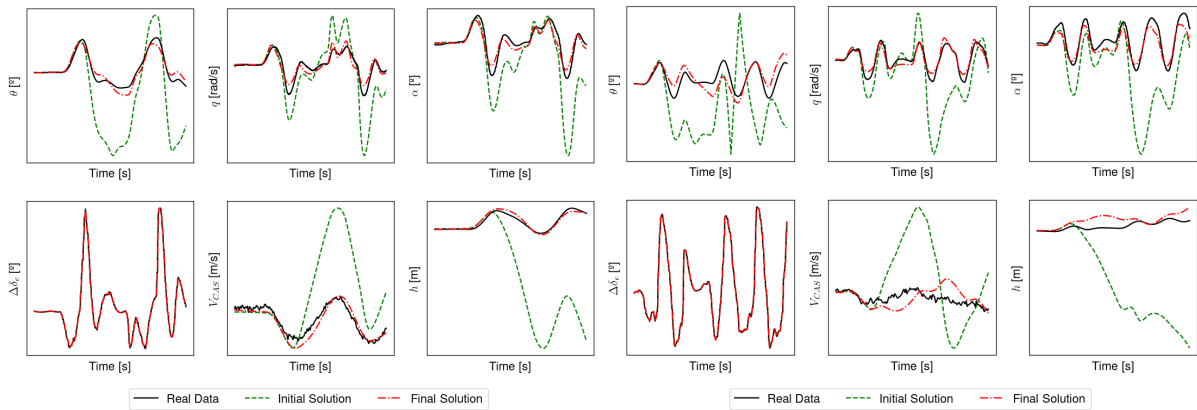
metrics were maximum. This can be due to poor flight log data of that manoeuvre or the initial conditions coherence.

Nonetheless, taking a closer look into the validation data (manoeuvres #2, #5 and #6), the final solution was able to accurately capture the real dynamics of the aircraft using single cycle portions of flight data from the other similar manoeuvres. Furthermore, the average and standard deviation error values are similar to the ones found for the optimized manoeuvres, thus, the qualities captured within the final solution model are representing the real flight log data. For a deeper look into the validation results, Figure 14 yields the response for of each manoeuvre using the initial, real and final solutions for comparison.

As seen, the final solution results have similar dynamics in relation to the real flight data in the duration of the first manoeuvre cycle and also maintain a reasonable accuracy as the number of cycles increases. The available data clearly shows an improvement in the model dynamic response and assures equivalence for the initial moments of the manoeuvre. Thus, the validation of the algorithm and final solution has found that the multi-manoevre optimization approach is able to accurately capture the real UAV dynamics with enough broadness that similar manoeuvres can be evaluated without recurring to new optimization analysis. The model can be used to test different manoeuvres or evaluate the aircraft longitudinal dynamics.

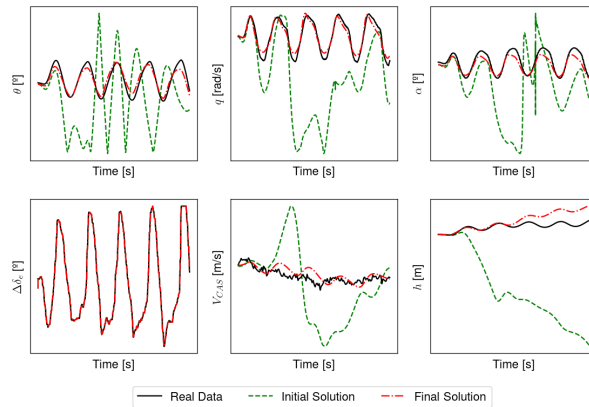
6 CONCLUSIONS

This work focused on the development of a system identification optimization framework where a six degree non-linear dynamic model (formulated in *JSBSim* and based in simple polynomial functions) was successfully modified, using the *SLSQP* gradient algorithm, to accurately predict and emulate real flight longitudinal manoeuvres. Furthermore, it has been shown that the usage of a low number of manoeuvres was sufficient to accurately correct and modify the model towards an equivalent longitudinal response, at least for a single cycle of nose up and nose down manoeuvres. The usage of routine structural flight testing was shown to be sufficient for the optimization of a longitudinal model that estimates the longitudinal dynamic characteristics of the aircraft.



(a) Results for the manoeuvre #2.

(b) Results for the manoeuvre #5.



(c) Results for the manoeuvre #6.

Figure 14: Multi-maneuvre results using the three validation manoeuvres of the same Tekever AR5 aircraft.

The comparison between single and multi-maneuvre environments concluded that a broader use of manoeuvres successfully captures more accurately the dynamic response of a system. However, the single manoeuvre optimization has shown to decrease the error found in the initial solution and improve the response in other equivalent manoeuvres. The training/validation subset division has shown to be important for the correct evaluation and validation of a final solution, where the multi-maneuvre algorithm proved to be effective at determining the real UAV longitudinal qualities.

The usage of single cycle portions of complete manoeuvres was shown to be enough to correctly model the UAV and to acquire sufficient data to allow the final solution to reasonably follow n cycle manoeuvres.

ACKNOWLEDGEMENTS

The authors would like to thank Tekever for supporting the production of this article by offering all the needed data to successfully produce the analyses here shown. This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] A. Klöckner. Geometry based flight dynamics modelling of unmanned air-planes. *AIAA Modeling and Simulation Technologies Conference*, August 2013. doi:10.2514/6.2013-5154. Boston, Massachusetts, USA.
- [2] A. Kamal, A. M. Aly, and A. Elshabka. Tuning of airplane flight dynamic model using flight testing. *AIAA Modeling and Simulation Technologies Conference*, January 2015. doi:10.2514/6.2015-1595. Dallas, Texas, USA.
- [3] P. Lichota and M. Lasek. Aircraft system identification from flight data recorder data. *ML-XV Mechanika w Lotnictwie (ML-XV Mechanics in Aviation)*, May 2012. Warszawa, Poland.
- [4] R. Jategaonkar, D. Fischenberg, and W. von Gruenhagen. Aerodynamic modeling and system identification from flight data-recent applications at DLR. *Journal of Aircraft*, 41(4):681–691, 2004. doi:10.2514/1.3165.
- [5] S. J. Koeberle, A. E. Albert, L. H. Nagel, and M. Hornung. Flight testing for flight dynamics estimation of medium-sized UAVs. *AIAA Scitech 2021 Forum*, January 2021. doi:10.2514/6.2021-1526. Virtual Event.
- [6] B. M. Simmons, H. G. McClelland, and C. A. Woolsey. Nonlinear model identification methodology for small, fixed-wing, unmanned aircraft. *Journal of Aircraft*, 56(3): 1056–1067, 2019. doi:10.2514/1.C035160.
- [7] J. Harris, F. Arthurs, J. Henrickson, and J. Valasek. Aircraft system identification using artificial neural networks with flight test data. *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 679–688, June 2016. doi:10.1109/ICUAS.2016.7502624. Arlington, Virginia, USA.
- [8] T. Stachiw, A. Crain, and J. Ricciardi. A physics-based neural network for flight dynamics modelling and simulation. *Advanced Modeling and Simulation in Engineering Sciences*, 9(13), April 2022. doi:10.1186/s40323-022-00227-7.
- [9] R. Cao, Y. Lu, and Z. He. System identification method based on interpretable machine learning for unknown aircraft dynamics. *Aerospace Science and Technology*, 126:107593, July 2022. doi:10.1016/j.ast.2022.107593.
- [10] V. Klein and E. Morelli. *Aircraft System Identification: Theory and Practice*. AIAA education series. American Institute of Aeronautics and Astronautics, 2006. ISBN 9781563478321.
- [11] R. Jategaonkar. *Flight Vehicle System Identification: A Time Domain Methodology*. 2nd edition, 2015. ISBN 978-1-62410-278-3.
- [12] K.-S. Wang and K. Iliff. Retrospective and recent examples of aircraft parameter identification at NASA dryden flight research center. *Journal of Aircraft*, 41:752–764, July 2004. doi:10.2514/1.332.
- [13] Tekever. URL <https://www.tekever.com/>.

- [14] B. Stevens and F. Lewis. *Aircraft Control and Simulation*. Wiley, 1992. ISBN 9780471613978.
- [15] W. Phillips. *Mechanics of Flight*. Engineering case studies online. Wiley, 2004. ISBN 9780471334583.
- [16] N. Hoffer, C. Coopmans, A. Jensen, and Y. Chen. Small low-cost unmanned aerial vehicle system identification: A survey and categorization. *2013 International Conference on Unmanned Aircraft Systems*, May 2013. doi:10.1109/ICUAS.2013.6564775. Atlanta, Georgia, USA.
- [17] G. Dussart, V. Portapas, A. Pontillo, and M. Lone. Flight dynamic modelling and simulation of large flexible aircraft. In K. Volkov, editor, *Flight Physics*, chapter 3. IntechOpen, Rijeka, 2018. doi:10.5772/intechopen.71050.
- [18] M. Drela and H. Youngren. Athena vortex lattice. URL <https://web.mit.edu/drela/Public/web/avl/>.
- [19] J. Berndt. JSBSim: An open source flight dynamics model in C++. *AIAA Modeling and Simulation Technologies Conference and Exhibit*, August 2004. doi:10.2514/6.2004-4923. Providence, Rhode Island, USA.
- [20] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi:10.1007/s00158-019-02211-z.
- [21] A. Forrester, A. Sobester, and A. Keane. *Engineering Design Via Surrogate Modelling: A Practical Guide*. Progress in astronautics and aeronautics. Wiley, 2008. ISBN 9780470060681.



VALIDATION OF EXTENDED FAILURE MODELS AND CRITERIA FOR AEROSPACE COMPOSITES

Giuseppe Corrado¹, José Reinoso² and Albertino Arteiro^{1*}

1: DEMec
Faculdade de Engenharia
Universidade do Porto
Rua Dr. Roberto Frias, s/n, Porto, 4200-465, Portugal
up201811504@edu.fe.up.pt, aarteiro@fe.up.pt, <https://sigarra.up.pt/feup>

2: Group of Elasticity and Strength of Materials
School of Engineering
University of Seville
Camino de los Descubrimientos s/n, 41092 Seville, Spain
jreinoso@us.es, <https://www.us.es/>

Abstract. *In the aerospace sector, to cope with the most recent sustainability goals, aerostructures will need to be designed and built by using advanced materials with unprecedented efficiency. Furthermore, performance-based certification using virtual qualification and certification tools is expected to become widely applied. In this context, research in advanced composite materials and their failure analysis plays a crucial role for optimization of aerostructures. As part of the European Industrial Doctorate (EID) OptiMACS, focussed on the development and application of new methods and criteria for multidisciplinary optimization of aerospace composite structures, advanced failure criteria and damage models have been extended to more general stress states, and validated. They include advanced phenomenological failure models based on invariant structural tensors for first-ply failure (FPF) “hot-spot” analysis considering general three-dimensional (3D) stress states. The 3D invariant-based criteria are then used to generate “omni strain” failure envelopes that enable quick last-ply failure (LPF) analysis of multidirectional composite structures, showing a satisfactory agreement with experimental data under complex triaxial stress states. It is, therefore, demonstrated that “omni strain” failure envelopes provide reliable and fast laminate failure analyses that can be particularly useful during conceptual and preliminary design. Finally, the 3D invariant-based criteria are coupled with continuum damage mechanics to model damage onset and propagation, and validated through virtual testing of undisturbed and notched coupons. Application of the continuum damage mechanics model is demonstrated through a local model of the bolted connection of a stringer runout, confirming the accuracy of these analyses in the context of detailed design.*

Keywords: Design, failure, polymer-matrix composites (PMCs), aerospace

1 INTRODUCTION

Materials and structures have been responsible for major improvements in the aerospace sector, mainly in terms of performance and safety. Among the innovative structural concepts and technologies, the development of advanced composite materials, witnessed during the past 40 years, have improved structural performance, reduced operational risk, and enhanced the overall capability of aerospace vehicle airframes. This growing interest in using composite materials for aerospace applications is justified by their beneficial mechanical properties, such as their very high specific stiffness and strength, which enable the design of improved lightweight primary structures. Furthermore, as additional beneficial properties, composites are fatigue and corrosion resistant and their properties can be easily tailored, thus providing additional design flexibility.

In addition to materials and structures, digitalisation is playing an increasingly important role in the efficiency of the aerospace sector. Performance-based certification using virtual qualification and certification techniques is expected to be widely applied, delivering higher efficiency and reliability than physical testing [1]. In this context, research in advanced failure models for composite materials are crucial. Fibre-reinforced composite materials proved to represent a valid alternative to traditional light metal alloys, being able to reduce the weight of aerospace components, while meeting their operational constraints (i.e., reliability, stability, strength, noise, manufacturability and cost). This material replacement gave to the designer the flexibility to select the fibre orientation on a layer by layer basis, building laminated fibre-reinforced composite materials with load-tailored layups. Additionally, enhanced damage tolerance can be built into the structure, by suppressing critical damage modes for the benefit of more favourable ones [2]. On the other hand, this flexibility comes at the cost of a considerably more complex internal material architecture. In fact, due to the inherent heterogeneity and anisotropy of laminated composites, their usage is challenged by the resulting increase of complexity of the design process. The additional parameters to be determined include the exact layering design of the structure (i.e., number, sequence, thickness and mechanical characteristics of each ply) and the manufacturing process to be followed for each component. Furthermore, the formulation of the design criteria required for the structural sizing optimization process (consisting of all relevant strength, fatigue, buckling, aeroelasticity, manufacturing and stiffness constraints) is more challenging due to the employment of these anisotropic materials. As a direct consequence, a detailed numerical modelling using the Finite Element Method (FEM) for an entire aeronautic product would result in millions of degrees of freedom, which is prohibitive for optimization purposes. The structural optimization of composite structures is therefore nowadays performed at a macro-scale level by considering homogenized properties for each laminate, which compromises the level of detail of the resulting optimal design [3–5].

A set of advanced software tools is currently employed within the European aerospace industry to perform structural optimization, incl. design of civil, military as well as experimental aircraft. Structural weight, aerodynamic and aeroelastic performance and stability, structural stability such as buckling, and structural strength are typical design and performance optimization criteria employed by the industry during preliminary multidisciplinary design optimization (MDO). However, to performance structural strength optimization, maximum strain and Tsai-Hill approaches are commonly used, in spite of being considered inadequate for many loading scenarios [6].

This work addresses the development of extended and advanced failure models and design criteria for aerospace composites (Fig.1). The contribution of this work is aimed at enhancing the industrial MDO software, by improving the performance-based optimization processes, providing more accurate and safe failure predictions. In particular, this work is focussed on criteria established (i) at the structural level, delivering detailed and accurate predictions of failure onset in a Finite Element (FE) context, and (ii) at the coupon and structural detail level to predict detailed damage evolution at local level.

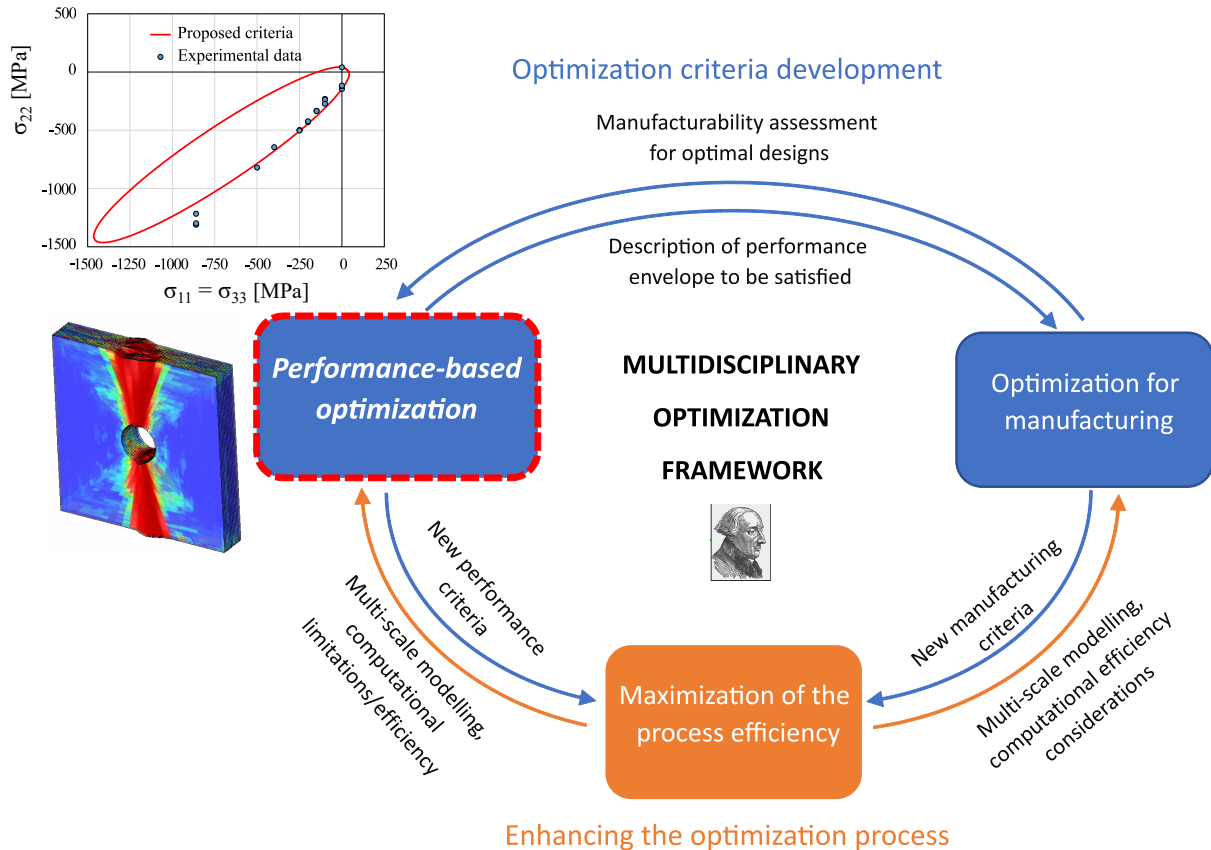


Figure 1: Global view of the European Union’s Horizon 2020 EID OptiMACS — Optimization of Multifunctional Aerospace Composite Structures [7, 8] — where the present research work on failure criteria and damage models contributes to an enhanced “Performance-based optimization”.

2 “HOT-SPOT” PLY AND LAMINATE FAILURE ANALYSIS

After almost sixty years of developments with regards to failure criteria for composite materials, the desirable aspects of a failure theory have been identified and they can be summarized as: (i) the ability to generate failure envelopes close to the experimentally observed failure stress states for different material systems and loading conditions, (ii) consistency in physics and mathematics, meaning that the criteria should, for instance, not allow open envelopes (i.e., infinite strength), and (iii) a stress invariant-based formulation to avoid the inclusion of additional parameters (e.g., the inclination parameters from Puck theory). For the reasons above, the 3D invariant-based failure criteria from Refs. [9, 10] were selected to perform “hot-spot” failure analysis and to develop an extended laminate

failure theory suitable for optimization purposes.

2.1 “Hot-spot” failure analysis

The implementation of failure criteria in a FE software for “hot-spot” failure analysis has been recently exploited to accurately predict critical locations in large global models [11–13]. With the accurate identification of “hot-spots”, it is possible to increase the modelling resolution at the critical locations only, enabling a more efficient and reliable global-local analysis, especially when addressing large-scale composite structures.

In this work, intra-laminar failure is predicted using 3D invariant-based failure criteria [9, 10], formulated based on invariant structural tensors. The onset of delamination is predicted with Ye’s failure theory [14]. Both criteria were implemented in a post-processing Python script, compatible with Abaqus/Standard and Abaqus/Explicit, to generate new element field outputs, by recovering the full stress tensor at each integration point and computing the failure index for each of the failure modes addressed by the criteria. Thus, this approach allows the identification of “hot-spots” and the corresponding failure mechanisms for damage initiation, creating 4 element output variables to predict fibre and matrix failure, under tension and compression, and an additional output variable for the onset of delamination. An alternative implementation is also available through a UVARM subroutine, programmed in FORTRAN, for Abaqus/Standard.

Figures 2 to 5 show an example of “hot-spot” identification at the coupon level, considering open-hole (OH) plates subjected to a quasi-pure bending test case [15]. The predicted failure mechanisms for damage onset are in agreement with the experimental observations from Ref. [15]. Fibre kinking was observed to govern the failure scenario of all open-hole laminates. Then, due to load redistribution, in the experiments failure was suddenly observed on the face under tension due to tensile fibre failure and transverse cracking.

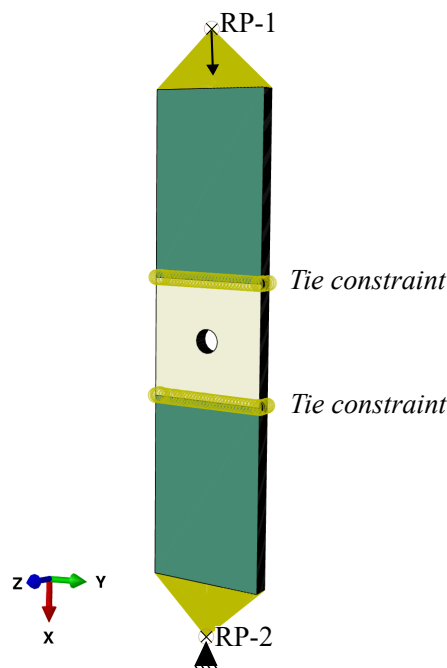


Figure 2: Computational solid model of an OH plate under quasi-pure bending.

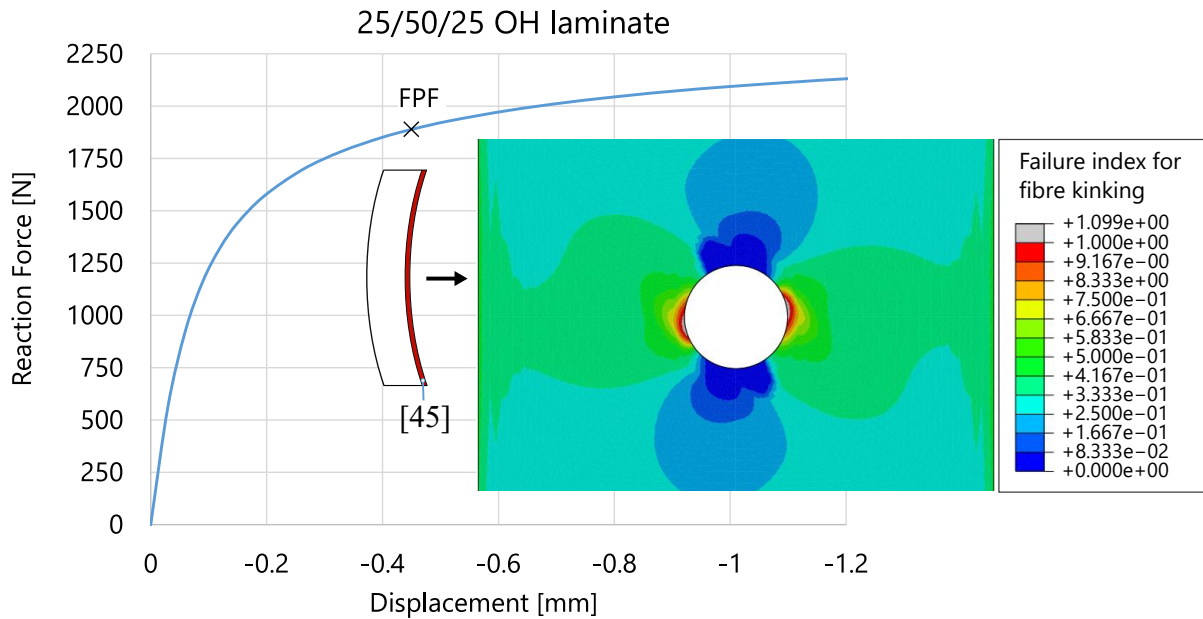


Figure 3: Reaction force-applied displacement plot of a 25/50/25 OH laminate under quasi-pure bending. A detailed view is provided next to the plots for the identification of FPF and the corresponding critical failure mode.

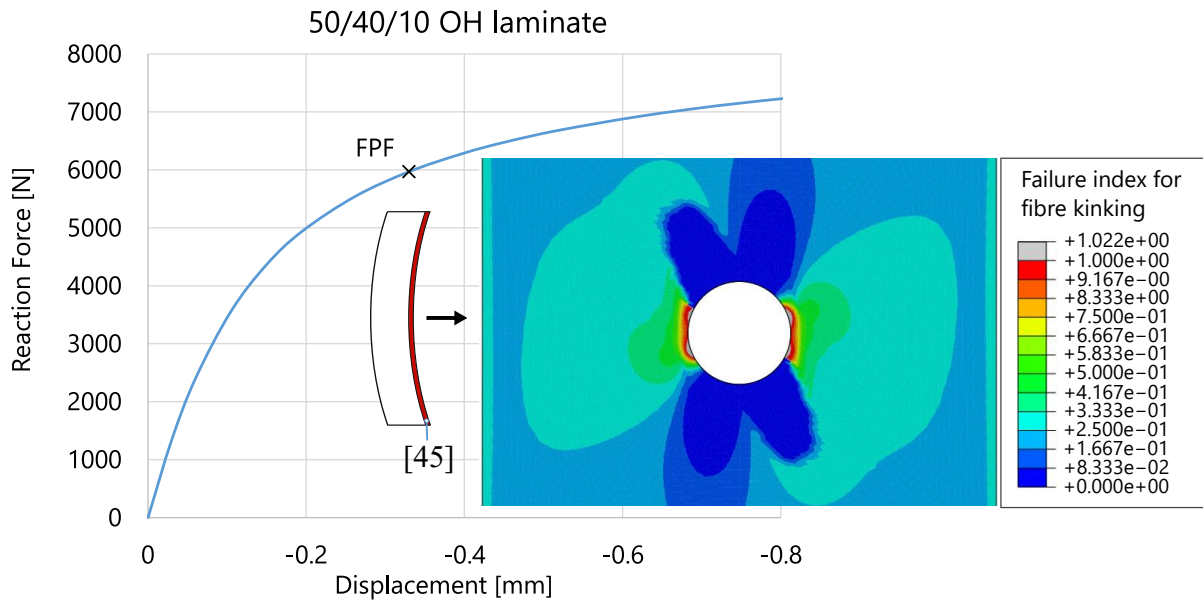


Figure 4: Reaction force-applied displacement plot of a 50/40/10 OH laminate under quasi-pure bending. A detailed view is provided next to the plots for the identification of FPF and the corresponding critical failure mode.

To demonstrate the scalability of this approach, the “hot-spot” failure analysis is also applied to an aeronautical reinforced panel, targeting the identification of the critical locations for damage onset in the runout region. The structure under analysis is composed of a flat composite skin, a composite T-stringer and two bolts. Due to the main assumption of plane stress states, standard shell models do not account for the full stress tensor, which is crucial for the runout region where a change of load path takes place [16]. For

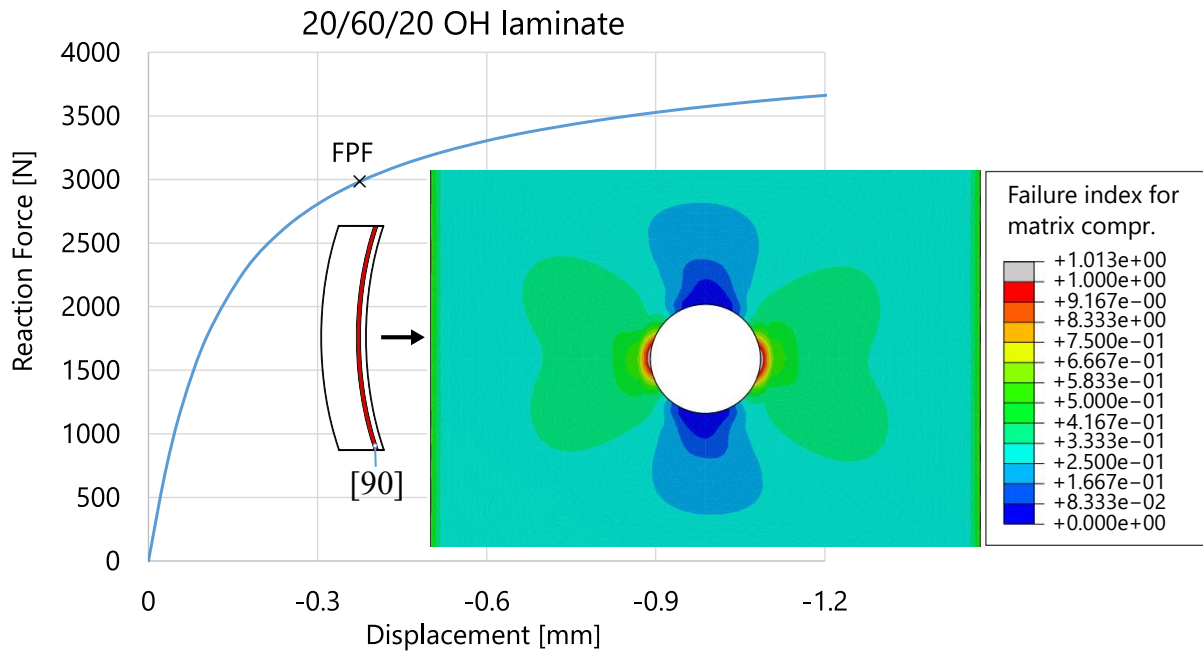


Figure 5: Reaction force-applied displacement plot of a 20/60/20 OH laminate under quasi-pure bending. A detailed view is provided next to the plots for the identification of FPF and the corresponding critical failure mode.

this reason, first-order solid elements C3D8R are used to discretize all the parts of the structure in Abaqus. Furthermore, first-order solid elements allow to properly set the contact behaviour between stringer and bolts, and between skin and bolts.

The analysis of this model is performed in two different steps: (i) a bolt load is applied to the two bolts included in the runout panels, by defining the “Pre-tension load” option in Abaqus and using low-stiffness dummy springs attached to the head of the bolts to prevent rigid body motions in this pre-load step; and (ii) an external tensile loading is introduced by defining “Multi-point constraints” (MPCs) at the two ends of the structure, one to apply the tensile loading and the other to apply the constraints, as schematically shown in Fig.6. These numerical constraints couple the degrees of freedom of the master node with those identified as slave nodes, using the perfectly rigid option for this study.

The discretization of the model is done using one element per ply and a smaller element size in the vicinity of the bolts, where a stress concentration is expected. Then, after running a geometrically non-linear elastic analysis using Abaqus/Standard, a post-processing script was run to perform the “hot-spot” identification with the 3D invariant-based failure theory. Given the large dimension of the model, this study was performed in the regions close to the runout for both skin and stringer, where FPF was expected to occur.

The results of the “hot-spot” failure analysis are shown in Fig.7 and Fig.8. The onset of damage in the skin region close to the runout is triggered by fibre kinking, and a similar failure scenario is observed in the stringer. Experimental results are in agreement with the qualitative predictions obtained with this approach.

2.2 Extended omni-strain envelopes for laminate failure predictions

The 3D invariant-based failure criteria [9, 10] exploited in the previous section deals with ply failure, thus requiring that the FE models are based on a ply by ply discretisation.

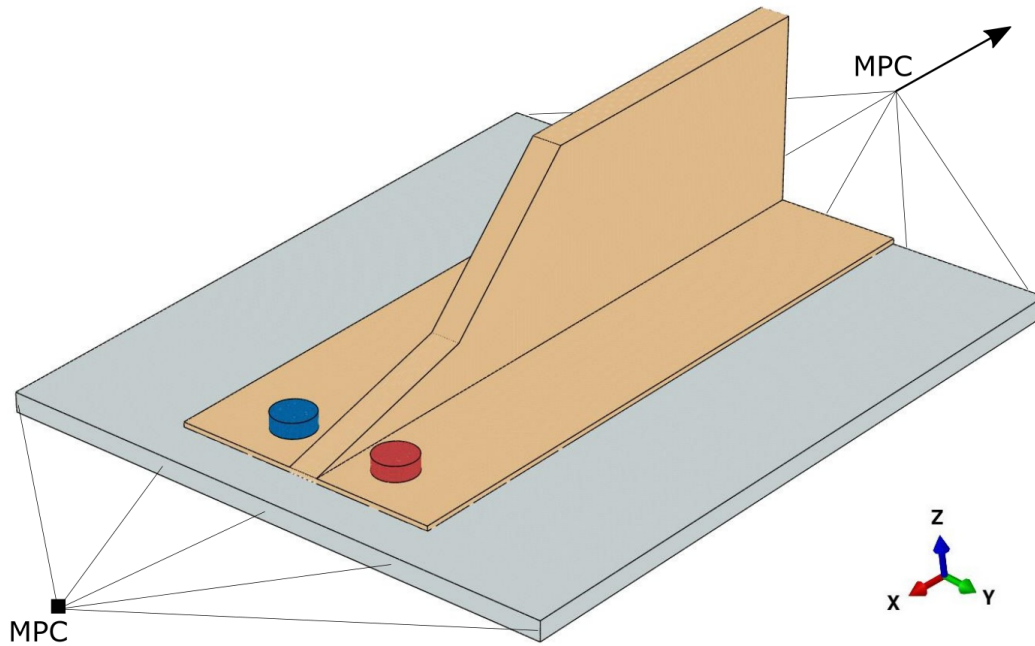


Figure 6: Schematic plot of the model and boundary conditions of the stringer runout.

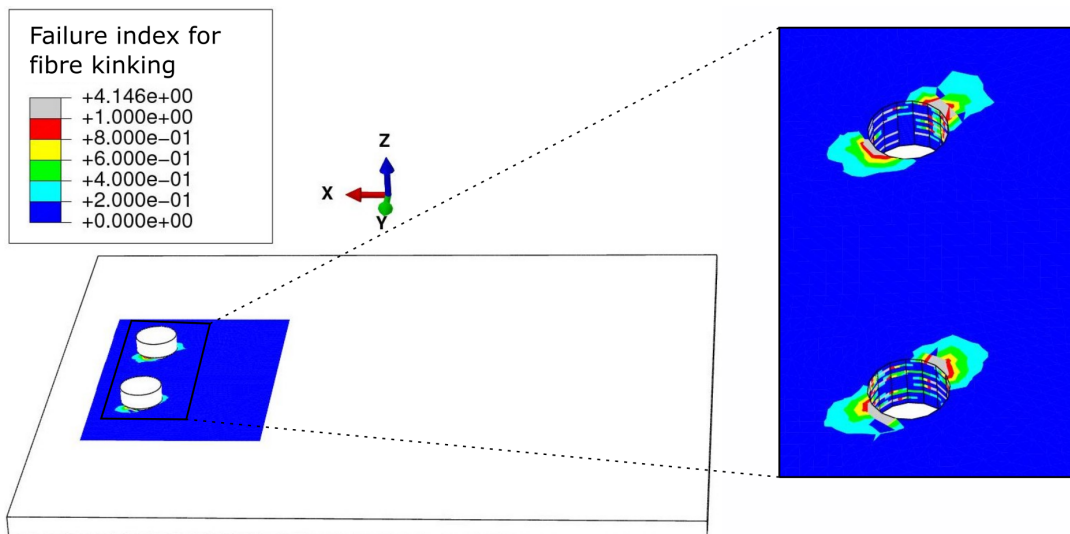


Figure 7: “Hot-spot” failure analysis in the runout region of the stringer. The prediction for failure onset is computed using the 3D invariant-based failure theory, and the critical failure mode is fibre kinking.

In addition, their application in “hot-spot” analysis only allows the identification of FPF. To predict the strength of a composite laminate or structure, LPF analysis may be more attractive, especially during preliminary design optimization.

To meet the industrial need for fast and practical tools at an early design stage, a simple approach for the prediction of LPF was recently proposed by Tsai and Melo [17] based on an extended version of omni strain envelopes. Omni strain failure envelopes, or minimum failure envelopes, can be identified by taking the inner failure envelope in strain space, i.e., by finding the controlling ply that would fail first for unit loading strain vectors

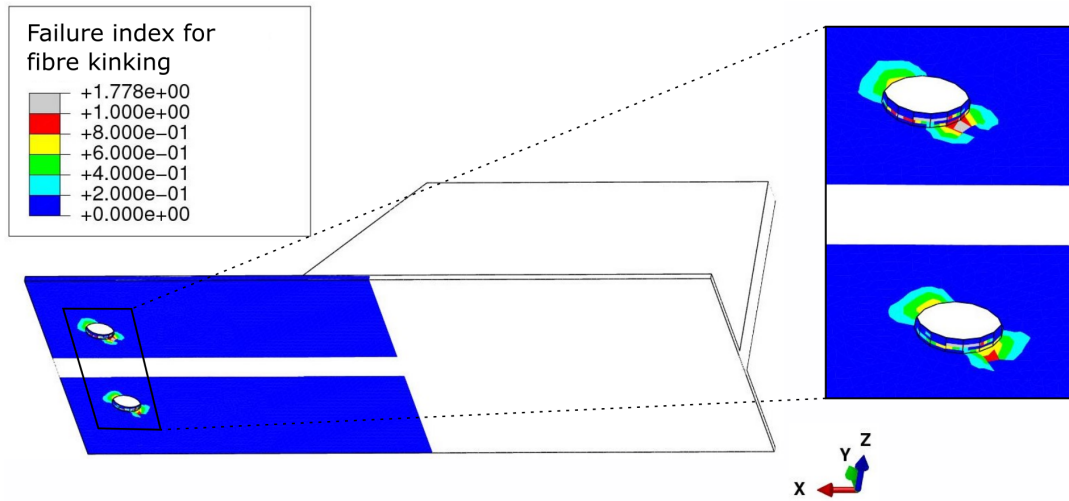


Figure 8: “Hot-spot” failure analysis in the runout region of the skin. The prediction for failure onset is computed using the 3D invariant-based failure theory, and the critical failure mode is fibre kinking.

from 0 to 2π [18]. Omni strain LPF envelopes target the prediction of the continued load-carrying capability of any laminate after damage initiation [17]. Unlike omni strain FPF envelopes, which are obtained using intact ply properties, omni strain LPF envelopes are defined using degraded ply properties, based on a matrix degradation factor (E_m^*) and micro-mechanics relationships [17].

Exploiting the fully 3D description of failure provided by the invariant-based criteria [9, 10], omni strain failure envelopes can be extended by finding the controlling plies in the 3D principal strain space [19]. Figures 9 and 10 show the comparison between experimental strength data from the first World-Wide Failure Exercise (WWFE-I) [20] and the proposed omni LPF envelopes based on the 3D invariant criteria for a carbon/epoxy laminate and an E-glass/epoxy laminate, respectively. The predictions of omni LPF envelopes based on the Tsai-Wu criterion and of a unit circle criterion [17] are also presented. As can be observed, the envelope obtained with the Tsai-Wu criterion and the unit circle envelope are slightly more conservative than the envelope obtained with the invariant-based criteria. The correlation between test data and predictions is excellent, except for the third quadrant where the predictions seem to overestimate the laminate strength under biaxial compression. However, those experimental results cannot be considered 100% reliable, as pointed out also by the organizers of the WWFE-I [20]. Those results were, in fact, characterized by a large scatter in the mean axial compressive strength, probably affected by buckling when testing longer specimens. Thus, the predictive capability of the envelopes cannot be reliably assessed in the compression-compression quadrant.

Figure 11 shows a comparison between 3D omni LPF envelopes and experimental strength data of an E-glass/epoxy laminate under triaxial stress states from the WWFE-II [21]. The laminate LPF envelope, obtained by taking the minimum envelope of the laminate ply orientations in strain space, demonstrates the conservatism of omni LPF envelopes, which are based on the minimum envelope for any possible ply orientation. However, in spite of providing conservative predictions, the omni LPF envelopes define, in a physically-based setting, a safe approach for laminate failure prediction that is independent of the particular layup sequence, thus making its application straightforward for

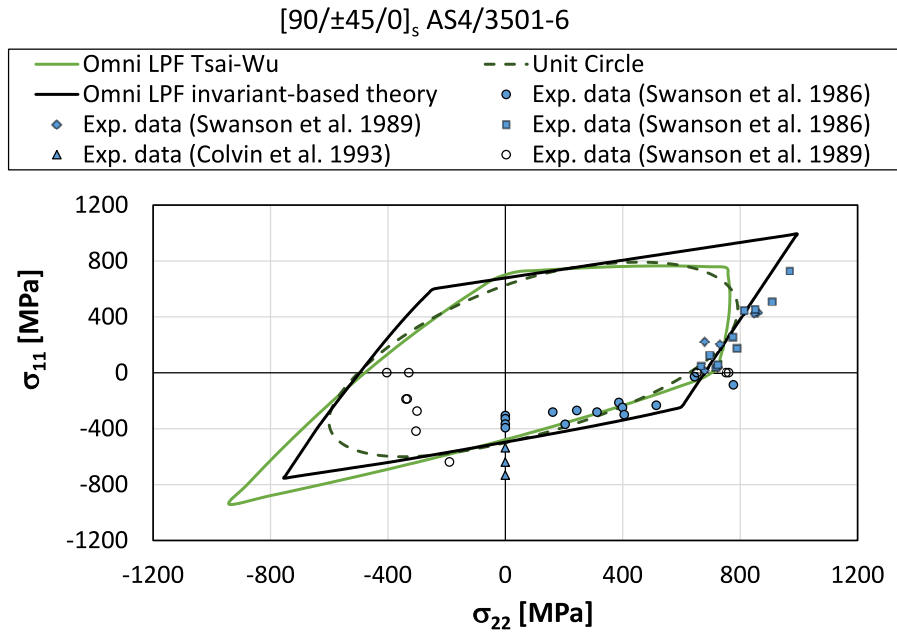


Figure 9: Comparison between failure envelopes and test results from the WWFE-I for a carbon/epoxy laminate [19].

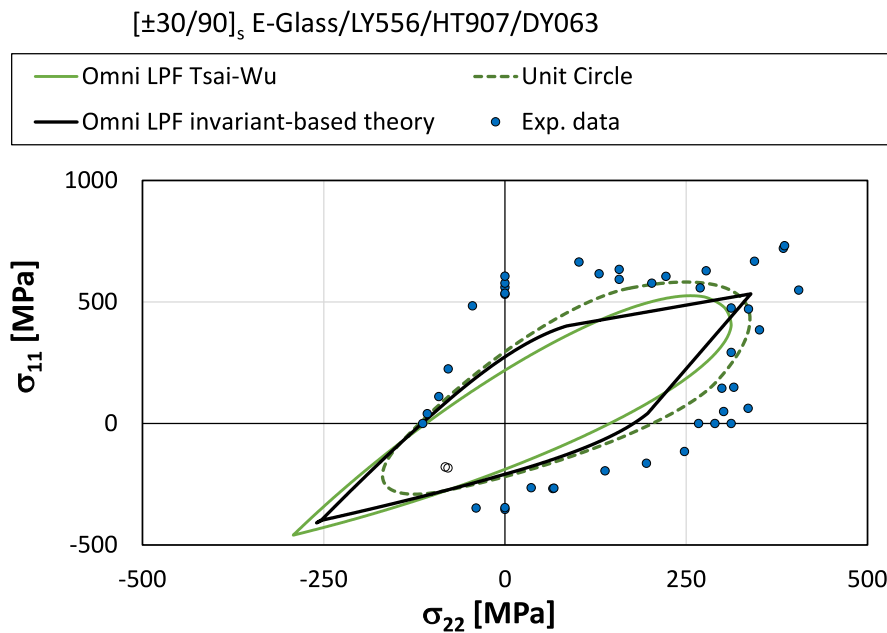


Figure 10: Comparison between failure envelopes and test results from the WWFE-I for an E-glass/epoxy laminate [19].

any laminate of a given material system, which can be a key advantage during preliminary design.

3 ADVANCED FAILURE MODELS FOR COMPOSITE MATERIALS

Although advanced failure criteria can be implemented to perform “hot-spot” FPF analysis, identifying the most critical areas for failure onset, it is not sufficient to pre-

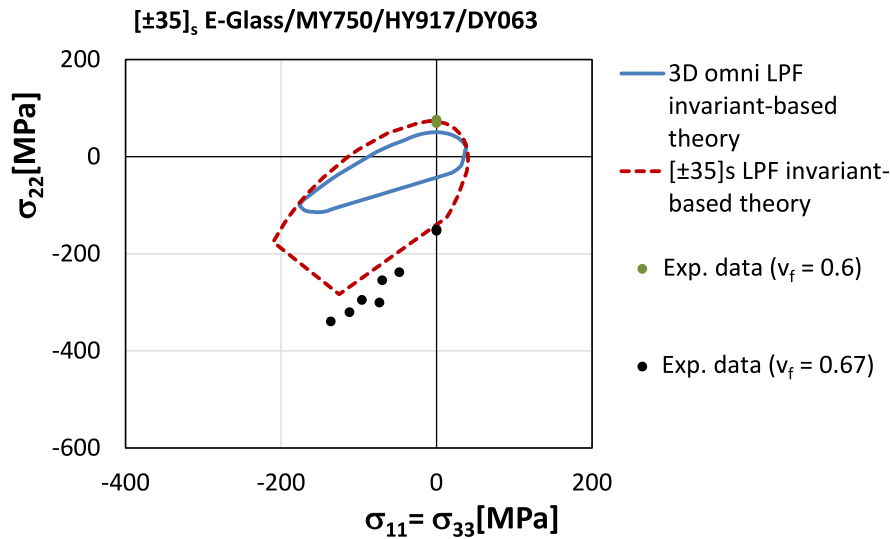


Figure 11: Comparison between failure envelopes and triaxial test results from the WWFE-II for an E-glass/epoxy laminate [19].

dict the ultimate load of composite laminates or structures, since failure of composites is governed by a progressive evolution of damage. Furthermore, although laminate failure theories can be used for “hot-spot” LPF analysis and predict the ultimate load of composite structures, the obtained predictions are not suitable for detailed analysis and accurate design of composite structures. To address inelastic deformation and failure of composite materials and structures, damage models are required.

To predict failure of composite laminates, a frictional smeared crack model [22], formulated in the frame of continuum damage mechanics, is extended to account for the effect of general 3D stress states in the initiation and broadening of fibre kinking. This is achieved through a consistent implementation of the damage activation functions based on the 3D invariant-based criteria [9, 10]. This intra-laminar damage model is then combined with cohesive elements to also predict delamination onset and growth. The proposed modelling strategy, implemented as VUMAT user-subroutines programmed in FORTRAN for Abaqus/Explicit, is firstly verified based on single-element tests and through the prediction of the strength of UD coupons subjected to off-axis tensile and compressive loads (Fig.12 and Fig.13). Then, the proposed modelling strategy is validated against experimental data of the strength of undisturbed and notched (OH) multidirectional laminates, followed by the local analysis of the critical location of the stringer runout from Sect. 2.1 — see Fig.14. It is observed that delamination governs the failure scenario. To study the effect of the bolt pre-load, different values of pre-loading were analysed. The predictions are summarized in Fig.15. As can be observed, the applied bolt pre-load has a minor influence on the mechanical response of the bolted connection, which is in line with the experimental findings on the stringer runout.

4 CONCLUSIONS

A set of 3D phenomenological failure criteria [9, 10] was implemented in a FE code to perform “hot-spot” ply failure analysis and identify the critical locations for damage onset and the corresponding ply failure modes. The “hot-spot” identification analysis resulted

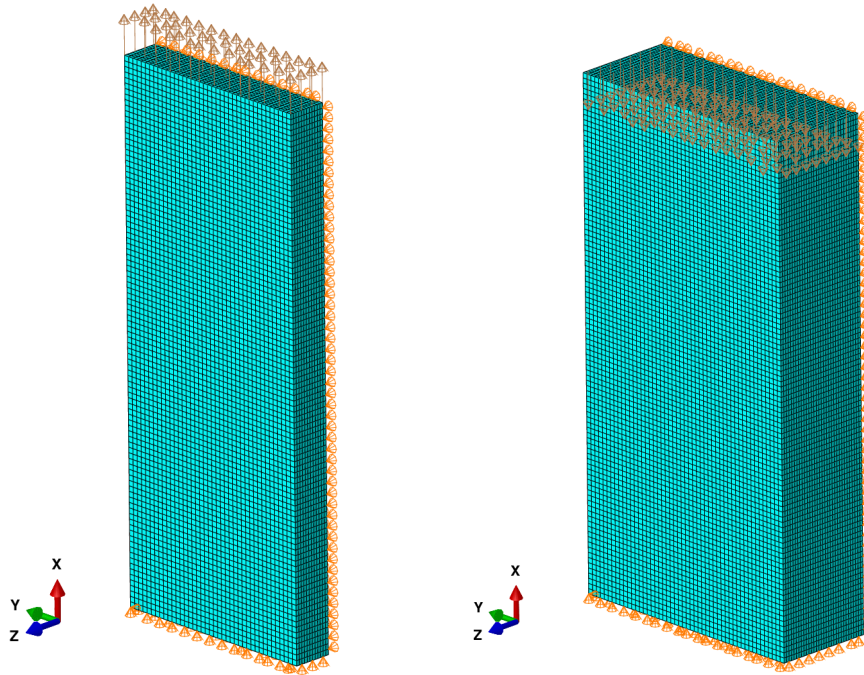


Figure 12: Mesh and boundary conditions for the simulation of off-axis specimens under tension and compression.

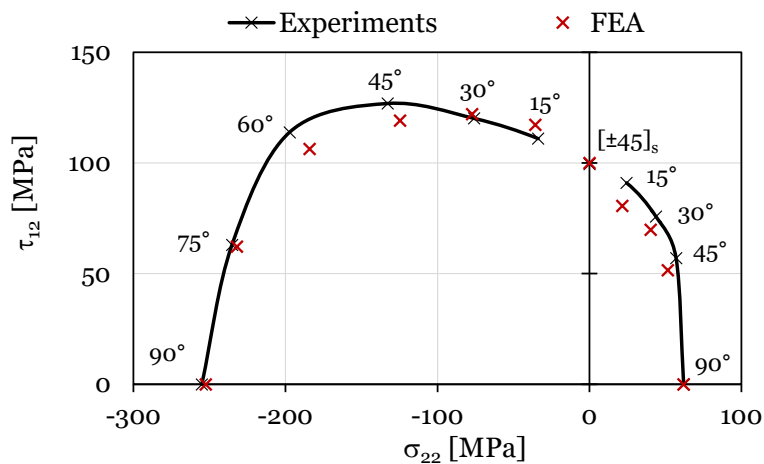


Figure 13: Overall correlation between experimental results and model predictions of off-axis coupons subjected to axial loading.

satisfactorily accurate for the prediction of critical locations for intra-laminate failure onset at the coupon and structural level, confirming the critical failure modes and the locations prone to damage onset observed in experiments.

To address failure at the laminate level, an extension of omni strain failure enveloped to triaxial stress states is proposed. Using as baseline criteria the 3D invariant-based failure criteria [9, 10], omni strain envelopes that predict laminate failure under general 3D stress states were generated. This concept allows simple generalization of ply-based criteria to laminate-based criteria, overcoming two important constraints in MDO of composite aerostructures: (i) computing time, by establishing laminate failure criteria independently

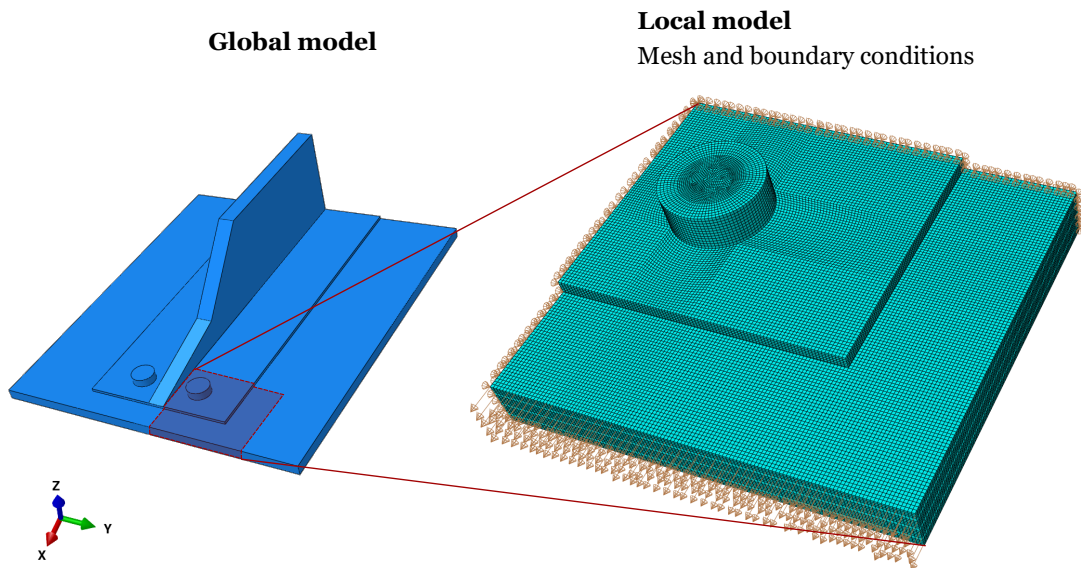


Figure 14: Uncoupled global and local models of a stringer runout subjected to tensile loading.

of ply discretisation, and (ii) excessive conservatism, by incorporating safe LPF criteria instead of over-conservative FPF criteria.

Finally, advanced meso-scale failure models for composite materials are implemented to predict the inelastic deformation up to ultimate failure in detailed models. The 3D invariant-based criteria [9, 10] were coupled with a meso-scale damage model that accounts for the kinematics of matrix cracking and kink band formation during damage propagation [22]. This model accounts for the effect of complex 3D stress states in the prediction of both onset and broadening of longitudinal and transverse failure mechanisms. The bolted connection of a stringer runout analysed for “hot-spot” identification was simulated using the proposed modelling strategy combining the smeared crack model and cohesive elements, showing that delamination governs the failure scenario. Furthermore, the effect of the applied bolt pre-load was studied, leading to a minor effect on the mechanical response of the bolted connection, as observed experimentally.

ACKNOWLEDGEMENTS

Albertino Arteiro would like to acknowledge the Fundação para a Ciência e a Tecnologia (FCT) for its financial support via the UIDP/50022/2020 project (LAETA Programatic Funding). José Reinoso has received funding from the Clean Sky 2 Joint Undertaking under the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 785463. This work was also supported by the OptiMACS project, which is funded by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764650.

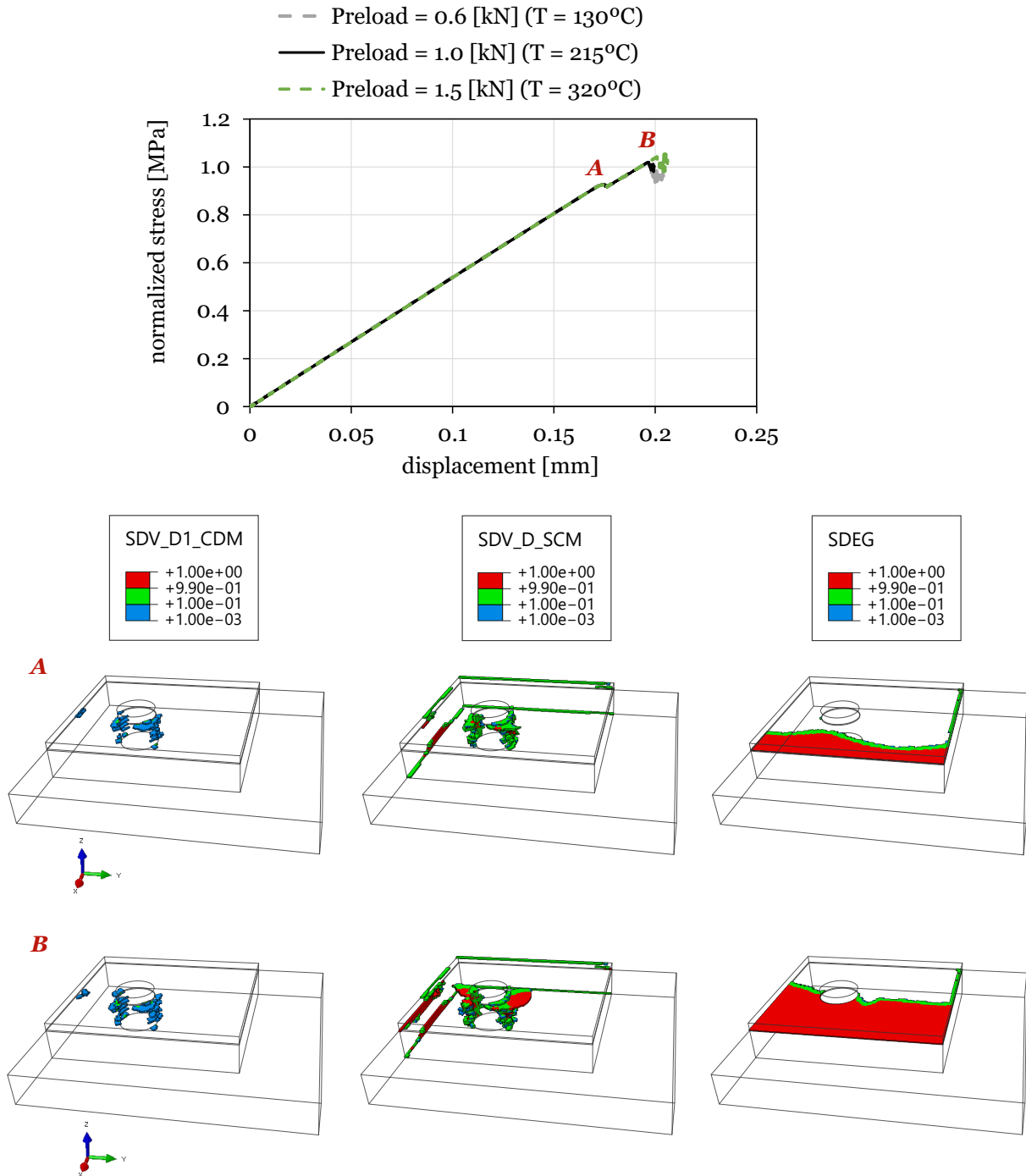


Figure 15: Stress-displacement curves, obtained applying different levels of pre-loading, for the bolted connection under tension. The pictures of damage evolution show, from left to right, the intra-laminar fibre, matrix and inter-laminar damage for the case of 1 kN pre-loading at points A and B in the stress-displacement plot.

REFERENCES

- [1] Directorate-General for Research European Commission and Unit C.3 — Low Emission Future Industries Innovation, Directorate C — Clean Planet. Fly the green deal, Europe’s vision for sustainable aviation. Report of the advisory council for aviation research and innovation in europe (acare), Advisory Council for Aviation Research

- and Innovation in Europe (ACARE), 2022.
- [2] C. S. Lopes, Z. Gürdal, and P. P. Camanho. Tailoring for strength of composite steered-fibre panels with cutouts. *Composites Part A: Applied Science and Manufacturing*, 41(12):1760–1767, 2010.
- [3] G. Schuhmacher, F. Daoud, O. Petersson, and M. Wagner. Multidisciplinary airframe design optimisation. In *28th International Congress of the Aeronautical Sciences*, page 1–13, 2012.
- [4] F. Daoud, O. Petersson, S. Deinert, and P. Bronny. Multidisciplinary airframe design process: Incorporation of steady and unsteady aeroelastic loads. In *12th AIAA Aviation Technology, Integration and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 1–15, September 2012.
- [5] A. Schumacher and T. Vietor. *Advances in Structural and Multidisciplinary Optimization*, 2018.
- [6] R. Jones. *Mechanics of Composite Materials*, second edition edition, 1999.
- [7] G. Corrado, G. Ntourmas, M. Sferza, N. Traiforos, A. Arteiro, L. B. amd D. Chronopoulos, F. Daoud, F. Glock, J. Ninic, E. Ozcan, J. Reinoso, G. Schuhmacher, and T. Turner. Recent progress, challenges and outlook for multidisciplinary structural optimization of aircraft and aerial vehicles. *Progress in Aerospace Sciences*, 135:100861, 2022.
- [8] G. Corrado, G. Ntourmas, M. Sferza, W. Tan, N. Traiforos, A. Arteiro, L. B. amd D. Chronopoulos, F. Daoud, F. Glock, J. Ninic, E. Ozcan, J. Reinoso, G. Schuhmacher, and T. Turner. Corrigendum to “recent progress, challenges and outlook for multidisciplinary structural optimization of aircraft and aerial vehicles”. *Progress in Aerospace Sciences*, 139:100903, 2023.
- [9] P. P. Camanho, A. Arteiro, A. R. Melro, G. Catalanotti, and M. Vogler. Three-dimensional invariant-based failure criteria for fibre-reinforced composites. *International Journal of Solids and Structures*, 55:92–107, 2015.
- [10] P. P. Camanho, A. Arteiro, G. Catalanotti, A. R. Melro, and M. Vogler. *Numerical Modelling of Failure in Advanced Composite Materials*, chapter 5. Three-dimensional invariant-based failure criteria for transversely isotropic fibre-reinforced composites, page 111–150. Woodhead Publishing, Cambridge, 2015.
- [11] H. Molker, R. Gutkin, and L. E. Asp. Implementation of failure criteria for transverse failure of orthotropic non-crimp fabric composite materials. *Composites Part A: Applied Science and Manufacturing*, 92:158–166, 2017.
- [12] H. Molker, R. Gutkin, S. Pinho, and L. E. Asp. Hot spot analysis in complex composite material structures. *Composite Structures*, 207:776–786, 2019.

- [13] X. Zou, S. Yan, M. Reza, L. Brown, and A. Jones. An abaqus plugin for efficient damage initiation hotspot identification in large-scale composite structures with repeated features. *Advances in Engineering Software*, 153:102964, 2021.
- [14] L. Ye. Role of matrix resin in delamination onset and growth in composite laminates. *Composites Science and Technology*, 33(4):257–277, 1988.
- [15] F. Laurin, C. Julien, and P. Paulmier. Damage and strength analysis of open-hole laminated plates under tensile, compressive and bending loadings. In *ECCM 2016 – Proceeding of the 17th European Conference on Composite Materials*, page 26–30, June 2016.
- [16] J. Reinoso, A. Blázquez, L. Távara, F. París, and C. Arellano. Damage tolerance of composite runout panels under tensile loading. *Composites Part B: Engineering*, 96:79–93, 2016.
- [17] S. W. Tsai and J. D. D. Melo. A unit circle failure criterion for carbon fiber reinforced polymer composites. *Composites Science and Technology*, 123:71–78, 2016.
- [18] S. W. Tsai and J. D. D. Melo. An invariant-based theory of composites. *Composites Science and Technology*, 100:237–243, 2014.
- [19] G. Corrado, A. Arteiro, A. T. Marques, J. Reinoso, F. Daoud, and F. Glock. An extended invariant approach to laminate failure of fibre-reinforced polymer structures. *The Aeronautical Journal*, 126:1045–1068, 2021.
- [20] M. J. Hinton, A. S. Kaddour, and P. D. Soden. *Failure Criteria in Fibre Reinforced Polymer Composites: The World-Wide Failure Exercise*, 2004.
- [21] M. J. Hinton and A. S. Kaddour. Triaxial test results for fibre-reinforced composites: The second world-wide failure exercise benchmark data. *Journal of Composite Materials*, 47(6-7):653–678, 2013.
- [22] F. Zhuang, A. Arteiro, C. Furtado, P. Chen, and P. Camanho. Mesoscale modelling of damage in single- and double-shear composite bolted joints. *Composite Structures*, 226:111210, 2019.



DEVELOPMENT OF COMPUTATIONAL AEROELASTIC ANALYSIS TOOLS AND UNCERTAINTY QUANTIFICATION TECHNIQUES FOR RELIABLE FLUTTER PREDICTION

Amin Fereidooni^{1*}, George Lu², Masayuki Yano² and Anant Grewal¹

1: Sustainable Aviation, Aeroacoustics and Structural Dynamics group
Aerospace Research Centre
National Research Council Canada
1200 Montreal Road, Building, Ottawa, Ontario, K1A 0R6, Canada
Amin.Fereidooni@nrc-cnrc.gc.ca, Anant.Grewal@nrc-cnrc.gc.ca
<https://nrc.canada.ca/>

2: University of Toronto Institute for Aerospace Studies (UTIAS)
Faculty of Applied Science and Engineering
University Of Toronto
4925 Dufferin St, North York Ontario, M3H 5T6, Canada
g.lu@mail.utoronto.ca, Masa.Yano@utoronto.ca
<https://www.utias.utoronto.ca/>

Abstract *In this presentation, the development of an OpenFOAM based CFD-CSD solver and its application for Uncertainty Quantification (UQ) of flutter predictions is described. In particular, a coupled compressible solver based on the construction of convective and viscous flux Jacobians is developed, and the Lower-Upper Symmetric Gauss-Seidel algorithm is implemented to solve the resultant matrix. This aerodynamic solver is then coupled with a six DoF structural dynamics solver to form an aeroelastic analysis tool. The simulation results obtained from this aeroelastic tool are then employed to construct a surrogate model using the Gaussian Process (GP) approach. Combined with an error-informed and efficient sampling strategy, the developed UQ framework is applied on the benchmark problem of Isogai 2D NACA 64A010 airfoil. It is shown that the developed framework is more efficient and much faster than the conventional approach in detecting the flutter boundary while providing a probabilistic error estimate of the results.*

Keywords: Uncertainty Quantification, Gaussian Process, coupled compressible solver, flux Jacobian, Lower-Upper Symmetric Gauss-Seidel



DESIGN OPTIMIZATION AND HUMAN SUBJECT SHAKER TESTING OF AN ACTIVE HELICOPTER SEAT MOUNT SYSTEM

Yong (Eric) Chen¹, Amin Fereidooni¹, Rene Laliberte¹ and Viresh Wickramasinghe^{1*}

1: National Research Council Canada
Aerospace Research Centre
1200 Montreal Road, Ottawa, K1A 0R6, Canada
viresh.wickramasinghe@nrc-cnrc.gc.ca

Abstract *Helicopter aircrew are exposed to high levels of Whole-Body Vibration (WBV) in flight operations, which may degrade their ride comfort and performance in the short-term, and contribute to adverse health issues in the long-term. The patented active seat mount technology presented in this paper was developed to reduce the whole-body vibration of the helicopter aircrew through active cancellation of the N/rev vibration peaks related to the helicopter main rotor speed. The latest development of the active seat mount technology included structural redesign for integration into the helicopter cabin and extensive control law tuning tests conducted on a shaker table with human subjects in order to verify and validate the performance, reliability and safety of system prior to flight tests. The shaker test results demonstrated that the redesigned active seat mount achieved significant reduction of the occupant WBV levels at the bottom seat cushion interface per ISO2631-1 metrics and also reduced the occupant head vibrations simultaneously. The promising results demonstrate that the active seat mount is capable of improving the ride quality of the Bell-412 helicopter in representative flight conditions and mitigating adverse long-term health issues of the helicopter pilots.*

Keywords: helicopter vibration, active seat mount, aircrew whole-body vibration, adaptive vibration control, shaker testing, Bell-412 helicopter

1. INTRODUCTION

Although the helicopter is a versatile aerial platform for military and civilian applications, its ride quality is generally poor mainly due to the harsh vibration and noise environment within the cabin. The helicopter fuselage vibration is excited by the rotor aerodynamics and inertial loads at N/rev frequencies of the main rotor speed, where N is the number of main rotor blades [1]. During flight, helicopter floor vibration is transmitted from the seat frame through the seat cushions to the seated aircrew, exposing aircrew to an environment known as Whole-Body Vibration (WBV). Exposure to high levels of WBV adversely affects ride comfort, leading to reduced operational performance. Long term exposure to high WBV levels contributes to health issues such as lower back pain and neck strain in the helicopter aircrew community, which in turn can result in the aircrew losing their flight status [2, 3].

Most helicopters are designed to operate at relatively low main rotor speeds due to aerodynamic considerations, resulting in a fundamental excitation frequency which is typically close to 5 Hz. Furthermore, the ISO 2631-1 standard indicates that the human body responds significantly to vibration within the low frequency range, dominantly between 0.5 Hz and 80 Hz [4]. In particular, the frequency range between 0.5 and 12 Hz is the most sensitive regime in the human spine direction that introduces significant vibrational effect on the human body. Therefore, the ISO 2631-1 standard requires the WBV of a seated human to be measured at the seat cushion interface. Moreover, this standard also provides detailed technical guidance on the methodologies for WBV measurement and assessment. Note that since the dominant N/rev peaks of the helicopter vibration are typically below 25 Hz, it is very important to control the amplitudes of these peaks in order to effectively reduce the WBV exposure of the helicopter aircrew during flight operations.

To provide effective WBV mitigation to the helicopter aircrew, especially for the critical low N/rev frequencies, active control designs for adaptive vibration cancellation have been explored by the National Research Council (NRC) Canada during the past two decades. The active mitigation of helicopter aircrew WBV was explored by Chen et al. [5, 6] using two parallel stacked piezo-stack actuators on a Bell-412 helicopter pilot seat. Under representative Bell-412 vibration profiles, the active seat was able to provide effective vibration reduction at major N/rev peaks except at the $1/\text{rev}$ frequency of 5.4Hz. This was mainly due to the fact that the piezo-stack actuators cannot provide the stroke required for reducing the $1/\text{rev}$ vibration. Further development of the technology using miniature electro-magnetic actuators achieved simultaneous WBV reduction at major N/rev peaks including the $1/\text{rev}$ on the shaker table [7-9]. These studies demonstrated that an active control solution is effective in mitigating WBV for helicopter aircrew. This active seat mount concept for helicopter aircrew vibration control titled “compact vibration reducing human support” has been patented by NRC in a number of jurisdictions such the US (patent no: 10774898B2), UK (patent no: 2559035), Germany (patent no: 102017128067) and Canada (patent application no: 2950508).

Inspired by these technical advances and patenting successes, NRC further improved the active seat mount technology to meet stringent airworthiness requirements for integration into the NRC Bell-412 helicopter. Extensive shaker testing with human subjects using a Bell-412 non-armoured pilot seat on a human-rated shaker table were performed. This paper presents the details of the design of the flightworthy active seat mount system integrated into the NRC Bell-412 helicopter. The results from the human-rated shaker tests are also discussed. Comprehensive shaker test results have convincingly demonstrated that the patented active seat mount technology is able to significantly reduce the WBV levels experienced by helicopter pilots seated on the Bell-412 non-armoured pilot seat in representative flight conditions to enhance the ride quality of the Bell-412 helicopter. Furthermore, this technology also has the

potential to reduce adverse long-term health issues hitherto experienced by the aircrew.

2. PROTOTYPE DESIGN OF THE ACTIVE SEAT MOUNT SYSTEM

During initial studies, the proof-of-concept multi-axis active seat mount system was designed to retrofit the Bell-412 helicopter non-armored pilot seat. Since the Bell-412 pilot seat has two short horizontal supports that slide in to the floor rails of the helicopter cockpit, the prototype seat mount design introduced two identical active seat mount assemblies. The active seat mount assemblies were designed for installation between the Bell-412 pilot seat frame and the helicopter floor rails. This active seat mount system integration arrangement was expected to minimize the impact on the crashworthiness certification of the Bell-412 pilot seat, and to facilitate integration with other existing helicopter seat designs [10, 11]. One of the early prototype assemblies is shown in Figure 1a, while the installation of a Bell-412 non-armoured pilot seat on the prototype active seat system is shown in Figure 1b.

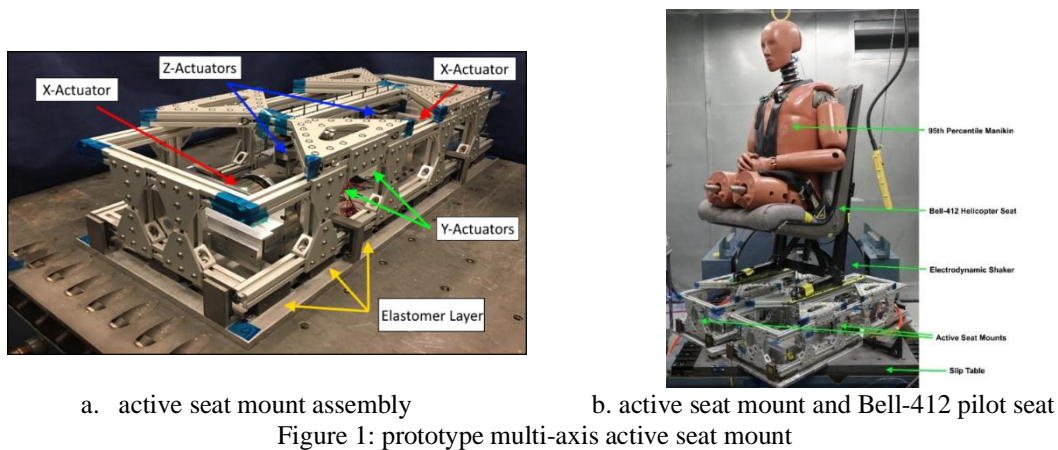


Figure 1: prototype multi-axis active seat mount

The prototype active seat mount assembly was designed with two parallel load-carrying paths. The upper frame of the active seat mount assembly was attached to the seat mount base frame through two structural elements, and the base frame was fastened directly to the seat rails on the helicopter floor:(1) the upper frame structure was attached to the seat mount base frame through a soft elastomer support which provided the baseline damping and stiffness for passive support of the pilot seat;(2) two miniature actuators were connected between the upper frame and the seat mount base to provide vertical vibration control capability in parallel. Similar arrangements were designed in the X and Y directions. Therefore, the active seat mount system has the potential to suppress the whole-body vibration exposure in helicopter aircrew in all 3 orthogonal directions to enhance their ride experience and improve their health protection in all helicopter flight conditions [12].

3. DESIGN IMPROVEMENTS OF ACTIVE SEAT MOUNT

In order to integrate the active seat mount system in an operational helicopter cabin, it was necessary to further improve the design and performance of the system. To achieve this objective, the two active seat mount assemblies were designed specifically to fit the Bell-412 helicopter cabin, while meeting stringent airworthiness requirements and maintaining full functionality. It is important to note that the modifications focused mainly on the installation of the active seat mount assemblies to fit within the NRC Bell-412 helicopter rear cabin for flight demonstration purposes, and it is reasonable to expect that additional structural modifications and further optimizations are required before the technology is ready for commercial product

development.

Since the Bell-412 helicopter pilot seat was already designed and certified to meet crashworthiness requirements, the active seat mount design had to be critically reviewed to maintain this status. Through consultation with airworthiness engineers, it was determined that active seat mount structure had to be evaluated as part of the airframe of the NRC Bell-412 helicopter and therefore all designs and changes were required to meet applicable FAA strength criteria for helicopter airframe structures. However, the installation of the Bell-412 non-armored pilot seat on top of the active seat mount system neither alters the airworthiness nor the crashworthiness status of the seat structure since there is no change to its design or usage.

The first structural requirement was to reduce the weight and dimension of the prototype active seat mount in order to fit into the limited space of the Bell-412 helicopter cabin. The candidate installation locations were the cockpit or the rear cabin. Specifically, the height of the prototype active seat mount needed to be reduced in order to accommodate suitable head clearance of seated human occupants or anthropomorphic test devices (ATD). The prototype design utilized two actuators for vertical vibration control, which were positioned in the middle of the active seat mount assembly resulting in a significant increase of the height of the active seat mount frame. Moreover, based on experimental results of the prototype design, the rated actuator force capacity in the vertical direction was required to allow for sufficient control authority of the high vibration levels experienced by the helicopter at high-speed flight and ground run conditions. In the redesign of the active seat mount, two larger and more powerful voice coil type actuators [13] were relocated at the two ends of the active seat mount assembly to allow for the centre top frame of the mount assembly to be lowered significantly in height. These structural modifications reduced the height of the active seat mount assembly by 127 mm (5") at the seat rail sections, which provided the required vertical space for the Bell-412 non-armored pilot seat frame and occupant wearing a flight helmet to be fitted into the NRC Bell-412 rear cabin with adequate head clearance.

The second structural requirement was centred on the fact that the prototype active seat mount assembly was made of segmented aluminum bars for proof-of-concept demonstration purposes in order to reduce fabrication cost and lead time. Unfortunately, this design was not able to meet airworthiness requirements for aircraft flight testing. Therefore, the aluminum bar frames were replaced with 2024-T3 aluminum sheet metal structural components. The high strength sheet metal structural design was shown to achieve greater strength and stiffness while reducing weight by 16 lbs.

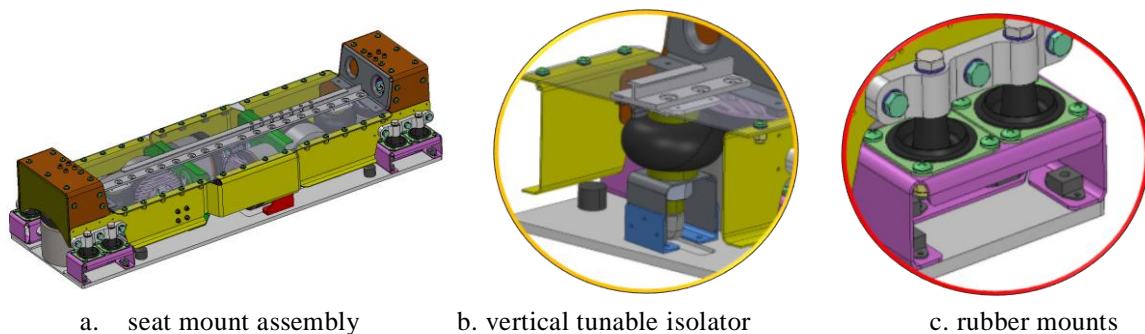


Figure 2: redesigned active seat mount assembly

To avoid excessive displacement of the Bell-412 pilot seat and occupant during aircraft flight maneuvers, the stiffness of the prototype active seat mount was redesigned in all three orthogonal directions. Two tunable stiffness vibration isolators were included to replace the

original soft elastomeric supporting elements in the vertical direction. These tunable isolators not only provided the required passive stiffness and damping for support in the vertical direction, but also offset the compressive displacement due to the static weight of the Bell-412 pilot seat and occupant. Additionally, the soft elastomeric supporting elements in the X and Y directions were also replaced with multiplane rubber mounts. These off-the-shelf rubber mounts were expected to provide predictable and reliable vibration isolation to limit the displacement in all three directions. The active seat mount assembly and the new isolators are schematically shown in Figure 2.

These structural modifications reduced the system's height and weight significantly, with the weight of each assembly reduced from 124 lbs to 51 lbs. The improved active seat mount became sufficiently compact and was able to fit into the NRC Bell-412 helicopter rear cabin for flight testing purposes. The active seat mount system installed with the Bell-412 non-armoured pilot seat frame within the NRC Bell-412 rear cabin, and on the shaker table are shown in Figure 3 a and Figure b, respectively.



a. active seat mount assembly



b. active seat mount tested on shaker table

Figure 3: active seat mount and Bell-412 pilot seat installations

The airworthiness authority of the NRC Flight Research Laboratory performed an independent review of the improved active seat mount structure, and confirmed that the crashworthiness compliance status of Bell-412 seat frame was not altered by installing it on top of the new active seat mount system. The design of active seat mount structure was shown to satisfy applicable airworthiness requirements regarding the integrity of Bell-412 airframe through finite element analysis. In addition, the control computer and other electronic equipment were powered and tested using the NRC Bell-412 helicopter project power supply, and shown to satisfy applicable safety and functionality requirements for flight testing. Therefore, the improved active seat mount was approved for installation on the NRC Bell-412 helicopter test platform for experimental flight test and demonstration purposes.

4. ADAPTIVE CONTROL ALGORITHM DEVELOPMENT

Helicopter flight tests show that the aircrew vibration spectra are dominated by N/rev harmonics of rotor dynamic load [12]. In particular, the measured aircrew body vibration was dominated by the $1/\text{rev}$ and $4/\text{rev}$ harmonic peaks while the contribution from the $2/\text{rev}$ and $6/\text{rev}$ harmonics was minor. Therefore, the control force generated by the actuators of the active seat mount was primarily required to counteract the forced vibration loads at N/rev harmonics in order to provide effective vibration suppression for the aircrew body. Secondly, critical resonant modes of the seated aircrew that are being excited due to random vibration and these vibration modes were required to be suppressed as well. With an effectively designed adaptive

control algorithm, the active seat mount system was able to provide vibration suppression of the N/rev harmonics as well as control of the resonant structural modes to significantly reduce the vibration transmitted to helicopter aircrew.

The performance objective of the control algorithm developed for this investigation was to reduce the aircrew vibration measured on the seat cushion interface and monitor the reduction in vibration response at the helmet. In particular, the accelerometer that measured the seat cushion vibration in the vertical direction was selected as the error sensor, $e(k)$, while accelerometers at the helmet in the vertical direction was monitored for overall system performance evaluation. The actuators in each X, Y, Z directions were electrically configured in parallel to operate with a single control input channel. This configuration enabled the implementation of a Single-Input-Single-Output (SISO) control system to control the active helicopter seat mount system in a single direction [14].

4.1 Adaptive Notch Concept to Control Harmonics

There are several control approaches that have been used in the past to suppress harmonic disturbances with varying degrees of success. Among the control strategies to eliminate harmonic disturbances, Morgan and Stanford developed a promising controller design approach that introduced “notches” at synchronous frequencies [15] to control harmonics. For narrow-band applications, a reliable sinusoidal reference signal is used to configure the Least Mean Square (LMS) as an adaptive notch filter by removing the primary spectral components within a narrow band centered about the reference frequency. Chen et al have shown previously that an adaptive notch controller implemented with a Filtered-x Least Mean Square (FLMS) algorithm with a suitable reference signal was able to cancel the helicopter rotor vibration [16]. This previously published principle of a FLMS algorithm formulation shown in Figure 4 was used for the current investigation [17]. It is important to note that discrete time implementation of the controller and plant model was used for the controller development.

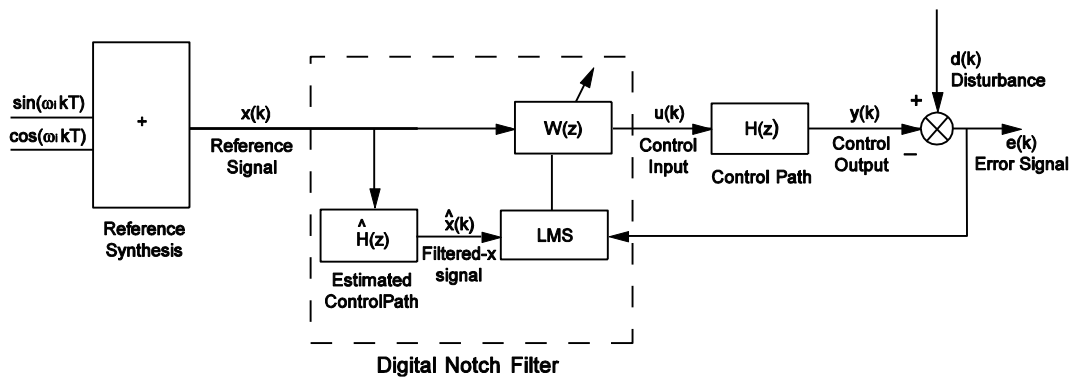


Figure 4: Adaptive notch controller diagram

Here, the disturbance $d(k)$ represents the external excitation forces such as the N/rev harmonics and random excitation of the aircrew. The error signal $e(k)$ is the acceleration at the aircrew helmet location used to satisfy the objective of the controller. The reference signal, control input and control output are denoted by $x(k)$, $u(k)$ and $y(k)$, respectively. Since the disturbance frequency ω_i is usually unknown to designers, an estimate of the frequency, $\hat{\omega}_i$, is necessary. Thereafter, the reference signal is synthesized as a combination of sinusoids at the estimated frequencies as following:

$$x(k) = \sum_{i=1}^o c_i \cos \hat{\omega}_i kT = \sum_{i=1}^o \frac{1}{2} [e^{j\hat{\omega}_i kT} + e^{-j\hat{\omega}_i kT}] \quad (1)$$

where $i = 1 \dots o$ while the o is the number of frequencies to be included in the reference signal. The reference signal is filtered with magnitude and delay through the control path. Assuming the transfer function of the control path, estimated as $\hat{H}(z)$, generates the weights of the FIR filter for the control path transfer function $h(k)$, the filtered signal can be expressed as:

$$\hat{x}(k) = h(k) * x(k) \quad (2)$$

The filtered signal $\hat{x}(k)$ is then adapted by the LMS algorithm as discussed previously in the published references [18]. $W(z)$ provides the controller coefficients of the iterative weights $w(k)$ in order to generate the adapted control input $u(k)$ using the reference signal $x(k)$ as shown below;

$$u(k) = w(k) * x(k) \quad (3)$$

The control output signal $y(k)$ denoted in the discrete domain as $Y(z)$ is generated by the FLMS algorithm and expressed as follows:

$$\begin{aligned} Y(z) = & \frac{n\mu}{4} E(z) \sum_{i=1}^m |\hat{H}(z)|^2 \left[\frac{1}{ze^{-j\hat{\omega}_i T} - 1} + \frac{1}{ze^{j\hat{\omega}_i T} - 1} \right] \\ & + \sum_{i=1}^m \sum_{k=1}^m \frac{\mu |\hat{H}(z)| |\hat{H}(z)|}{4} \frac{\sin\left(\frac{\hat{\omega}_i - \hat{\omega}_k}{2} nT\right)}{\sin\left(\frac{\hat{\omega}_i - \hat{\omega}_k}{2} T\right)} \Big|_{i \neq k} \\ & + \sum_{i=1}^m \sum_{k=1}^m \frac{\mu |\hat{H}(z)| |\hat{H}(z)|}{4} \frac{\sin\left(\frac{\hat{\omega}_i + \hat{\omega}_k}{2} nT\right)}{\sin\left(\frac{\hat{\omega}_i + \hat{\omega}_k}{2} T\right)} \end{aligned} \quad (4)$$

where, $E(z)$ is the error signal in the discrete domain. Assuming that no error in the frequency estimate is present and the control path is Linear Time Invariant (LTI), the first term in the control signal is time-invariant, which produces notches at the estimated frequencies, while terms 2 and 3 are time-varying. However, if the length of the FIR filter, n , is comparatively long and the convergence coefficient, μ , is relatively small, the time-varying components will be small and can be neglected. As a result, an equivalent filter with a notch at frequency $\hat{\omega}_i$ will be obtained. The equivalent closed-loop transfer function $C(z)$ between the harmonic reference signal and the error signal has the form of a classical bandpass filter, with notches at the reference frequencies, and can be expressed as:

$$C(z) = \frac{1}{1 + \frac{n\mu}{4} \sum_{i=1}^m |\hat{H}(z)|^2 \left[\frac{1}{e^{-j\hat{\omega}_i T} - 1} + \frac{1}{e^{j\hat{\omega}_i T} - 1} \right]} \quad (5)$$

If the disturbance frequency varies slowly with time, it is necessary to identify the variation in

frequency in real time to update the reference signal for the FLMS controller, such that a tunable notch filter can be formed adaptively.

However, due to the nature of identification, a frequency estimate error between $\hat{\omega}_i$ and ω_i is inevitable. As a result, the modulation between the error signal $e(k)$ and the filtered reference signal $\hat{x}(k)$ will lead to an oscillation in the weights of the FLMS algorithm. If the estimate error is small so that the $\omega_i - \hat{\omega}_i$ component in the controller weight is stronger than $\omega_i + \hat{\omega}_i$, the notch filter effect would also be present. Therefore, an efficient and accurate frequency estimation method is essential for the algorithm to track the shift in the disturbance frequencies.

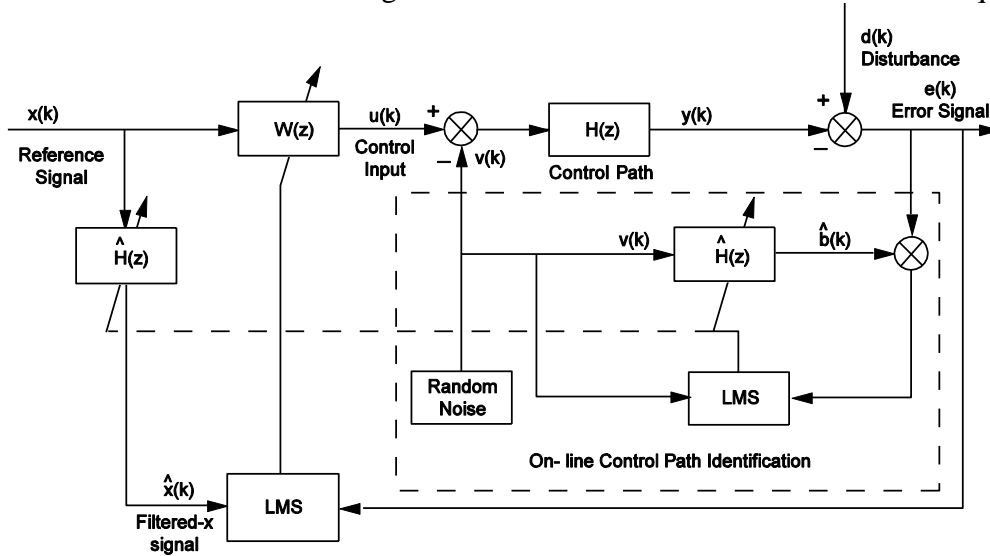


Figure 5: On-line identification of control path

4.2 Control Path Identification Method

As mentioned previously, the aircrew vibration frequency varies with the flight conditions, aircrew weight, posture and many other factors. Therefore, the transfer function of the control path, $H(z)$, is time-varying and therefore needs to be estimated on-line. The block diagram of a SISO adaptive feed-forward control law in Figure 5 shows the identification process for the control path, represented by the transfer function between the stacked piezoelectric actuators and the accelerometer at the helmet location. By introducing white noise, $v(k)$, into the control input as the identification excitation, the contribution of this excitation $b(k)$ is measured by the error sensor such that

$$b(k) = h(k) * v(k) \quad (6)$$

If the white noise and control signal are not correlated, an estimate of the control path $\hat{H}(z)$ can be obtained using the LMS algorithm. Assuming $\hat{H}(z)$ is represented by a FIR filter, the weights $w_h(k)$ can be obtained through iteration as follows:

$$W_h(k+1) = W_h(k) + \mu_h v(k) [(b(k) + d(k) - h(k) * u(k) - \hat{b}(k))] \quad (7)$$

By defining

$$\hat{b}(k) = \hat{h}(k) * v(k) \quad (8)$$

$$o(k) = d(k) - h(k) * u(k) = d(k) - h(k) * w(k) * x(k) \quad (9)$$

the adaptive law governing the weights of the control path is expressed as follows:

$$W_h(k+1) = W_h(k) + \mu_h v(k)[(b(k) + o(k) - \hat{b}(k))] \quad (10)$$

Compared to the off-line system identification method, it can be shown that $o(k)$ represents a signal from the feed-forward controller. This parameter interferes with the control path identification process, which may slow down the convergence speed and even lead to a divergence of the process [19].

A compensation applied to $o(k)$ as shown in Figure 6 is able to improve the efficiency in the control path identification process. Because $o(k)$ is a narrow band signal that consists of multiple harmonic frequencies, it can be cancelled by introducing another adaptive filter, $G(z)$, which has multiple notches at the corresponding frequencies in order to eliminate the interference in control path identification.

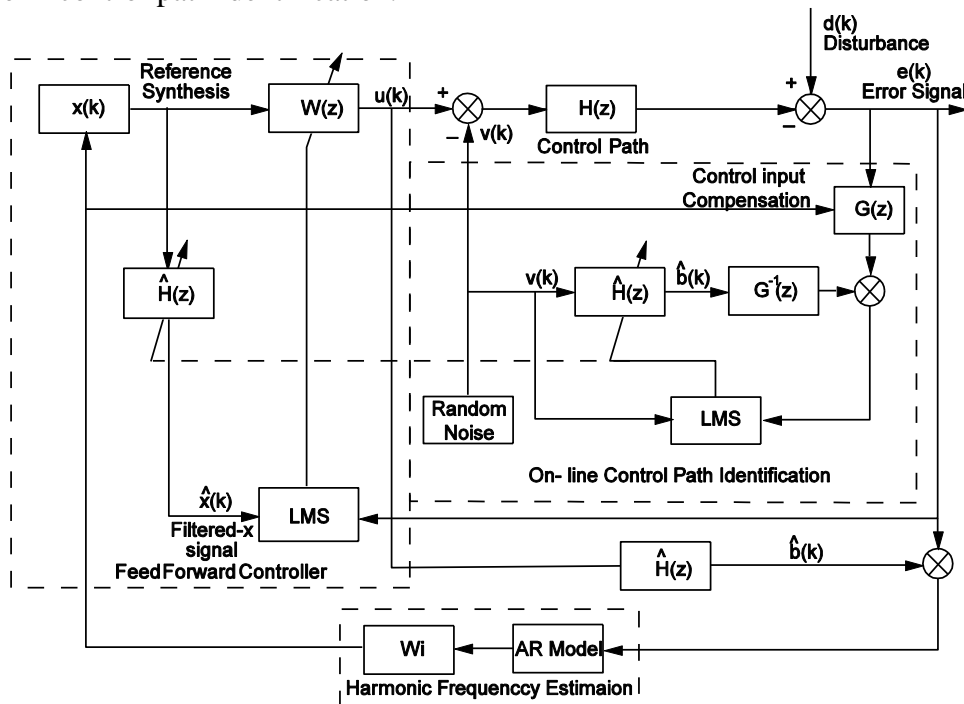


Figure 6: Compensator for control path identification

The filter $G(z)$ can be designed for frequency $\hat{\omega}_i$ as seen from Figure 7, only two weights are needed to cancel one frequency, which can be expressed as

$$\begin{aligned} w_{G,1}(k+1) &= w_G(k) + c\mu_1 ee(k) \cos(k\hat{\omega}_i T + \Phi_i), \\ w_{G,2}(k+1) &= w_{G,2}(k) - c\mu_1 ee(k) \sin(k\hat{\omega}_i T + \Phi_i) \end{aligned} \quad (11)$$

where $ee(k)$ is error signal of the adaptive notch filter, μ_1 is the convergence coefficient of the compensator and c is the amplitude of the synthesized reference signal. The equivalent transfer function of the compensator filter is as follows:

$$G(z) = \frac{z^2 - 2z \cos \omega_0 T + 1}{z^2 - 2(1 - \mu_1 c^2)z \cos \omega_0 T + 1 - 2\mu_1 c^2} \quad (12)$$

As long as a reliable reference signal that includes controllable sinusoidal interference is available, the adaptive notch filter automatically creates a notch over each sinusoid and tracks its frequency drift. Therefore, this adaptive notch filter provides a simple and effective alternative to other methods of tracking and eliminating N/rev harmonics in the helicopter seat vibration control application.

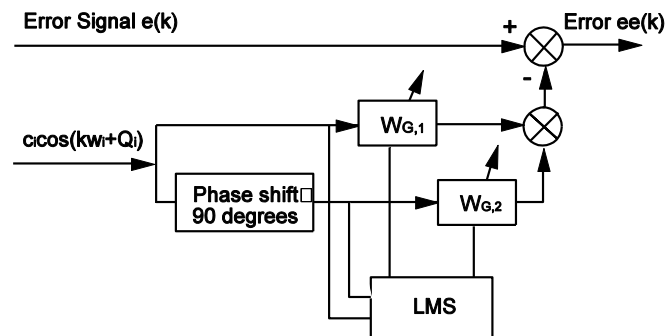


Figure 7: Adaptive notch filter

4.3 Reference Signal Synthesis Method

A reliable reference signal is very important for the successful application of the FLMS algorithm for vibration control because it guarantees that the controller is stable in the closed-loop system. Among a number of possible approaches, the signal synthesis method based on digital signal processing techniques was selected because it avoided introduction of feedback loops in the control path. This reference signal synthesis approach guarantees that the controller is purely feed-forward and stable, and also assumes that the variation of disturbance frequency is slow, which is appropriate for helicopter vibration control applications.

An accurate and efficient estimate of the disturbance frequency is necessary to achieve reliable cancellation of harmonic vibrations transmitted to the aircrew through the seat. There are two approaches, namely, non-parametric and parametric, that were used to estimate the frequency content of a signal based on a finite set of data observations. The non-parametric methods are based on the Fast Fourier Transform (FFT) technique. Since the accuracy of this method depends on the length of the data series, comparatively long data series are required to obtain an accurate estimate. Therefore, this type of FFT based frequency estimation method is not suitable for real-time applications. The high computational burden of FFT analysis is another challenge for real-time controller implementation. An alternative approach is to use a parametric method that is based on digital signal processing techniques. Although it is relatively fast, this approach provides high resolution frequency estimates despite short data series, making it an attractive approach for the present adaptive control algorithm. The parametric method uses a predetermined model to estimate the frequency components of the signal. A covariance autoregressive (AR) method is adopted in the present investigation. The AR model can be expressed as:

$$P(z^{-1}) = \frac{K}{A(z^{-1})} = \frac{K}{1 + a_2 z^{-2} + a_3 z^{-3} + \dots + a_{p+1} z^{-q+1}} \quad (13)$$

where, q , K and a_i are model order, scalar gain and coefficients, respectively. Due to the fact that the poles of the AR model correspond to the estimated frequencies, the frequencies can be calculated from the AR polynomial once an estimate of the AR model coefficients is obtained. Therefore, a reference signal for the adaptive algorithm could be synthesized as a linear combination of multiple sinusoids as shown in Eq. (1).

An effective adaptive controller to counteract helicopter rotor harmonics was successfully developed with such a synthesized reference signal. Since the harmonic frequencies in aircrew excitation are estimated on-line and selectively included in the reference signal, the algorithm does not only suppress vibration modes selectively, but also tracks the change in the vibration spectra. In this investigation, the floor vibration that excites the seat frame was measured directly by an accelerometer and used as the reference signal in the adaptive feed-forward control law. This approach guaranteed that the FLMS control system was stable in the closed-loop system.

4. AIRCREW WBV MITIGATION ON THE SHAKER TABLE

Prior to flight testing on the NRC Bell-412 helicopter, validation testing of the improved active seat mount system was performed on a human-rated shaker table using human occupants. Bell-412 floor vibration profiles measured during actual flights were used as the shaker vibration profiles in the laboratory, with the approval of NRC Research Ethics Board since testing was being performed using human subjects.

The human-rated shaker table was designed to provide simulated vibration excitations in the vertical direction only. The improved active seat mount was fastened to a baseplate matching the Bell-412 floor interface pattern, and then installed on the shaker table. A Bell-412 non-armored pilot seat was mounted on the upper frames of the active seat mount assembly through two Bell-412 floor rails. A 178 cm tall male volunteer weighing 180 lbs was seated on the Bell-412 pilot seat as the occupant, and secured using a four-point pilot seat harness. This human subject closely represented a 50th percentile male per US anthropometric data. The shaker test configuration is shown in Figure 3b.

The shaker table provided a statistically stationary vibration input to excite the active seat mount assemblies vertically. The simulated vibration profile, which was derived from Bell-412 floor vibration measurements, included eight N/rev harmonic peaks each scaled to match with the Bell-412 floor vibration spectrum at 120 kts. In addition, a low-level random vibration between 3 and 50 Hz was superimposed to simulate broadband aerodynamic excitations.

Closed-loop tests have been performed to reduce the whole-body vibrations per metrics suggested in ISO 2631-1 standard. First, the WBV responses of the human subject on the Bell-412 seat were recorded and used as the baseline parameters. Next, the FLMS adaptive controller was activated to control the active seat mount assembly in the vertical direction to reduce the WBV of the occupant. The vibration responses of the occupant and active seat mount system were monitored and recorded throughout the experiment. When the controller converged, the WBV levels of the occupant were compared with the baseline parameters to evaluate the WBV mitigation performance.

4.1 WBV Vibration at Seat Cushion Interface

The vibrations measured at the interfaces between the seat cushions and the human body were frequency weighted using the W_k factor defined in ISO 2631-1:1997 and then used as the metric to determine the occupant WBV exposure levels. Test result of the 50th percentile male subject measured as a time history of the W_k -weighted WBV at the bottom seat cushion interface is shown in Figure 8. The entire duration of the test was 480 seconds, which followed the below

sequence: 1. during the first 30 seconds, the occupant experienced the vertical WBV level without control; 2. the actuator was then powered by the FLMS controller starting from 30s and adaptation of the controller terminated at 330s; 3. next, the converged adaptive controller stopped the adaptation but maintained control between 330s and 380s; 4. the control voltage was then stopped between 380s and 420s; 5. the control voltage was activated again at 420s and stopped at 470s; 6. finally, the controller was disconnected at 480s.

When the active seat mount was not activated, it was clearly shown that the human occupant was subjected to relatively high WBV levels, and the time history also showed unsymmetric vibration which may be related to the periodical N/rev harmonic excitations from the shaker table. Based on ISO 2631-1, the ISO Wk-weighted rms level of the vertical vibration was 0.786 m/s^2 , and the ride comfort level was qualitatively rated as “Fairly Uncomfortable”. Based on the health guidance in ISO 2631-1, the permitted daily exposure time was 1.6 hrs.

Despite running at a relatively conservative convergence rate, the FLMS controller was able to quickly reduce the WBV vibration and converged within 20 seconds by adaptively suppressing the dominant N/rev peaks, which resulted in significant reduction of the Wk-weighted WBV vibration. By evaluating the response between 330 and 380th seconds, the Wk-weighted rms WBV level decreased to 0.43 m/s^2 , and the ride comfort level was qualitatively rated as “A Little Uncomfortable”. Per ISO standard, the WBV level was reduced by 45.2%, extending the daily permitted exposure time to 18.4 hrs, which effectively eliminated the daily exposure time limit for this simulated flight condition.

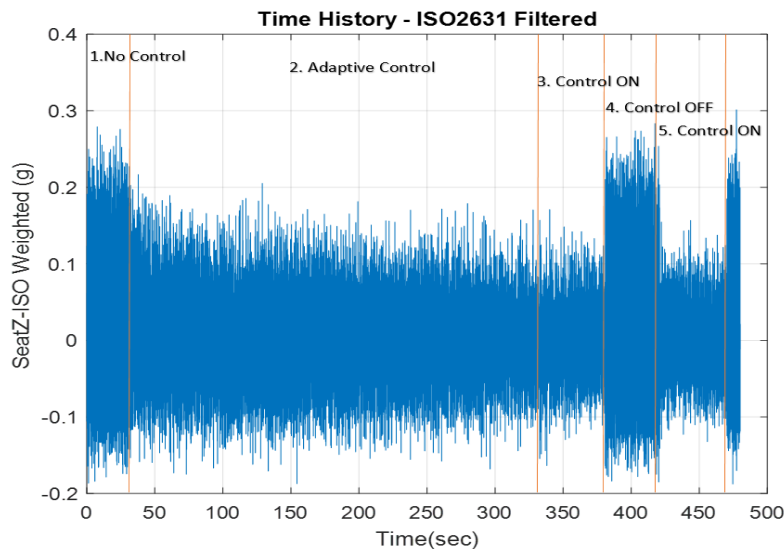


Figure 8: Time trace of the vertical vibration level at the occupant and seat cushion interface

Spectral analysis of the ISO-weighted WBV for the “No Control” and “Controlled” segments is shown in Figure 9. It was clearly shown that the dominant WBV peaks including 1/rev, 2/rev and 4/rev were effectively suppressed by the active seat mount. In particular, the amplitudes of the 1, 2 and 4/rev peaks were reduced by 68.7%, 96.5% and 98.9%, respectively. There was no significant control spillover within the frequency range below 80 Hz, indicating that the FLMS controller was effective and robust in the control of the active seat mount on the human-rated shaker table.

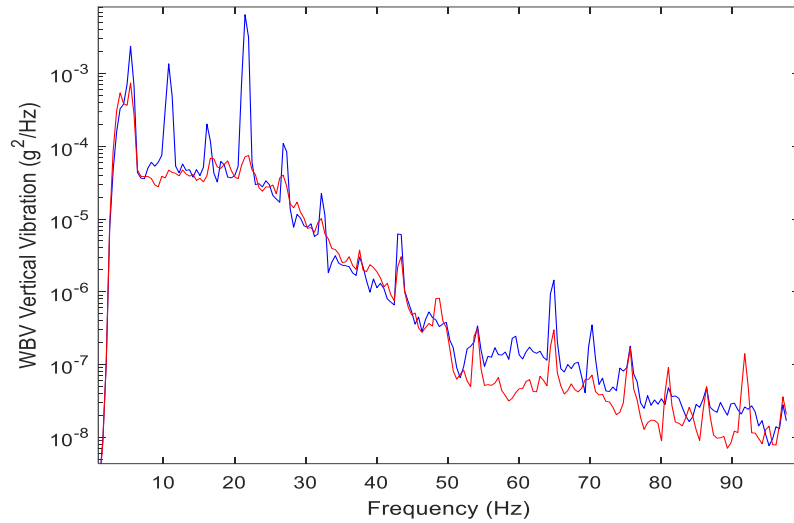
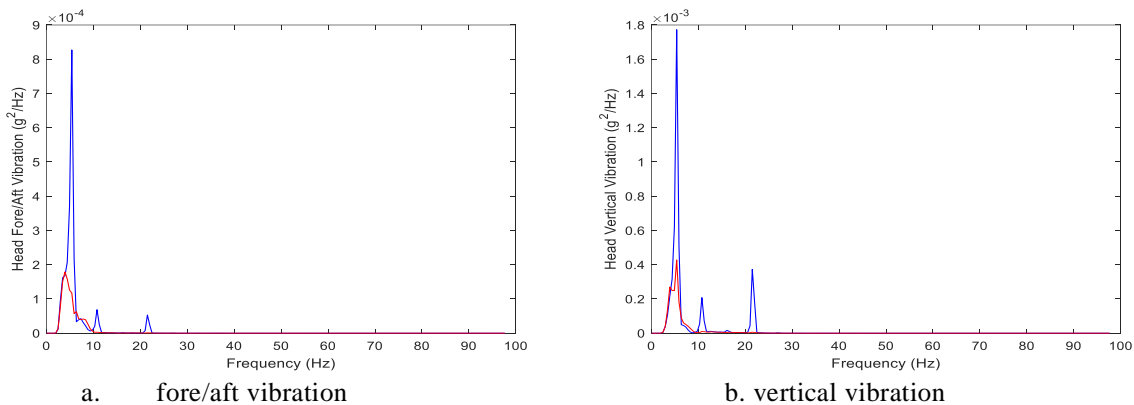


Figure 8: spectral plot of Wk weighted vertical vibration level at the occupant and seat cushion interface

4.2 Aircrew Head Vibration Evaluation

Besides the WBV levels, the vibration levels of the occupant head were also evaluated. The head vibrations were measured by a tri-axial accelerometer bonded to the top of the HGU-56P helicopter flight helmet worn by the occupant. Spectral analysis of the occupant's head vibration is shown in Figure 9.

The plots indicated that human biodynamics was critical in the transmission of vibration from the seat cushion interfaces to the occupant head. Different from the vibration at the bottom seat cushion interface in Figure 4b, which was dominated by multiple N/rev peaks, it was clearly shown that the occupant head vibration was dominated by a single 1/rev peak in both fore/aft and vertical directions. Evident in ISO 2631-1 for the Wk-weighting factor, the human spine biodynamics introduces significant amplification between 5 and 8 Hz. Coupled with the 1/rev frequency of the Bell-412 main rotor speed at 5.4 Hz, the occupant head vibration showed a distinctively high peak at the 1/rev frequency. Compared to the fore/aft direction, the vertical vibration level was higher although both were characterized by a dominant 1/rev peak. The overall rms vibration was 0.33 in the fore/aft direction and 0.47 m/s² in the vertical direction. Since only vertical excitation is provided on the shaker table, the head vibration results indicated that significant human biodynamic coupling existed in the head fore/aft and vertical directions.



a. fore/aft vibration

b. vertical vibration

Figure 9: occupant head vibration on the shaker table

With the adaptive control of the active seat mount in the vertical direction, not only were WBV vibration levels at the bottom seat cushion interface reduced, but the occupant head vibration levels were also reduced simultaneously. As shown in Figure 9b, the amplitude of 1/rev peak was reduced by 85.9% for fore/aft and 75.9% for vertical direction. As a result, the overall rms vibration level was reduced by 27.6% in the fore/aft and 35.9% in the vertical direction, respectively.

The shaker test results clearly demonstrated that the active helicopter seat mount system was able to reduce the occupant WBV at the bottom seat cushion interface to improve the ride quality, and also reduced the head vibrations simultaneously. The enhanced ride comfort and head/helmet stabilization would improve the aircrew performance by reducing their neck muscle fatigue, increasing visual clarity of the instrument panel and enhancing situational awareness in flight operations.

5. CONCLUSIONS

The redesign and flight testing of the NRC-patented active seat mount technology to reduce aircrew whole-body vibration levels on the NRC Bell-412 helicopter is presented. The flight worthy active seat mount system was integrated with a Bell-412 non-armored pilot seat. The WBV mitigation performance was evaluated using human subjects through extensive tests performed on a human-rated shaker table.

Through adaptive suppression of multiple major N/rev vibration peaks at the bottom seat cushion interface, the NRC-patented active seat mount system was successful in reducing the occupant WBV as well as head vibrations simultaneously in accordance with ISO 2631-1 metrics. The active seat mount performed reliably and consistently for all of the human subjects in all vibration levels corresponding to flight conditions. The extensive shaker test results convincingly demonstrated that the active seat mount technology was able to significantly enhance the ride quality of helicopter pilot seats, mitigate adverse WBV effects on the pilot and improve their operational performance in realistic flight conditions.

REFERENCES

- [1] Seddon, J. M., and Newman, S., *Basic Helicopter Aerodynamics*, John Wiley & Sons, West Sussex, UK, 2011.
- [2] Bongers, P., Hulshof, C., Dijkstra, L., Boshuizen, H., Groenhout, H. and Valken, E., "Back pain and exposure to whole body vibration in helicopter pilots," *Ergonomics*, Vol. 33, No. 8, 1990, pp. 1007-1026.
- [3] Branco, C., Rodriguez, E., "The Vibroacoustic Disease - An Emerging Pathology", *Aviation, space and environmental medicine*, Vol. 70, 1999, pp. A1-6.
- [4] International Standards Organization, *Mechanical vibration and shock - Evaluation of human exposure to whole-body vibration - Part 1: General requirements*, Geneva, Switzerland, 1997.
- [5] Harrer, K. L., Yniquez, D., Majar, M., Ellenbecker, D., Estrada, N. and Geiger, M., "Whole Body Vibration Exposure for MH-60S Pilots," *Forty Third Annual SAFE Association Symposium*, Crewell, OR, 2005.
- [6] Gan, Z., Hillis, A. J. and Darling, J., "Adaptive control of an active seat for occupant vibration reduction," *Journal of Sound and Vibration*, Vol. 349, 2015, pp. 39-55.
- [7] Hiemenz, G., Hu, W., and Wereley, N., "Semi-Active Magnetorheological Helicopter Crew Seat Suspension for Vibration Isolation", *Journal of Aircraft*, Vol. 45, 2008, pp. 945-953.
- [8] Chen, Y., Wickramasinghe, V., and Zimcik, D., "Development of Adaptive Seat Mounts

- for Helicopter Aircrew Body Vibration Reduction”, *Journal of Vibration and Control*, Vol. 15, 2009, pp. 683-706.
- [9] Chen, Y., Wickramasinghe, V., and Zimcik, D. G., “Development of Adaptive Helicopter Seat for Aircrew Vibration Reduction,” *Journal of Intelligent Material Systems and Structures*, Vol. 22, No. 5, 2011, pp. 489-502.
- [10] Graham, S., Fereidooni, A., Chen, Y., and Wickramasinghe, V. K., “Investigation of a Parallel Active Vibration Isolation Mount for Mitigating N/Rev Helicopter Vibrations,” 25th AIAA/AHS Adaptive Structures Conference, AIAA 2017-1441, 2017.
- [11] Hadj-Moussa, F., Fereidooni, A., Chen, Y., and Wickramasinghe, V., “Development of a Multi-axis Active Seat Mount to Mitigate Vibration Transmission to Helicopter Aircrew,” 2018 AIAA SciTech Forum, Kissimmee, Florida, January 8-12, 2018.
- [12] Chang, J. Fereidooni, A. Wickramasinghe, V. and Chen, Y., “Development of a multi-axis Active Seat Mount system for helicopter aircrew whole-body vibration mitigation”, 30th International Conferences on Adaptive Structures and Technologies, Montreal, Canada, October 8-10, 2019.
- [13] Moticon website, <http://www.pwr-con.com/gvcm-095-038-01.htm>.
- [14] Hansen, C., and Snyder, S., *Active Control of Noise and Vibration*, E & FN Spon, London, UK, 1997.
- [15] Morgan, D. R. and Sanford, C., “A control theory approach to the stability and transient analysis of the filtered-X LMS adaptive notch filter,” *IEEE Transactions on Signal Processing*, Vol. 40, No. 9, 1992, pp. 2341-2346.
- [16] Chen Y., Zimcik D. G. and Wickramasinghe V.K., 2005, “An Adaptive Impedance Control Algorithm for Helicopter Blade Vibration Suppression,” *Journal of Vibration and Control*, 11(4), pp. 543-560.
- [17] Chen Y., Wickramasinghe, V. and Zimcik, D. G., 2009, “Development of Adaptive Seat Mounts for Helicopter Aircrew Body Vibration Reduction,” *Journal of Vibration and Control*, 15(12), pp. 1809-1825
- [18] Glover J. R., 1997, “Adaptive Noise Canceling Applied to Sinusoidal Interferences,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(6), pp. 484-491
- [19] Eriksson L J, Allie M. A., 1989, “Use of Random Noise for Online Transducer Estimate in an Adaptive Active Attenuation System”, *Journal of the Acoustical Society of America*, 85(2), pp. 797-802.



ADAPTIVE AIRBAG SYSTEMS FOR PROTECTION OF GENERAL AVIATION

**Jan Holnicki-Szulc¹, Rami Faraj¹, Cezary Graczykowski¹, Grzegorz Mikułowski¹,
Piotr Pawłowski¹, Andrzej Świercz¹, Zbigniew Wolejsza¹, Lech Knap²,
Krzysztof Sekuła³, Dariusz Wiacek³**

1: Institute of Fundamental Technological Research
Polish Academy of Sciences
Pawinskiego 5B, 02-106 Warsaw, Poland
{holnicki, rfaraj, cgraczyk, gmikulow, ppawl, aswiercz, zwolejsza}@ippt.pan.pl,
<http://www.ippt.pan.pl>

2: Institute of Vehicles and Construction Machinery
Warsaw University of Technology
Narbutta 84, 02-524 Warsaw, Poland
Lech.Knap@pw.edu.pl, <http://www.simr.pw.edu.pl>

3: Adaptronica Sp. z o. o.
Szpitalna 32, 05-092 Łomianki, Poland
{krzysztof.sekula, dariusz.wiacek}@adaptronica.pl,
<http://www.adaptronica.pl>

Abstract *The contribution describes three innovative external airbag systems developed by the authors for the protection of flying objects during emergency landings. The first one is the AdBag system dedicated for small drones, which is designed to protect the carried equipment and prevent damages to objects or injuries to people at the crash location. The second system is external airbag designed for ultralight aircraft Skyleader 600, which provides significant reduction of touchdown velocity and deceleration levels during emergency landings, thereby improving protection of the pilot and the passengers. Finally, the last presented solution is the Spring-Drop system with specialized airbag deployment technique, which is dedicated for specialised airdrop operations where the touchdown conditions can be extremely harsh and unexpected, while protection of transported cargo is of crucial importance. Both conceptual studies, numerical simulations and experimental tests of the three proposed systems are presented and discussed.*

Keywords: External airbags, adaptive system, emergency landing, human safety

1. INTRODUCTION

Most of the flying objects of general aviation such as drones and small aircrafts are endangered of emergency situations caused by equipment failure or human error, which result in the requirement of emergency landing. Although such flying objects are typically equipped with parachute rescue systems, they usually do not provide sufficient protection of human or carried cargo in the case of emergency situations occurring at relatively low altitudes. The possible solution of this problem is application of external adaptive airbags located below the flying object, which are deployed and inflated when the situation of emergency landing is detected. During touchdown, the activation and adjustment of special vents provides proper release of gas from the airbag, resulting in efficient impact mitigation and improved level of protection of the landing object.

The design of adaptive airbags is associated with multiple theoretical and practical challenges. In general, the concept of adaptive airbags follows a more general concept of Adaptive Impact Absorption, in which the system is adapted in semi-active way to actual dynamic loading with the use of structural fuses based on smart materials [1, 2, 3]. In a specific case of adaptive airbags, the first problem is precise identification of the landing conditions, including mass, velocity components and spatial orientation of the object [4]. The second challenge is the requirement of fast deployment and inflation of the airbag in the short period of time between detection of failure and emergency landing [5]. Finally, the third problem is providing appropriate large release of the gas from the airbag during the short period of landing [6, 7]. Due to the requirement of large gas flow rates and process robustness, neither the typical electromechanical valves nor piezoelectric valves [8, 9, 10], which are successfully used in adaptive shock absorbers, can be applied. Instead, the attention is focused on standard passive vents, dedicated membrane valves [11] or semi-passive solutions [12, 13, 14]. All the above factors cause that design and construction of adaptive external airbags for emergency landing still remains challenging engineering problem.

The following part of the paper presents three different airbag systems the protection of flying objects during emergency landings developed during several years of cooperation between Institute of Fundamental Technological Research of the Polish Academy of Sciences, Adaptronica Company and Warsaw University of Technology. The Section 2 describes AdBag system dedicated for small drones, which protects the carried equipment and prevents damages at the crash location. The Section 3 presents external airbag designed for ultralight aircraft Skyleader 600, which provides significant reduction of deceleration during emergency landings. In turn, the Section 3 highlights recently developed Spring-Drop system based on specialized airbag deployment technique, which seems promising in extremely harsh landing conditions. Eventually, in the last section summarizes the achievements and draws basic conclusions regarding the proposed systems of adaptive external airbags.

2. ADBAG – ADAPTIVE AIRBAG FOR SMALL UAV

Small drones with a total mass of several kilograms are becoming increasingly popular in various applications, including aerial surveillance, airborne observation and laser scanning. The popularity of such applications increases the likelihood of the occurrence of emergency situations. Given that drones often operate over densely populated urban areas, it is necessary to ensure protection of individuals or objects on the ground, and to minimize the transmitted shock loads to the fragile electronic equipment on the drone, such as cameras, lenses, and lidars.

The developed adaptive emergency landing system is designed specifically for unmanned aerial vehicles (UAVs), but it can also be utilized for the protection of crewed aircraft and other

flying objects. The proposed device comprises an airbag with at least one pre-cut vent hole, an inflation module containing a fan drive, a gas release activation module, a fault detection module, and a ground proximity detection module. The device is also equipped with a pyrotechnic charge and an igniter located on the airbag envelope in the vicinity of the pre-cut vent hole. Additionally, the device incorporates mechanical stabilizers for spatial orientation of the object positioned above the airbag, which are designed for compact folding and rapid deployment. The operation of the device involves activating and inflating the airbag upon detection of an emergency situation, pyrotechnic activation of the vent holes, and executing the landing process through controlled gas release. The proposed device ensures a stable landing of the UAV in the appropriate spatial orientation, adapts to impact conditions by activating a selected number of vent holes based on prediction of impact conditions, and effectively mitigates the impact forces during the landing process.

The initial stage of the elaboration of the Ad-Bag system was development of a dedicated airbag design method, which included application of simplified analytical models for approximate adjustment of system parameters as well as application of fully nonlinear dynamic FEM models (LS-Dyna explicit solver) for precise airbag shape design (Fig. 1). Moreover, a drop test stand with the maximal drop height of 20 m, composed of a vertical guiding system, remote release of suspended object, and a base plate with piezoelectric force transducers for measurement of total vertical loads was designed and constructed. The series of drop test conducted with the use target-shaped airbag (Fig. 2) allowed determine system dynamic response the required area of the valve for various mass and velocity of the landing object.

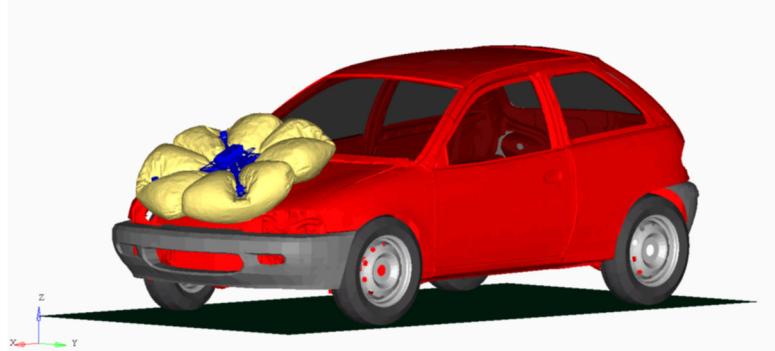


Figure 1. Visualization of the numerical finite element simulation showing emergency landing of the drone equipped with external airbag on the standard city car (numerical simulation by Mr. Krzysztof Hinc, [15]).



Figure 2. Complete AdBag system with a target-shaped airbag, prepared for drop tests

In the following stage of research the AdBag system has been integrated with the "Iron bird" for the free-fall testing purposes. The free-fall test were conducted in accordance to general guidelines of the project and has proved that the AdBag System demonstrator operates successfully during free-fall in field conditions. Selected frames captured by a high-speed camera during free-fall and landing are presented in Fig. 3a. Moreover, a quadcopter drone was equipped with the complete integrated AdBag system and prepared to be used in flight tests (Fig. 3b).

Ultimately, the flight tests of the developed system were conducted. During the test, the drone has achieved the height of 14.3 m. The descent speed at the moment of impact with the ground was equal 8.9 m/s. The vent holes were successfully ignited at the correct altitude. The drone impacted the ground at an angle of approximately 40 degrees, which was attributed to partial malfunction of one aerodynamic stabilizer. Nevertheless, the test demonstrated effectiveness of the AdBag system even in the case of a rotated drone during descent. After the impact with the ground with the AdBag system, the drone did not lose its flight capabilities.

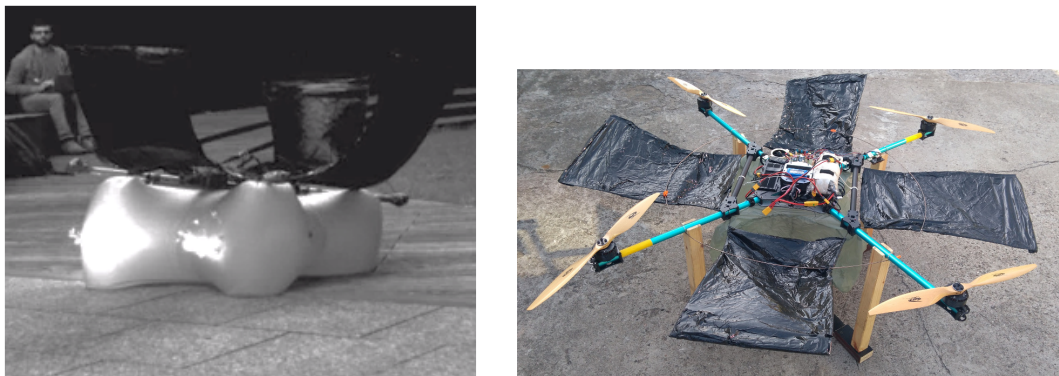


Figure 3. a) Free-fall tests of the complete AdBag system, b) drone equipped with the AdBag system and stabilizers constructed for flight testing purposes.

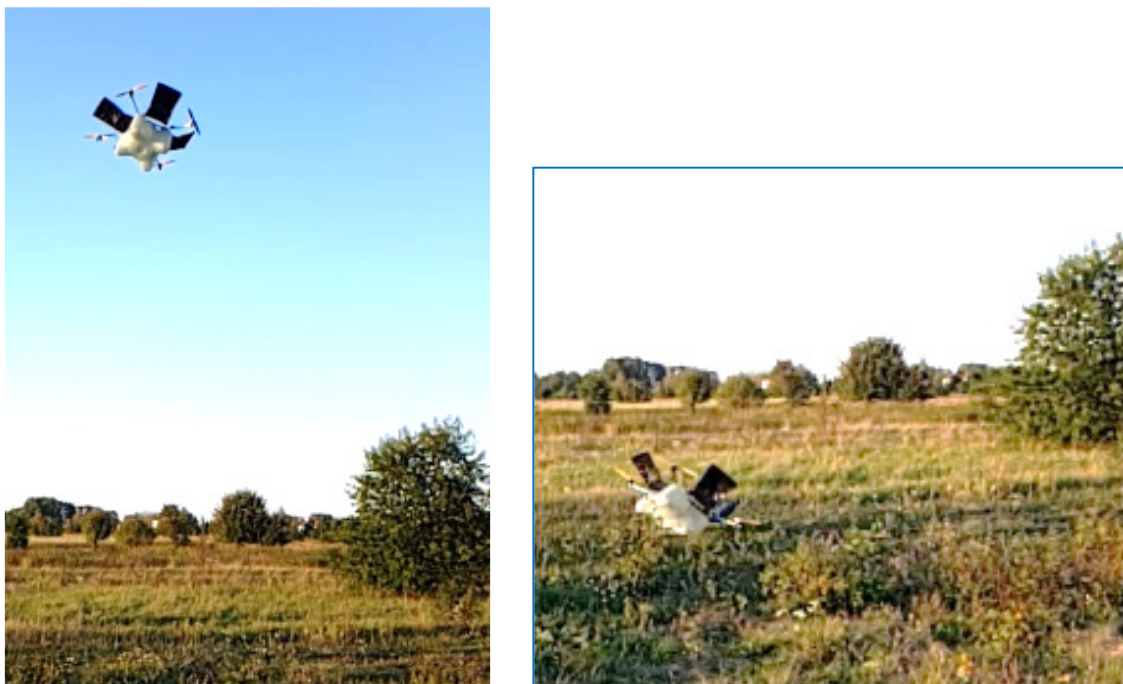


Figure 4. Results of the conducted flight test: a) free fall of the drone (propellers off, airbag inflated), b) start of the collision with the ground

Summarizing the conducted research, the AdBag System allows for the absorption of the total energy of a falling drone from an arbitrary height, with a minimum operational height of 4 m. Depending on the height at which the drone experiences a malfunction, one, two, or three vent holes are activated during collision with the ground, which provides efficient system adaptation. During both laboratory and flight tests, the AdBag System operated fully satisfactorily, provided absorption of the entire impact energy and efficient UAV protection.

3. ADAPTIVE AIRBAG FOR SKYLEADER 600 AIRCRAFT

An innovative adaptive external airbag to enhance the safety of ultralight aircraft (up to 600kg) has been developed, manufactured, and thoroughly tested within the framework of the SOFTLAND project. The fabricated prototype airbag was aimed to fulfil the following fundamental requirements set by the aircraft manufacturer:

- Possibility of using in conjunction with a parachute;
- Reduction of vertical touchdown velocity from 7.5 m/s (descent with a parachute) to 2.5 m/s;
- Limitation of the deceleration to 5 g;
- Minimal mass, up to a maximum of 15kg;
- Deployment time below 13 seconds (approx. 100 m descent with the parachute);
- No collision with the aircraft landing gear;
- Compatibility with other currently utilized aircrafts.

The capabilities of two different methods for inflating the emergency airbag were examined. In the first method, the inflation was achieved by utilizing high-performance fans, and the entire airbag volume was filled with air from the atmosphere. A significant advantage of this approach was the ability to replenish gas losses, thereby eliminating the requirement for the airbag to be completely airtight and enabling its construction through sewing technology. In turn, the second tested inflation method relied on the use of cylinders with compressed gas. In such a case, due to the limitations of system mass, it was necessary to apply solution requiring minimal gas storage. Thus, the inflated airbag had to be as small as possible and completely airtight. As constructing an airtight pneumatic structure using sewing technology proved to be highly challenging, a design consisting of two layers of fabric was proposed: an internal layer to ensure air-tightness and an external sewn layer to provide mechanical strength.

Moreover, various methods of the release of gas from the emergency landing airbag were analyzed: an active (electromagnetic) valve and a passive valve opened by gas pressure. To compare the effectiveness of both valves, the preliminary drops were performed on a model equipped with an electromagnetic valve, and compared to those obtained for a passive valve. The obtained results (braking force, braking distance, and maximal deceleration) appeared to be relatively similar and to fulfil the initial requirements. These allowed to resign from the use of active valve, which was potentially prone to failures and required power supply, in favour of a passive valve.

The proposed passive valve was automatically activated during the airbag compression process. The triggering factor was the change of internal pressure inside airbag, which caused change of airbag geometry and corresponding change of internal forces in the airbag envelope. The critical aspect of constructing a passive valve was determining the safe force level during initial airbag inflation, as well as the force level triggering the valve during the airbag compression phase.

The developed large-scale drop testing stand (Fig. 5) allowed for dropping of the objects with the size and weight of the Skyleader 600 aircraft at the required velocity of 7.5 m/s.

The stand was equipped with systems for measurement of pressure inside the airbag as well as accelerations and displacements at selected locations of the aircraft. In particular, acceleration measurements were performed using accelerometers installed on the rigid components of the model aircraft. The drop tests were conducted using the Skyleader 600 model aircraft provided by the aircraft manufacturer, Zall Jihlavan Airplanes, and constructed according to the specifications of Adaptronica company. The applied model faithfully replicated the geometry and dimensions of the lower part of the Skyleader 600 aircraft.

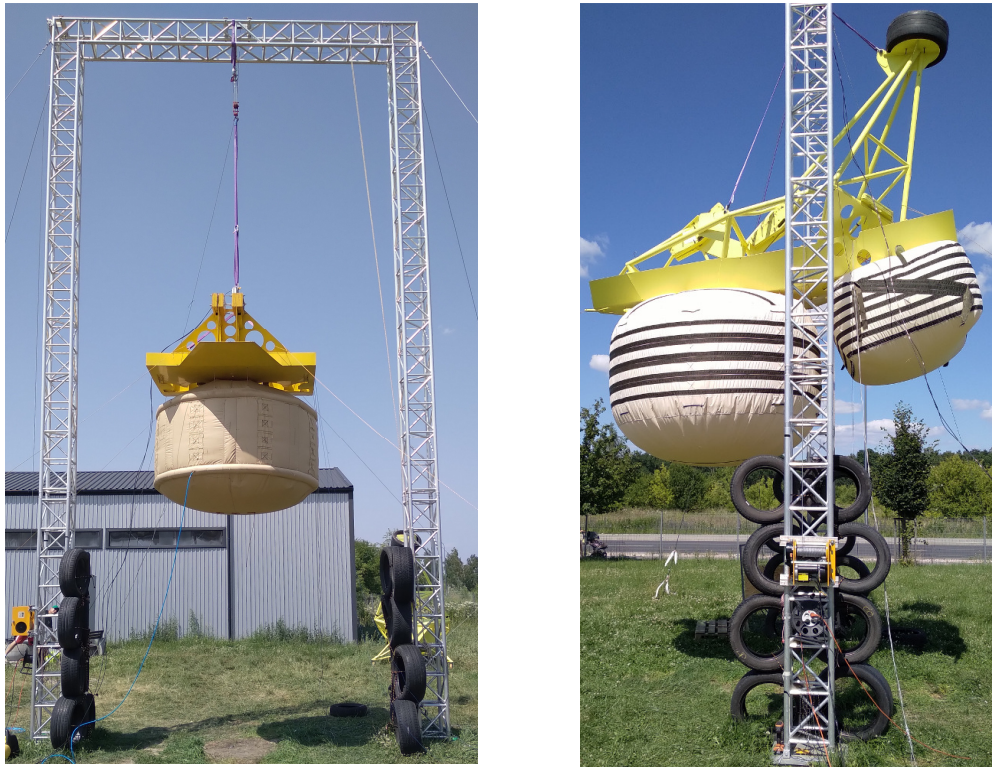


Figure 5. Drop tests conducted at the experimental stand

The total of 31 full-scale drop tests were conducted, including 21 tests with the model aircraft. The drop height was chosen to achieve a vertical velocity of 7.5 m/s at the moment of the contact of the emergency landing airbag's with the ground. During testing, the following various characteristics and parameters of the airbag were taken into account:

- Airbag size (volumes ranging from 3-5 m³);
- Airbag shape (cylindrical, rectangular, rosette-shaped);
- Airbag inflation method (an electric fan, compressed air cylinder, hybrid solution combining both methods).

The results obtained from the subsequent drop tests were used to verify the functioning of each analyzed design type of the SOFTLAND system. In particular, the change of object's velocity and acceleration in terms of distance to the ground was presented in Fig. 6. It can be observed that after landing the object's velocity drops below 1.5 m/s, which is lower than the required safety level equal to 2.5 m/s (Fig. 6a). Moreover, the deceleration of the landing object exceeds safety level of 5 g only during a very short period of the landing process (Fig. 6b). Finally, the obtained deceleration-displacement characteristics remains relatively flat during the entire process, which confirms high efficiency of the energy absorption.

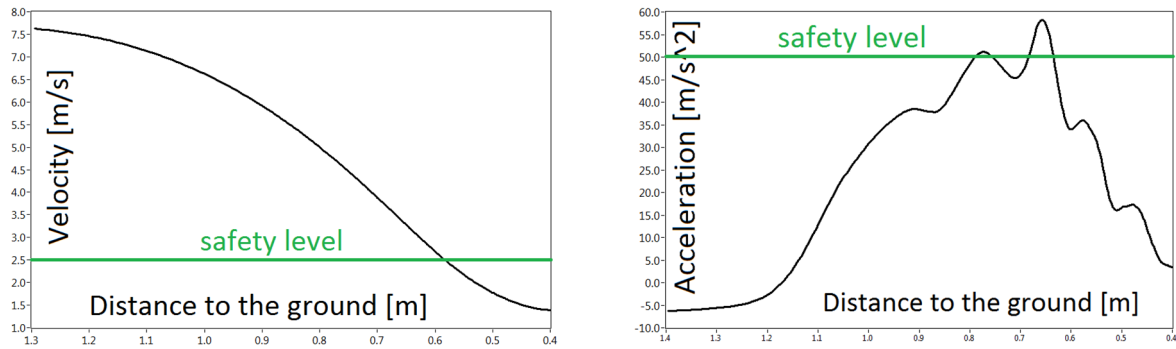


Figure 6. Measurements recorded during the drop test: a. object velocity as a function of distance to the ground, b. object acceleration as a function of distance to the ground

In the subsequent phase of the project, the engineers involved in the development of the Softland system visited several times the Zall Jihlavan Airplanes factory in Czech Republic. During these visits, the system of adaptive airbag was installed on Skylader airplane (Fig. 7a). The verification tests proved proper system operation, including fast deployment and inflation of the air airbag in industrial conditions (Fig. 7b).

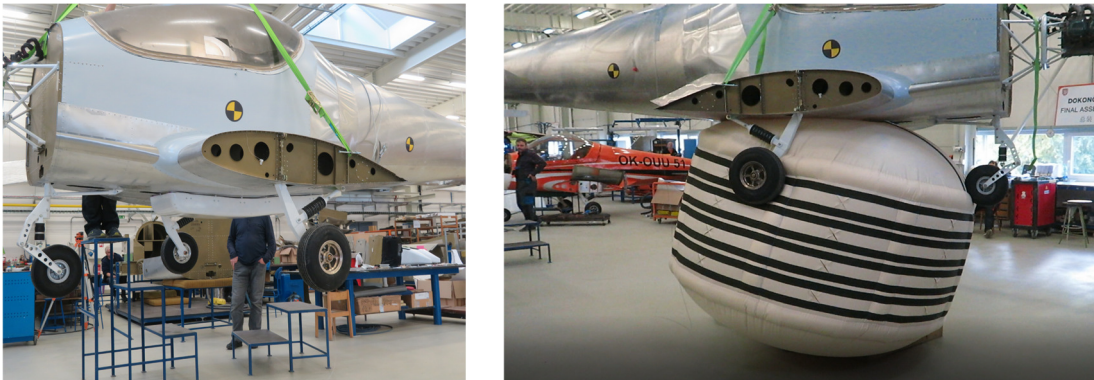


Figure 7. The Softland system installed on the aircraft: a. airbag in the closed state, b. airbag in the open state

The flight test was conducted at the Milovice Airport in the Czech Republic. The helicopter lifted the Skylader aircraft equipped with the parachute and Softland system to an altitude of approx. 400 meters (Fig. 8a). When the operator released the attachment, the parachute opened and Softland system was deployed correctly. Unfortunately, the ropes suspending the aircraft were improperly attached, resulting in a landing with a significant vertical rotation angle. As a result, the front part of the aircraft absorbed the energy from the initial impact. Nevertheless, the Softland system absorbed the impact energy in the second stage of the process, protecting the aircraft from further damage.

From the perspective of evaluating the Softland system, the following aspects of the airbag's operation were positively verified:

- Effective initiation of the safety system by deploying the parachute from the compartment located behind the cockpit;
- Proper time instant of initiating the deployment of the airbag;
- Proper time required for the airbag deployment under aerodynamic flow conditions (suitable for an aircraft descending with a parachute);
- Correct shape of the airbag after deployment under aerodynamic resistance.



Figure 8. View of the aircraft equipped with the Softland system during a flight test

The effectiveness of energy absorption during the aircraft's collision with the ground was very hard to assess due to the abnormal course of the final phase of the experiment. Nevertheless, the measured maximum acceleration of the aircraft structure was below 5 g, which perfectly fits within the initial requirements of the project.

4. ADAPTIVE AIR-DROP SYSTEM

New systems for specialized airdrop operations are continuously being developed and utilized in military, commercial and rescue applications. Each application entails specific requirements concerning system design and performance. Furthermore, landing conditions can vary significantly, necessitating the airdrop system to be adaptable to various impact scenarios. This section discusses the development of a novel, versatile airdrop system that achieves high performance by an innovative construction and deployment mechanism. The proposed system ensures efficient impact mitigation during touchdown and enables effective safeguard of the transported cargo, particularly in scenarios in which the drop height remains relatively low.

According to the proposed concept, the capsule consists of a base with openings and cylindrical pneumatic chamber attached from its underside. The pneumatic chamber contains a stabilizing telescopic system, which maintains the shape of the pneumatic chamber. The telescopic system is equipped with a specially designed tensioning structure, which effectively supports system deployment but does not hamper its compression during landing and effectiveness of the impact absorption process. The detailed specific features of the system are currently being developed and patent claims are being prepared.

The objective of the preliminary tests is to assess the free fall speed of the device, which has a fixed mass of 2 kg and is equipped with a parachute. These tests aim to provide information regarding the time needed for object's velocity to stabilize and the value of constant free-fall velocity. Such information is required for computation of the amount of kinetic energy that has to be dissipated during touchdown.

The research team conducted a total of 5 tests involving the dropping of the capsule from 3 different heights: 4m (1 attempt), 8m (1 attempt), and 25m (3 attempts). At each planned height, the suspended capsule was remotely released, and its movement was captured using a conventional mobile phone. The recordings were taken against the backdrop of the modular structure, serving as a reference mesh, on the facade of the IPPT PAN building. Three selected snapshots captured during the test drop from a height of 25 m

are shown in Fig. 9, depicting the initial position before releasing the capsule, the mid-flight stage, and the aftermath of touchdown. The corresponding velocities obtained from the tests are presented in Fig. 10. The maximum velocity components recorded during the drop tests are summarized in Table 1.

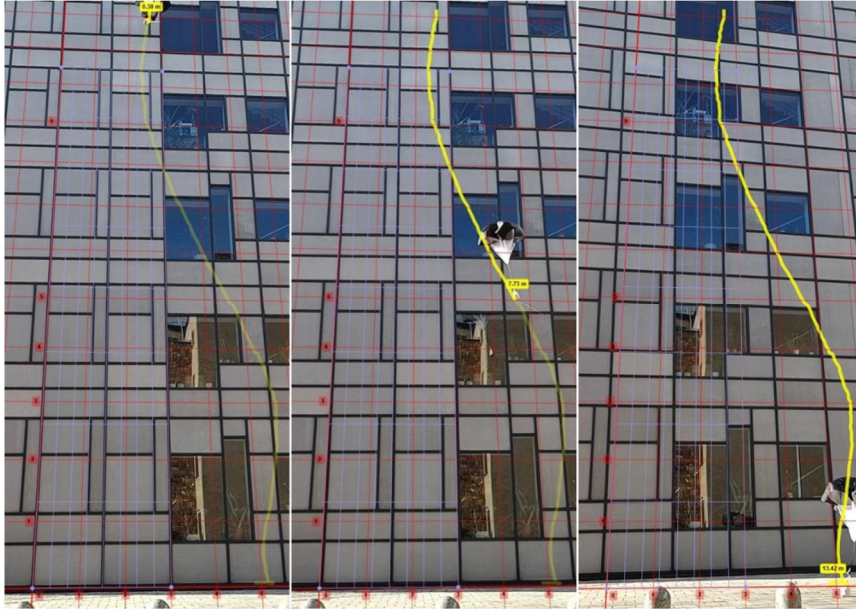


Figure 9. Highlighted snapshots showcasing the trajectory of the device during free fall from a height of 25 m.

The recorded images were analyzed using Kinovea software which was used to extract the kinematic data regarding the capsule motion. For each drop test, horizontal, vertical, and total velocities were determined. The analysis assumed two key conditions: (a) the movement of the load occurs within the designated vertical plane, and (b) the recording camera remains stationary relative to the chosen reference system. These conditions facilitated the accurate determination of the results.

The tests conducted from lower heights (4m and 8m) displayed a systematic but irregular rise in the total descent velocity, reaching approximately 5 m/s (4m) and 7 m/s (8m) respectively. In both cases, there was a noticeable increase in speed just before the contact with the ground. However, Table 1 demonstrates that the descent speeds stabilized below 9m/s for the drop test conducted from a height of 25 m.

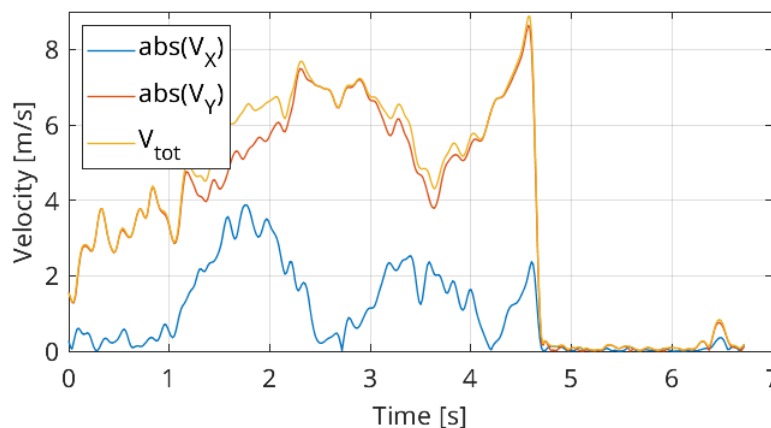


Figure 10. Velocities calculated for the drop tests conducted from a height of 25 m.

Table 1. Velocities obtained from drop tests.

Max. velocity [m/s]	Height of test drop				
	4m	8m	25m	25m	25m
Horizontal	2.60	2.15	3.06	2.59	3.88
Vertical	4.09	6.81	7.05	8.31	8.64
Total	4.74	7.12	7.48	8.50	8.88

5. CONCLUSIONS

Safety systems incorporating airbags for passenger protection have been used in the automotive industry since the 1970s. Currently, new solutions based on external airbag systems are being developed for protection of vehicles and pedestrians during road collisions. The systems for energy dissipation presented in this paper constitute a continuation of this technical concept, but tailored to the aviation industry. Although these solutions are at various stages of development, they reveal a high potential for practical implementation.

The first proposed system is AdBag, an external airbag designed specifically for drones, which are often equipped with specialized and expensive measurement devices. Protecting such flying objects is of great interest to both operators and manufacturers of these devices. Conversely, in the case of ultralight aircrafts the application of a safety system is oriented not only towards safeguard of the structure but also towards protection of human life, which constitutes a compelling rationale for their implementation. On the other hand, incorporating new systems in aviation is subjected to rigorous industry procedures and requires holistic testing to confirm their reliable operation. The final developed solution is an aidrop capsule, which can be successfully applied in rapid delivery of critical and life-saving payloads in emergency situations, such as food in disaster-stricken areas or medications during rescue missions in high-altitude regions. A wide range of conducted research, encompassing both numerical, laboratory, and field studies, contributes significantly to the development of new airbag-based safety systems for general aviation.

ACKNOWLEDGMENTS

The article was prepared with the support of the projects "Adaptive airbags for unmanned aerial vehicles emergency landing (ADBAG)" co-financed from the European Regional Development Fund within the Operational Programme Smart Growth 2014-2020, POIR.01.02.00-00-0083/16 and the project "SOFTLAND – Airbag system for increasing the small aircraft safety during emergency landing", E113407. The authors acknowledge the support of the National Centre for Research and Development and the National Science Centre, Poland, granted in the framework of the TANGO 4 programme (project TANGO-IV-C/0001/2019-00). The support of the National Science Centre, Poland, granted through the Agreement 2018/31/D/ST8/03178 is gratefully acknowledged.

REFERENCES

- [1] J. Holnicki-Szulc, Ed., *Smart technologies for safety engineering*, Wiley, 2008.

- [2] J. Holnicki-Szulc, C. Graczykowski C., G. Mikułowski, A. Mróz and P. Pawłowski, Smart Technologies for Adaptive Impact Absorption, *Solid State Phenomena*, 154, pp. 187-194, 2009.
- [3] J. Holnicki-Szulc, C. Graczykowski, G. Mikulowski, A. Mroz, P. Pawlowski, R. Wiszowaty, Adaptive impact absorption - the concept and potential applications, *International Journal of Protective Structures*, vol. 6(2), pp. 357-377, 2015.
- [4] K. Sekuła, C. Graczykowski, J. Holnicki-Szulc, On-line impact load identification, *Shock and Vibration*, vol. 20(1), pp. 123-14, 2013
- [5] Z. Wołęjsza, J. Holnicki-Szulc, C. Graczykowski, K. Hinc, R. Faraj, T. Kowalski, G. Mikułowski, K. Kaźmierczak, R. Wiszowaty, P. Pawłowski, Dynamics and control of adaptive airbags for UAV impact protection, ISMA 2018 / USD 2018, *International Conference on Noise and Vibration Engineering 2018-09-17/09-19, Leuven (BE)*, pp.3661-3670, 2018
- [6] J. Warrick, C. Lee, "Advanced airbag system for cargo airdrop", 16th AIAA Aerodynamic Decelerator Systems Technology Conf. and Seminar, Boston, USA, 2001, pp. 293-303
- [7] J. Richert, D. Coutellier, C. Götz, W. Eberle, "Advanced smart airbags: The solution for real-life safety?", *Int. J. Crashworthiness*, vol. 12, pp. 159-171, 2007
- [8] G. Mikułowski, R. Wiszowaty, Pneumatic Adaptive Absorber: Mathematical Modelling with Experimental Verification, *Mathematical Problems in Engineering*, ISSN: 1024-123X, vol.2016, pp.7074206-1-14, 2016
- [9] R. Faraj, G. Mikułowski, R. Wiszowaty, Study on the state-dependent path-tracking for smart pneumatic shock-absorber, *Smart materials and Structures*, ISSN: 0964-1726, vol.29(11), pp.115008-1-25, 2020
- [10] L. Knap, M. Makowski, K. Siczek, P. Kubiak, A. Mrowicki, Hydraulic Vehicle Damper Controlled by Piezoelectric Valve, *Sensors*, 23 (4), art. no. 2007, 2023
- [11] P. Pawłowski, M. Ostrowski, C. Graczykowski, High performance valves for adaptive inflatable structures with flow driven control, 6th Eccomas Thematic Conference On Smart Structures And Materials, 2013-09-03/09-06, Torino (IT), pp.1-10
- [12] R. Faraj, C. Graczykowski, J. Holnicki-Szulc, Adaptable pneumatic shock absorber, *Journal of vibration and Control*, ISSN: 1077-5463, vol.25(3), pp.711-721, 2019
- [13] M. Fanton, H.V. Alizadeh, A. Domel, M. Devlin, M. Kurt, G.M. Mungal, D.B. Camarilo, E. Hawkes, "Variable area, constant force shock absorption motivated by traumatic brain injury prevention", *Smart Mater. Struct.*, vol. 29(8), 085023, 2020
- [14] R. Faraj, B. Popławski, D. Gabryel, T. Kowalski, K. Hinc, Adaptive airbag system for increased evacuation safety, *Engineering Structures*, ISSN: 0141-0296, vol.270, pp.114853-1-5, 2022
- [15] T. Kowalski, R. Faraj, C. Graczykowski, K. Hinc, G. Mikułowski, P. Pawłowski, Z. Wołęjsza, Development of adaptive airbags for emergency landing of small uav, *SolMech 2018, 41st Solid Mechanics Conference, 2018-08-27/08-31, Warszawa (PL)*, pp.1-2, 2018



OPTIMAL MULTI-SENSOR OBSTACLE DETECTION SYSTEM FOR SMALL FIXED-WING UAV

Marta Portugal and André C. Marta

IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
{marta.portugal, andre.marta}@tecnico.ulisboa.pt, <https://mdo.tecnico.ulisboa.pt/>

Abstract. *This work provides a solution for the safety enhancement of small fixed-wing UAVs regarding obstacle detection during flight. The main goal is to implement an optimal multi-sensor system configuration. To achieve it, preceding works regarding the integration of available sensors in such systems were studied. As a result, select sensors (ultrasonic sensor, laser rangefinder, LIDAR and RADAR) were modeled for collision detection and avoidance simulations using the potential fields method. An optimization study using a genetic algorithm was conducted to find the sets of sensors and respective orientation that result in the best collision avoidance performance. To do so, a set of randomly generated collision scenarios with both stationary and moving obstacles were generated. This study resulted in relatively simple detection configurations that still provided high collision avoidance success rate. The ultrasonic sensor revealed to be inappropriate given its short range, while the laser rangefinder benefited from long range but had very limited field-of-view. In contrast, both the LIDAR and the RADAR are the most promising, as they exhibit not only a significant range but also a broad field-of-view. The best multi-sensor configurations were either a front-facing LIDAR or RADAR, complimented by a pair of laser rangefinders pointing sideways at an angle of 10 or 63 degrees, respectively. Once the hardware that should integrate an optimal system was known and available, the assembly of the final system, including the sensors and a PixHawk flight controller, was designed. The appropriate software (PX4 and QGroundControl) was also built and adapted to the current work. To validate the proposed system, all sensors were first individually tested before assembling the complete system. The bench tests attested the accuracy of the sensor specifications and previous simulations. As such, ground tests using a simple rover shall follow. Once the system is validated under these conditions, flight tests may begin.*

Keywords: Sense and avoidance, collision avoidance, sensor fusion, optimization, laser rangefinder, LIDAR

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have received considerable attention in a myriad of operations due to their enhanced stability and endurance. Despite being initially developed for military purposes [1], there has been a notable upsurge in the civilian market for UAVs [2].

Some applications of UAVs present high collision risk. Due to their ability to work in a collaborative and cooperative manner, swarms of drones are typically used for surveillance purposes, tracking and localizing objects. One of the most significant challenges regarding the navigation of a swarm of agents is collision avoidance. Collision avoidance systems are responsible for guiding an autonomous agent to safely and reliably avoid potential collisions with other agents in the swarm as well as with other objects in the environment. The capacity to locally sense and avoid items in the environment becomes more crucial for agents to be fully autonomous and, in turn, for systems to be more robust [3]. Drones are also required to exhibit a practical resolution for a Sense and Avoid feature as part of the NextGen [4] strategy for integrating UAVs into the U.S. National Airspace System (NAS). In fact, all UAVs must deploy an automated Sense and Avoid intelligent system that provides safety levels comparable to or even superior to those of manned aircraft [5].

Fittingly, this work specifically addresses the safety enhancement of small fixed-wing UAVs (maximum take-off weight < 25 kg, range < 10 km, endurance < 2h and flight altitude < 120 m), particularly with regard to the detection of obstacles during flight and the automatically triggered collision avoidance maneuver. It is part of a comprehensive obstacle detection and collision avoidance system, representing a two-stage "sense" and "avoid" problem, being this work more focused on the former. Preceding this work, different detection systems were simulated using laser rangefinders and RADARs in different configurations [6]. Through the Potential Fields method and resorting to an optimization algorithm, a possible configuration of the UAV detection system was reached. Subsequently, ultrasonic sensors and laser rangefinders have been employed in the hardware implementation of an effective Sense and Avoid System on a simple rover [7]. In this work, the main goal is to implement an adapted version of the forementioned systems on a small fixed-wing UAV, integrating an optimal multi-sensor configuration that can include ultrasonic sensors, laser rangefinders, RADARs and LIDARs.

2 SENSOR MODELLING

The next subsections describe different models of active non-cooperative sensors: the ultrasonic sensor, the laser rangefinder, the LIDAR (Light Detection and Ranging) and the RADAR (Radio Detection and Ranging), as illustrated in Fig. 1, followed by their comparative analysis in Tab. 1. Different models were developed by [8] and further adapted to the present work.

2.1 Ultrasonic Sensor

This sensor generates a sound, which is then reflected by the obstacle and recorded by the sensor. If the velocity of the radiated sound in the air medium is known, the distance from the point of greatest reflection to the obstacle can be calculated [13]. Using ultrasonic sensors proves to be advantageous mainly due to the ease at which this simple technology can be sized down. However, because these are proximal sensors, their signal

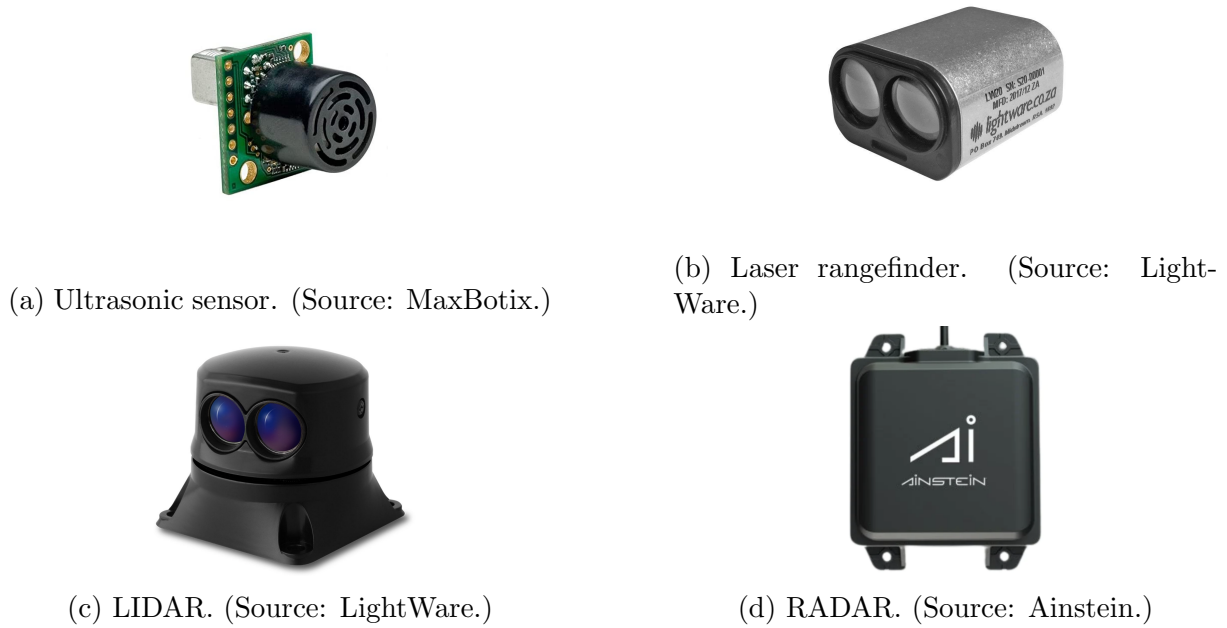


Figure 1: Active non-cooperative sensors.

Table 1: Sensor hardware specifications.

	Ultrasonic sensor MB1242 [9]	Laser rangefinder LW20/C [10]	LIDAR SF45/B [11]	RADAR US-D1 [12]
Range (m)	7	100	45	50
Horizontal FOV (°)	0	0.3	20-320	43
Resolution (cm)	1	1	1	–
Accuracy (m)	0.1	0.1	0.1	0.04
Update rate (Hz)	7	388	5000	100
Power supply voltage (V)	3-5.5	4.5-5.5	4.5-5.5	5-5.5
Power supply current (mA)	4.4	100	300	400
Outputs and interfaces	Serial and I2C	Serial and I2C	Serial and I2C, Micro USB	UART, CAN
Dimensions (mm)	22x19x15	30x20x43	51x48x44	108x79x20
Weight (g)	5.9	20	59	110

quickly attenuates and their capacity to measure distance is typically limited to less than 10 meters [14]. This type of sensor has a wide FOV that translates to a beam pattern with axial symmetry, as represented in Fig.2.

Since the ultrasonic sensor only outputs a distance, it leaves all the interior beam points located at a specific distance from the UAV as potential object positions. This results in errors that shall be avoided, as well as other issues that arise from sound reflection. The sound reflection law states that the reflected sound wave’s angle with the normal of the surface is preserved. Thus, the ultrasonic sensor requires a perpendicular surface in order to detect an object, which in turn implies that the targets format is crucial to the mission’s success. It is vital to recognize the final results can only be used as

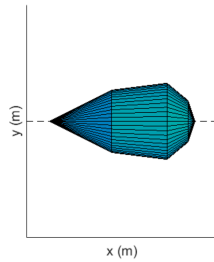


Figure 2: Ultrasonic sensor beam pattern [7].

a reference given that these simulations only use spherical-shaped targets (and different target formats could either improve or worsen the outcome). In short, the model must check for these possibilities at all times:

1. The presence of any spherical surface point within the sonar beam pattern;
2. The perpendicularity of the sound wave direction with its reflecting surface.

Verifying these conditions requires considerable computing time. Therefore, a progressively complex approach that avoids unnecessary blocks of code was implemented [7]. First, the beam pattern is reduced to a cylinder. When the center of the obstacle is found to be inside the cylinder, a more thorough analysis is performed to identify which portion of the spherical surface, if any, is in fact inside the beam pattern. The last stage addresses the perpendicularity issue. The final surface computed in the preceding phase is defined as a list of points with a sampling ratio 25 times bigger for each spherical coordinate (in relation to the first list of points).

2.2 Laser Rangefinder

Laser rangefinders are able to compute distances to obstacles by emitting a laser pulse and measuring the time it takes for the reflected beam to be detected, given that laser light beams move at a known speed. This principle is quite common among sensors, accounting for lightweight, low-cost technology [15]. However, it is limited by weather conditions, as laser light might scatter in the presence of clouds, fog or atmospheric attenuation.

Given that all sensors' models may be implemented at an angle β relative to the longitudinal axis, our model assumes the use of two symmetrical sensors at the angles β and $-\beta$ whenever $\beta \neq 0$.

Considering the obstacles as spheres, this can be modeled as a simple interception between a line and a spherical surface given by

$$\|\mathbf{x} - \mathbf{c}\|^2 = r^2 \quad (1a)$$

$$\mathbf{x} = \mathbf{o} + d\hat{\mathbf{u}}, \quad (1b)$$

where \mathbf{x} is a generic point on the line and/or sphere, \mathbf{c} is the centre point of the sphere, r is its radius, $\hat{\mathbf{u}}$ is the unit vector that defines the line direction in 3D space and d is the distance from the origin of the line. Combining both equations leads to an easily solvable quadratic equation,

$$d^2(\hat{\mathbf{u}} \cdot \hat{\mathbf{u}}) + 2d[\hat{\mathbf{u}} \cdot (\mathbf{o} - \mathbf{c})] + (\mathbf{o} - \mathbf{c}) \cdot (\mathbf{o} - \mathbf{c}) - r^2 = 0, \quad (2)$$

that returns a solution if $0 < d_{\text{sol}} < \mathbf{R}_d$. In real conditions, the laser would not reach the furthest point, reflecting on the closest one. Therefore, if there are two solutions in this interval, only the smallest one prevails. The reflection point with the spherical surface can be easily obtained from Eq. (1b).

2.3 LIDAR

Light Detection and Ranging (LIDAR) emits short and precise laser light impulses with high frequency, that in turn, are reflected and received again by the sensor, measuring the time it took for them to return. Although this technology is similar to the laser rangefinder's, it is multidirectional. Thus, its execution goes beyond simply detecting an obstacle's range and 3-D point cloud can be acquired through a vast array of distance measurements.

The LIDAR model is very similar to the laser rangefinder's. As such, only the points that are closest to the sensor are detected. This implies that if an object is totally visible, it is considered that its half was detected and the remaining of the obstacle is reconstructed assuming symmetry, where the center of symmetry is the medium point of the segment connecting the first and last point of the cluster. In the present simulations, this distance corresponds to the diameter of the obstacle. This model discards obstacles that are hidden or outside FOV.

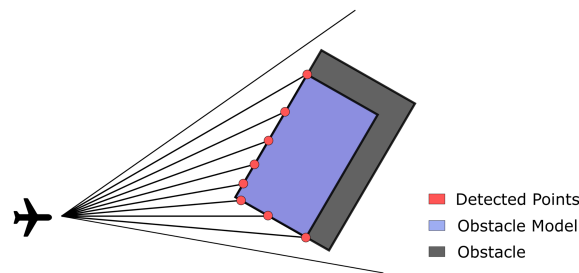


Figure 3: Obstacle reconstruction using a LIDAR [8].

A common issue lies within the higher distance between consecutive points in farther obstacles which results in smaller detected dimensions (see Fig. 3). To solve this problem, the measured diameter is passed through the time filter [16],

$$D_k = D_{k-1} + G(D_m - D_{k-1}), \quad (3)$$

where $G(0 < G < 1)$ is the filter gain, D_k is the filtered diameter at instant t_k , D_{k-1} is the filtered diameter at instant t_{k-1} , and D_m is the measured dimension at instant t_k . The gain must be carefully chosen because it affects how quickly the dimensions change. While a small gain (*i.e.*, slow variation) is better for noisy surroundings, it is not appropriate for objects with high relative speeds. The gain is given by

$$G = 1 - \sqrt[n]{1 - p}, \quad (4)$$

where p corresponds to a fraction that represents the desired accuracy of the dimensions and n corresponds to the number of filter cycles required to get an accuracy of p . Classic Kalman filters [17] were employed for the tracking phase, where the motion of obstacles

was assumed to be two-dimensional, linear, and constant over successive scans. This simplification, which takes into account a high scanning frequency, accurately captures the targets' state.

2.4 RADAR

Radio Detection and Ranging (RADAR) is one of the most popular sensing technologies. It consists of a transmitting antenna producing electromagnetic waves (in the radio or microwave spectrum) and a receiving antenna, which collects waves echoed from static or dynamic obstacles [18]. By measuring the time lapse between the transmitted and received signal, it is possible to determine the distance between the sensor and the target, since radio waves move at a known speed, in a way that can be projected mathematically. Despite being very similar to the LIDAR, RADAR technology is distinguished by the frequency of the emitted radiation.

In this case, the state estimation is more complex than the one employed in the LIDAR model, given the RADAR sensor provides the range, bearing, and elevation of the observed obstacles. These outputs are polar, while the intruder dynamics are best described in rectangular coordinates. Due to its straightforward implementation, the converted measurement Kalman filter (CMKF) was chosen in [8]. The 2-D model used in the simulations shown is represented by

$$\begin{cases} x_m^u &= \lambda_\alpha^{-1} r_m \cos(\alpha_m) \\ y_m^u &= \lambda_\alpha^{-1} r_m \sin(\alpha_m) \end{cases} \quad (5)$$

where (x_m^u, y_m^u) are the measurements converted to the Cartesian frame, r_m is the measured range, α_m is the measured azimuth and λ_α is the bias compensation factor expressed as

$$\lambda_\alpha = e^{-\sigma_\alpha^2/2}, \quad (6)$$

where σ_α is the standard deviation of the noise in the azimuth measurements. The compensation of the bias is multiplicative due to the use of the unbiased conversion and modeling the measurement errors as Gaussian white noise. The covariance matrix used in the Kalman Filter is given by

$$\mathbf{R}_u = \begin{bmatrix} \text{var}(x_m^u | r_m, \alpha_m) & \text{cov}(x_m^u, y_m^u | r_m, \alpha_m) \\ \text{cov}(x_m^u, y_m^u | r_m, \alpha_m) & \text{var}(y_m^u | r_m, \alpha_m) \end{bmatrix}, \quad (7)$$

with the details of the computation of these variances found in [19].

2.5 Multi-Sensor Data Fusion

All of these sensors (and respective models) provide input that allow the avoidance system to actuate. However, if the system's architecture is composed by more than one sensor, the data provided must be merged in some way. Following best practices [20], the weighted filter method is used in the present study. The principle behind this method is simple: each sensor is given a weight that is based on how reliable it is. Reference data sensors that provide information about the UAV state must be installed. Considering that changes in the distance to obstacles correspond to changes in the UAV location, reference data sensors like IMUs and optical flow sensors are used to assess the accuracy of the main

data and aid in selecting the best sensor. In the particular case of fixed obstacles, the aforementioned variances in distance ought to match. The weights are then calculated by applying a differential norm to compare all conceivable sensor combinations of main data and reference data. In each instant, the obstacle distance measurement corresponding to the sensor with the lowest weight is chosen, and the remaining measurements are discarded on the grounds that they are corrupted. Nonetheless, the sensor readings are fused in accordance with their weights if the computed weights have a low variation.

3 OPTIMAL SENSING SYSTEM

An optimization study was conducted to find the types of sensors and respective orientation that result in the best collision avoidance performance. To do so, a set of randomly generated collision scenarios with both stationary and moving obstacles were generated. The sensors modelled in Sec.2 were tested for each of these scenarios, varying their orientation until optimal configurations were reached. The scenario generation algorithm and multi-sensor optimization was further developed based on [8].

3.1 Scenarios Generation

In order to create scenarios that are suitable for this study, a scenario generation algorithm was created. Each scenario must specify the obstacle's initial position, velocity and radius. It also includes a pre-planned path and waypoints that the UAV must follow.

Figure 4 is based on the graphical representation of this algorithm [7], depicting the processes that lead to generating a scenario. Different bounds are defined regarding the kinematic and dimensional properties of the obstacles and the UAV itself. Various stochastic and partially stochastic processes were then extracted from these intervals, creating random values for the different variables.

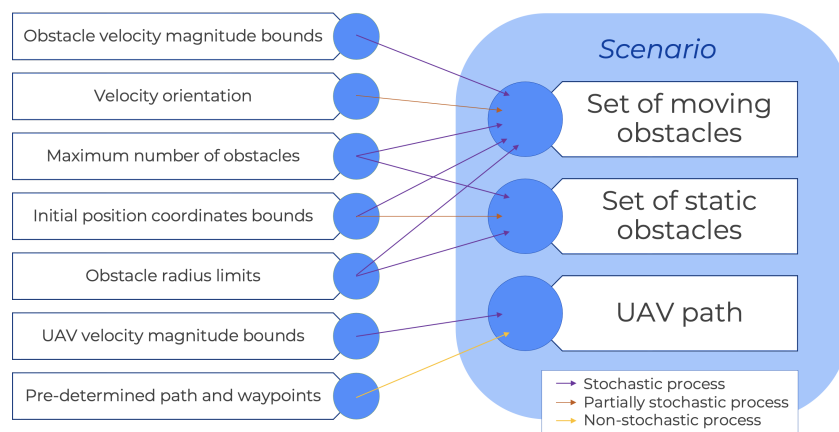


Figure 4: Scenario generation algorithm [7].

Partially stochastic processes have been used in two different cases: determining the velocity orientation of moving obstacles and setting the position of static obstacles. In the former, the goal is to ensure that deviation from the obstacles to the center of the graphical window is not predicted by initial conditions, *i.e.*, initially, the direction of the obstacle's velocity shall point to the centre of the window, rather than pointing outwards, increasing the possibility of collision. In the latter, the initial position of the static obstacle

must not be within the safety radius around the waypoint, given that the UAV must pass through it.

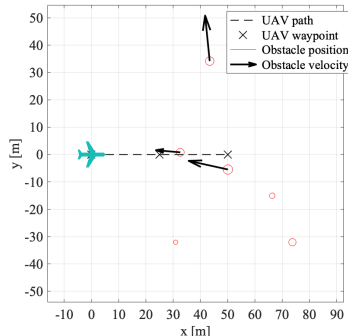


Figure 5: Randomly generated scenario.

An example of a resulting scenario is plotted in Fig. 5. This scenario generating function simply accepts the predetermined path and waypoints of the UAV as an input before combining them with a list of moving and static obstacles to produce a scenario. If the UAV does not go beyond any obstacle’s safety radius throughout the whole simulation (without any sensors), the scenario will be discarded. Until there are n scenarios with an impending collision, this process is repeated.

3.2 Optimization Technique and Problem Formulation

To determine the optimal sensor configuration, different sensor sets were tested. The parameters that characterize each sensor model were obtained from their technical manuals or inferred from available data summarized in Tab. 1. Since our simulations were restricted to the horizontal plane of motion, the vertical FOV is not relevant.

Forty collision-leading scenarios were randomly generated, with obstacle parameters varying according to the limits set in Tab. 2.

Table 2: Data for randomly generated imminent collision scenarios.

UAV speed	# fixed obst.	# moving obst.	obst.radius	obst.speed	obst.direction
[5, 15]m/s	{0, 1, 2}	{0, 1, 2}	[0.5, 2]m	[5, 15]m/s	[0, 90]°

In order to optimize the sensor orientation β , a S&A metric function $f(\beta)$, to be minimized, was defined as

$$f(\beta) = \sum_j \sum_i (-d_{\min}(i) + \phi_1 |\max(R_s(i) - d_{\min}(i), 0)|^2 + \phi_2 |\max(R_c(i) - d_{\min}(i), 0)|^2), \quad (8)$$

where the first term drives the evasion maneuver to maximize the minimum distance d_{\min} between the UAV and the obstacle i , the second term represents the penalty when the minimum distance violates the safety radius R_s ($d_{\min} \leq R_s$), and the last term represents the penalty when the minimum distance violates the obstacle collision radius R_c ($d_{\min} \leq R_c$). The metric accumulates not only for every obstacle i in each scenario but also for all scenarios j . In order to penalize collision cases more than close-calls, the weights were set to $\phi_1 = 10$ and $\phi_2 = 50$.



(a) Pair of laser rangefinders.

(b) Pair of laser rangefinders and a RADAR.

Figure 6: S&A metric as function of laser rangefinder orientation.

Figure 6 shows the metric defined in Eq. (8) for two particular sensor solution cases: i) using a pair of laser rangefinders with a 100m range, symmetrically pointing forward with an angle β with respect to the UAV longitudinal axis; and ii) adding a RADAR with a 120m range pointing in the direction of the UAV longitudinal axis.

In both cases, the metric proves to be noisy. Thus, the optimization technique selected to find the minimum of $f(\beta)$ was the Genetic Algorithm (GA) implemented in MATLAB. This gradient-free, population-based method, deals with a set of solutions that are updated simultaneously in each iteration. In practice, compared to other minimization algorithms, this reduces the likelihood of the result being a relative minimum. This problem can be posed as

$$\begin{aligned} & \text{Minimize } f(\beta) \\ & \text{w.r.t. } \beta \\ & \text{subject to } \beta_{min} \leq \beta \leq \beta_{max} \end{aligned} \quad (9)$$

where β_{min} and β_{max} are the lower and upper bounds of β , respectively, to be defined for each particular case. Notice that β is a vector if multiple sensors are used.

The initial population was set to be created with a uniform distribution; the crossover function was set to create 80% of the population in each generation; because the variables are bounded, the mutation function randomly generates directions that are adaptive with respect to the last successful or unsuccessful generation, where the chosen direction and step length satisfy the set bounds. The convergence criteria were set such that the global minimum was found in a timely but accurate manner: a function convergence of 10^{-3} was used with 10 stall generations, and a maximum of 50 generations prescribed. The population size was set to 30 individuals. These parameters were chosen following best practices. The simulations were run on an 1,4 GHz Intel quad-core i5 with 8 GB 2133 MHz RAM.

3.3 Optimal Sensing Configurations

The following subsections are dedicated to detailing the proposed sensing architectures, further explaining each solution and the respective optimal result. In the end, the performance of the different sensor sets will be summarized and compared, in order to implement the best solutions.

3.3.1 Two Ultrasonic Sensors

For a set of two ultrasonic sensors, the orientation of each sensor was bounded between 0° and 90° from the longitudinal axis and the range was set to 6m. To simplify the problem, the two sonars were considered to have a symmetrical orientation, resulting in just one design variable. A narrow beam pattern was adopted to reduce computational cost.

The GA minimization terminated at 20 iterations, due to average change in the fitness value less than the specified tolerance, after performing 592 function evaluations for 39 hours and 40 minutes. It reached an optimal orientation of 36.5° (see Fig. 7).

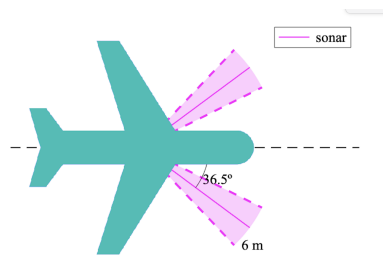


Figure 7: Optimal orientation for two ultrasonic sensors configuration.

The results, summarized in the first line of Tab. 3, are not satisfactory for either case, since the sensor's scanning pattern allows for the safety radius to be breached too many times. This was expected due to the short range of ultrasonic sensors, that makes it impossible for the UAV to detect the obstacle and replan its trajectory in a timely manner.

3.3.2 Two Laser Rangefinders

Analogous to the previous case, a set of two laser rangefinders with symmetrical orientation was considered, but adopting a sensing range of 100 m.

After 19 generations, the GA optimization algorithm finished, corresponding to 564 function evaluations and a computing time of approximately 6 hours. The optimal sensor orientation was 34.4° , which corresponds well with one of the approximate minimum shown in the preliminary study in Fig. 6. The optimal two laser rangefinder sensor configuration is illustrated Fig. 8.

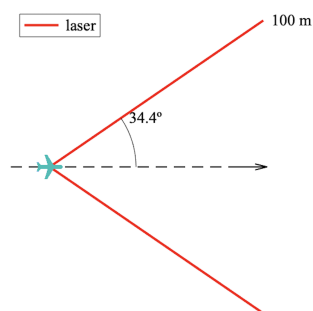


Figure 8: Optimal orientation for two laser rangefinder configuration.

The performance of this optimal configuration is summarized in the second line of Tab. 3. Although the optimal configuration only fails once in 40 scenarios, the safety radius was breached in 23 of them. This result was expected, since a UAV equipped only with two laser rangefinders is not capable of properly tracking the moving obstacles when collisions are imminent.

Compared to the previous case of ultrasonic sensors, these simulations demonstrated that laser rangefinders not only prevent more collisions but also more close calls. Overall, these sensors perform better under the given circumstances.

3.3.3 Two RADARs

Once again, the two RADAR sensors were considered to be symmetrical about the UAV longitudinal axis and the orientation spanned from 0° to 90° . Each RADAR had a range of 50 m, an accuracy of 0.04 m and a FOV of 43° .

After 11 generations, the optimizer finished 340 function evaluations. The optimal RADAR orientation was 9.2° , as illustrated in Fig.9.

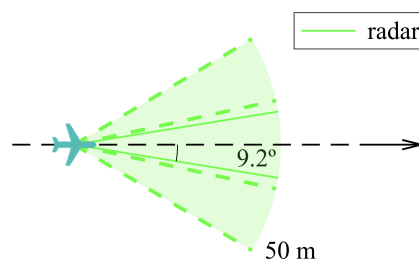


Figure 9: Optimal orientation for two RADAR configuration.

Another configuration worth studying would be a sensor orientation close to 21.5° , which would yield the same result as if the UAV were equipped with a single RADAR with double FOV (86°). Table 3 includes the comparison between this configuration, the optimal orientation and a single RADAR pointing forward.

Regarding actual collisions, obstacles that approach the UAV from an angle are more likely to be detected by the optimal solution rather than by the single RADAR configuration. As can be seen in Tab. 3, the number of failures increase as the orientation decreases (for this particular case), which in turn makes the success rate decrease.

By overlapping the FOV of the two sensors, the accuracy is reduced through the data fusion algorithm. Thus, in this case, having a narrower FOV ($\beta = 9.2^\circ$) and in turn, the juxtaposition of both RADARs proved to be almost as effective as the double FOV configuration ($\beta = 21.5^\circ$).

These simulations showed that the reduced accuracy of the RADAR proves to be impactful on the precision of obstacle tracking compared to that of the laser sensors. Despite having a broader FOV and resulting in less close calls, the RADAR solution led to just as many collisions, which means that the two laser rangefinder configuration remains as promising (same success rate). It is reasonable to say that while RADAR FOV is more crucial for detecting obstacles, the sensor's accuracy is the most significant factor for effective collision avoidance.

3.3.4 Two LIDARs

Each LIDAR was modelled with a range of 45 m, an accuracy of 0.1 m and a variable FOV. According to hardware specifications (see Tab.1), this FOV can range from 20° to 320° , thus, a FOV of 180° was chosen. This value ensures a reasonable trade-off between timely scanning frequency and a broad scope.

However, this makes optimization redundant, as illustrated in Fig. 10. This happens due to the nature of the scenario generation algorithm used: because the obstacles are spawned inside the limits of the scenario, it is worthless to track the area behind the UAV in the initial instant. Furthermore, from this instant on, if an obstacle were positioned behind the UAV, it would have already been tracked before due to the wide FOV and long range of the LIDAR. The overlapping of the FOV in the case of a two LIDAR solution does not prove to be advantageous either. Note that this is only verified for a FOV of 180° . If the FOV were smaller, it would be convenient to optimize the sensor orientation.

In this particular case, it is fair to state that the most beneficial solution would be to use a single LIDAR pointing forward, since it decreases hardware cost. This configuration is illustrated in Fig. 11.

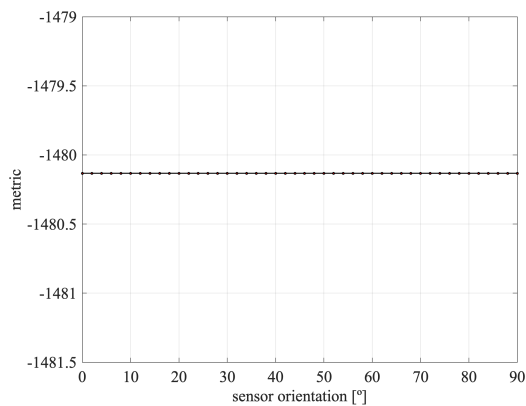


Figure 10: S&A metric as function of sensor orientation for a set of two LIDAR.

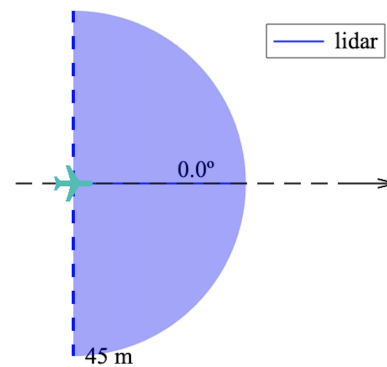


Figure 11: Single LIDAR configuration.

Table 3 includes the performance comparison for different orientations of two LIDAR. As mentioned above, the success rate is the same for both cases. Compared to the previous types of sensors studied, the LIDAR performs better overall. The wide FOV reduces the chances of close calls and eliminates the possibility of failure.

3.3.5 Performance Comparison of Sensor Sets

Other solutions that involved three sensors were optimized, for example, including two laser rangefinders symmetrical about the UAV longitudinal axis, whose orientations were bonded between 0° and 70° ; and one fixed RADAR pointing forward. This configuration was also replicated with two lasers and one LIDAR, two RADARs and one laser, and 2 RADARs and one LIDAR. The performance of the optimal version of these sets of sensors is summarized in Tab. 3, as well as the results from the solutions with only one type of sensor. Optimizations with different sets of sensors were performed but left out of this table in order to avoid redundancy of results.

Table 3: Comparison of the optimal performance for the different sensor sets studied.

Sensors	Metric	Failure	Close call	Success rate
2 SONARs @ 36.5 °	804.0	4/40	30/40	90.0%
2 lasers @ 34.4 °	-414.0	1/40	23/40	97.5%
2 lasers @ 63.4 ° + 1 RADAR @ 0 °	-1240.4	0/40	11/40	100.0%
2 lasers @ 10.0 ° + 1 LIDAR @ 0 °	-1606.4	0/40	8/40	100.0%
2 RADARs @ 9.2 °	-1171.0	1/40	12/40	97.5%
2 RADARs @ 21.5 °	-1141.7	1/40	12/40	97.5%
2 RADARs @ 35.3 ° + 1 laser @ 0 °	-1480.1	0/40	9/40	100.0%
2 RADARs @ 28.1 ° + 1 LIDAR @ 0 °	-1574.3	0/40	9/40	100.0%
1 LIDAR @ 0 °	-1480.1	0/40	9/40	100.0%

For the set of scenarios tested, the RADAR performed better than the laser rangefinder, which in turn performed better than the ultrasonic sensor if only one sensor type is to be used. Nonetheless, this is tightly dependent on the sensor characteristics, such as range, FOV and accuracy. Furthermore, a single LIDAR was enough to outperform all other types of sensor.

As expected, all the solutions that present a 100% success rate include either a RADAR or a LIDAR in their configuration. If the LIDAR is kept out, it is the two RADAR and one laser rangefinder solution that produced the least collisions and led to the least close calls. From these findings, and because of the 0° FOV of a laser rangefinder, it is expected that increasing even more the number of sensors would lead to even better performance, thought at a higher hardware cost.

Comparing the solutions that include a LIDAR, it is proved that it is not significantly advantageous to pair it with other types of sensors, since it already performs distinctively well on its own. Regardless, the two laser and the two RADAR solution are beneficial due to reducing the likelihood of close calls. Despite the LIDAR having a wide FOV that is not increased by either configuration, the chances of breaching the safety radius decrease because the other sensors provide additional detection capacity. *I.e.*, since the LIDAR sweeps the designated area at a certain frequency, there are time instants when a fraction of the area within the LIDAR FOV is 'unsupervised'. Therefore, it is useful to have another set of sensors that track obstacles approaching from that specific area.

To summarize, the optimized configuration had a very similar performance in four different cases (reflected in the *Metric* column), being the most promising one composed of one LIDAR pointing forward, complemented by two laser rangefinders pointing at 10° sideways. These four configurations are illustrated in Fig. 12.

4 HARDWARE AND SOFTWARE IMPLEMENTATION

This section provides an introduction to the flight controller and ground control station, including the steps that must be taken to build and adapt this software to the current work, and the electrical layout of the final system.

Some basic concepts are needed in order to build and fly an unmanned vehicle using PX4 [21]. PX4 is a core part of a broader drone platform that includes the QGroundControl ground station, the Pixhawk hardware, and MAVSDK for integration with companion computers, cameras and other hardware using the MAVLink protocol.

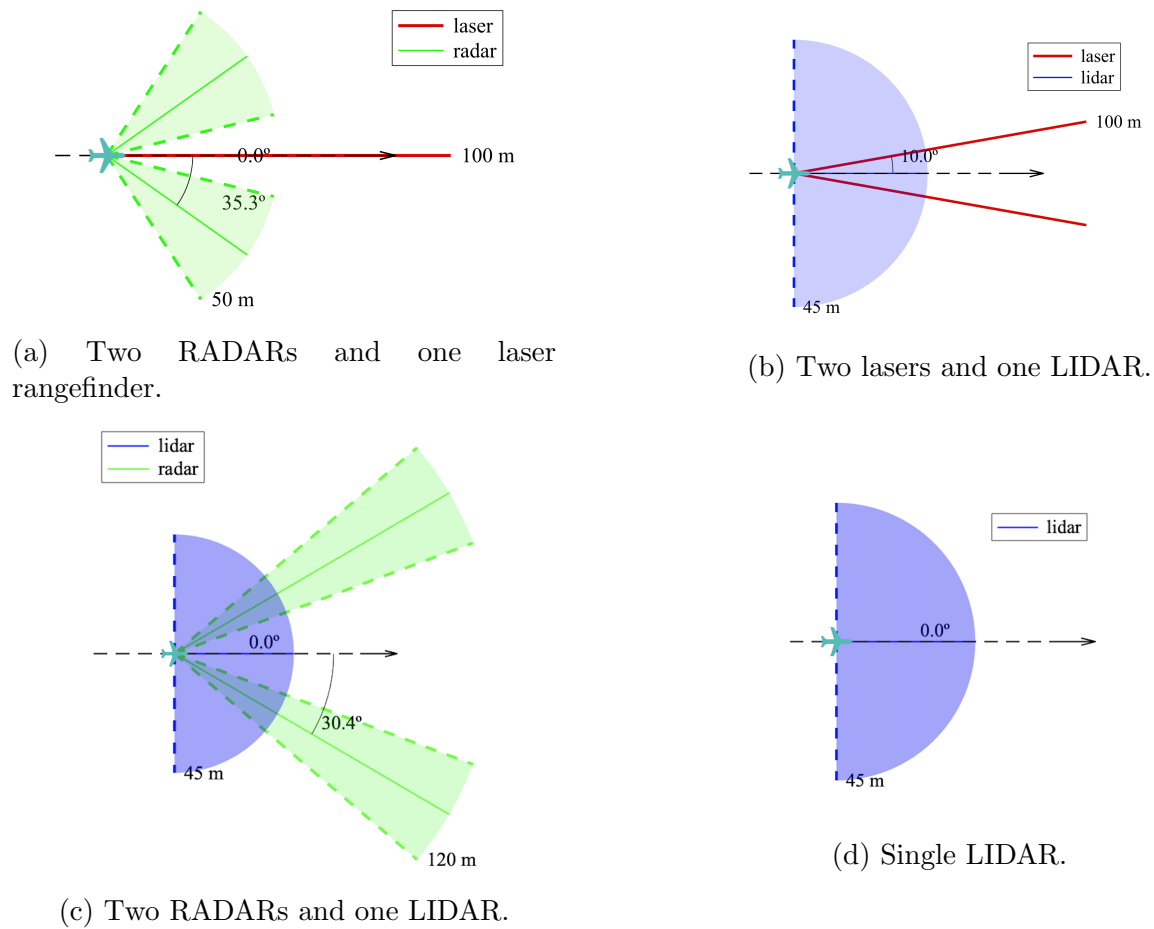


Figure 12: Optimal sensor configurations to be tested.

4.1 Flight Controller

PX4 is a powerful open source autopilot flight stack that can be built on a console or in an IDE, for both simulated and hardware targets. PX4 can be run on various hardware platforms, including Pixhawk. The Pixhawk 2.1, or Hex Cube Black, was the chosen controller in [7], so it will also be used in the current work. Generally, the most recent stable released version of PX4 ought to be used, to benefit from bug fixes and get updated features. As such, this is the version that is installed by default and integrated into QGroundControl. However, the current stable release (v1.13.3) does not include the driver for the LIDAR used in this work. Consequently, it was necessary to switch to a more recent beta release (v1.14) that includes it. The PX4 source code is stored on a Github repository called PX4/PX4-Autopilot. To get release 1.14 and enable the necessary drivers, the PX4 Developer Guide includes tutorials on Building PX4 Software [22] and PX4 Board Configuration [23].

Once these steps are completed, the PX4 Autopilot firmware will be compiled, generating an executable file that can be uploaded onto the flight controller.

4.2 Ground Control

A ground control station works as a virtual cockpit, serving as an interface between a flight controller and a human operator. Typically, a software running on a computer

is connected to the flight controller through wireless telemetry. This enables the human operator to communicate with the aircraft during flight, allowing the acquisition of relevant data such as position, velocity, acceleration, or any other sensor data. It can be installed simply by running the executable file available in the QGroundControl online user manual [24].

The PX4 firmware can be installed onto the flight controller by following the steps in the PX4 User Guide [25]. The user is then prompted by QGroundControl to calibrate the vehicle, including the configuration of the controller's built-in sensors, radio receiver, flying modes, power, and motors.

4.3 Electrical Wiring Layout

The electrical layout can be designed once the hardware is chosen and calibrated.

It is possible to connect all the components as shown in Fig. 13 using the connections and supplementary devices (GPS and power module) included within the Cube Black package. It is necessary to employ a power module to provide the flight controller a regulated power source and power the electronic speed controller (ESC) at the same time. The ESC also draws power from a battery and operates the motor using a PWM signal from one of the PWM I/O entries. A PPM Sum Receiver is also present, and it needs to be connected to an RX IN input. This component converts the PWM signals from the radio receiver into a single PPM signal that the flight controller can process. The telemetry module communicates through radio waves with a second telemetry module that is linked to a ground station. This allows real-time data to be exchanged and orders to be sent to the vehicle. Lastly, the I2C ports can be used to connect sensors, namely the ultrasonic sensor and the laser rangefinder.

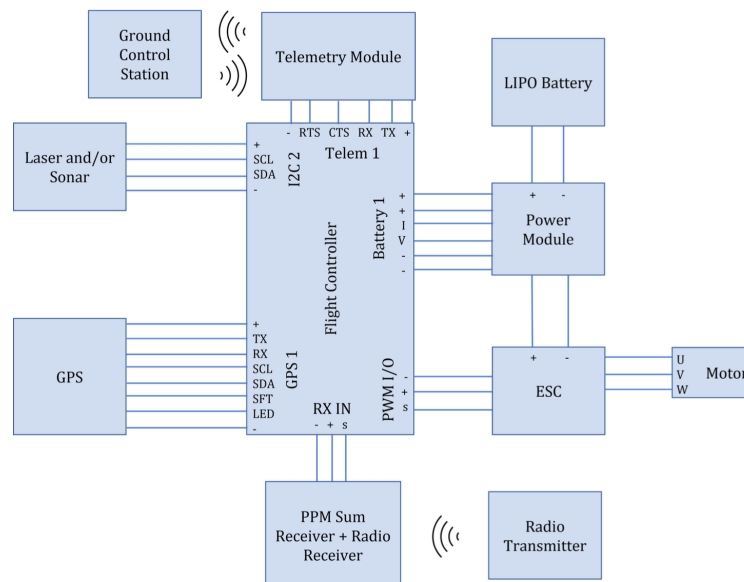


Figure 13: Electrical wiring diagram [7].

5 SENSOR BENCH TESTS

To validate the capabilities of a S&A system, several experiments were performed. More specifically, the sensors that integrate the optimized configurations found in Sec. 3 were individually tested before the complete system. Due to the risk associated with flight testing, experiments were based on ground tests using a simple rover. Once the system is validated under these conditions, flight tests using a small fixed-wing UAV may begin.

5.1 Ultrasonic Sensor

The MB1242 ultrasonic sensor must be connected to the Pixhawk 2.1 and activated in QGroundControl in order to perform the bench tests [7]. These tests included variations in material of the detected obstacle and angles. Figure 14a) demonstrates an experiment where the object to detect is in front of the sensor. In Fig. 14b), the idea is to determine the sonar capability of detecting an object which has an angular deflection (θ) in relation to the sensor.

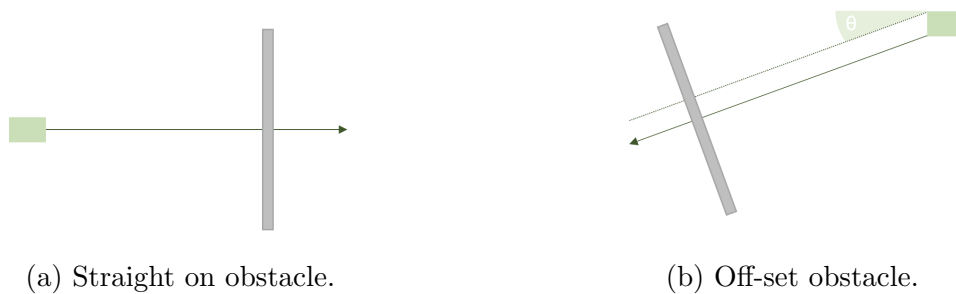


Figure 14: Ultrasonic sensor bench tests.

For these first experiments, the target object is a rectangular wooden board with size 30x25cm as seen in Fig. 15. The frontal test was also repeated with a rectangular XPS board (125.5x60cm) as the target object.

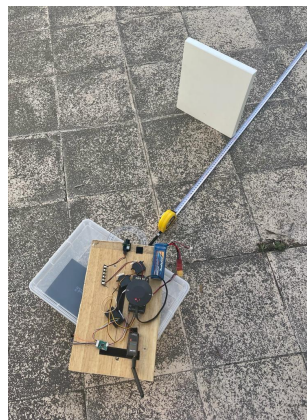


Figure 15: Experimental setup.

During these experiments, each board was positioned at different distances relative to the sonar (20cm to 760cm) and the sensor data was recorded for 30 seconds for each position. This method was aimed at determining, for each position, the fraction of time

where the sensor was actually detecting its target and how much these measurements were deviated from the correct distance.

It is important to consider that the target material could affect the performance of this sensor. The MB1242 data sheet [9] mentions that this sensor's ideal surface to detect is hard, smooth and non-porous. Although wood is not a perfect example of an ideal surface, its properties are not far from that category. To determine whether these conclusions are applicable to other materials. A target made of XPS was also tested, which is not as hard as wood, yet is more porous. Figure 16 represents the detection rate for both materials. When testing with a XPS board, the maximum range decreased and the sensor performed worse overall. This decrease in performance was foreseeable since XPS's properties do not match those of an ideal material (hard, smooth and non-porous).

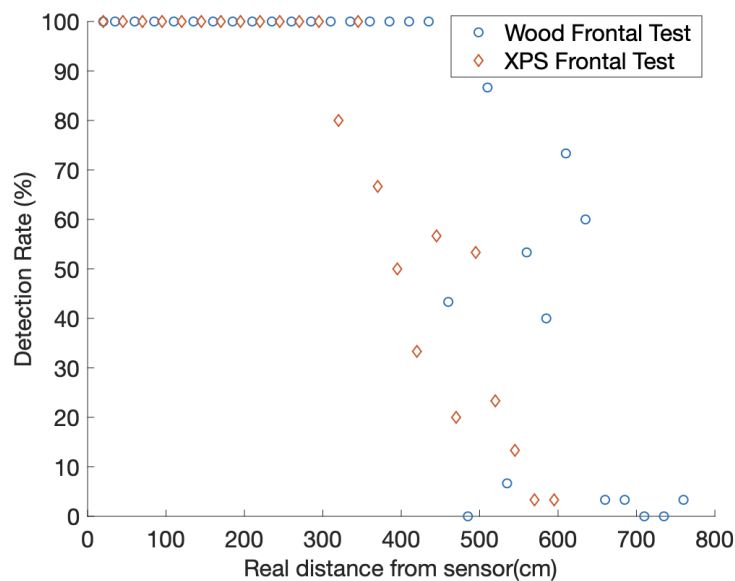


Figure 16: MB1242 detection rate for different materials.

The last significant point is that the performance of this sensor is impacted by the target's rotation within its inertial referential. Only when the sound is reflected back from the target can the sonar identify it. According to the principles of sound reflection, this is only conceivable if the normal vector of the surface in question is parallel to the trajectory of the sound being emitted until it reaches the desired target.

Empirically, this translates to the results that follow. Figure 17 shows the sensor's detection rate for various distances and orientations. As expected, the sensor performed better when the obstacle was completely in front of it, achieving a maximum range of 435 cm with perfect a detection rate, although the datasheet states 640 cm. Additionally, the maximum range decreased when augmenting θ , which was also an expected behaviour. Moreover, this sensor proved to be very directional as it stopped detecting any targets for $\theta \geq 40^\circ$.

Figure 18 shows the average absolute error at each distance from the sensor. No relation between the real distance from the sensor and its error was detected, given that all points seem to be almost randomly dispersed from 300 cm onwards. Moreover, the average absolute error never surpassed 30 cm. This error occurs in the wood frontal test, which leads to the conclusion that this board might not match the ideal surface for this type

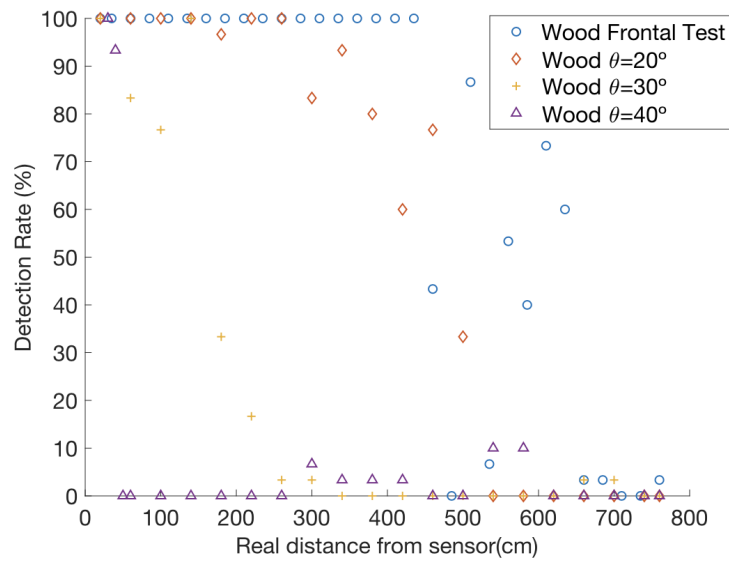


Figure 17: MB1242 detection rate for different angles of incidence.

of sensor to detect. However, the collision avoidance system needs to be robust enough for most materials, *i.e.*, if the available sensor fails this bench test, it is not sufficiently accurate for the purpose of this work.

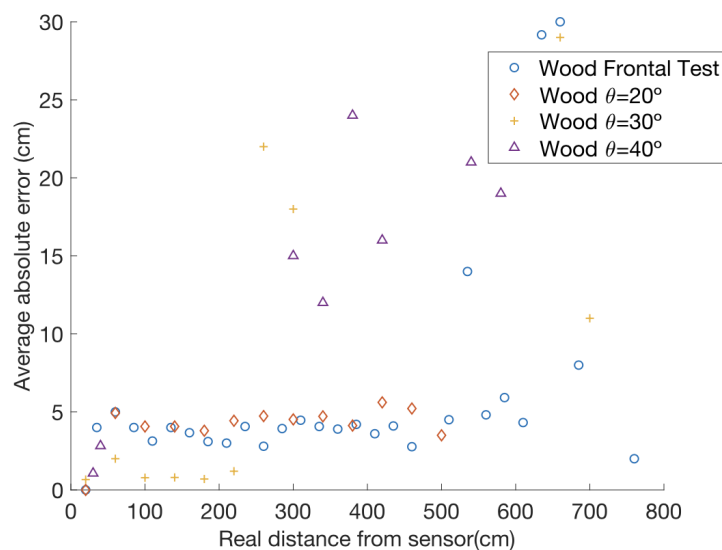


Figure 18: MB1242 average absolute error for different angles of incidence.

According to the datasheet [9], the sensor is calibrated and tested to provide stable range readings to large targets even in electrically and acoustically noisy environments. It also states that the sonar should ideally be used indoors. Nonetheless, it is important to note that these tests were done outdoors, due to the nature of this project. Since this sensor is intended to integrate a collision avoidance system for small UAVs, its applications require that it performs well outdoors. However, this also means that the results shown can and have been affected by exterior noise.

Additionally, an experimental beam pattern was generated using the maximum range recorded for each orientation, shown in Fig. 19. Such beam patterns tend to be particu-

larly advantageous for S&A systems since they restrict their detecting volume, ultimately allowing the controlling device to pinpoint the target's location with a high degree of accuracy.

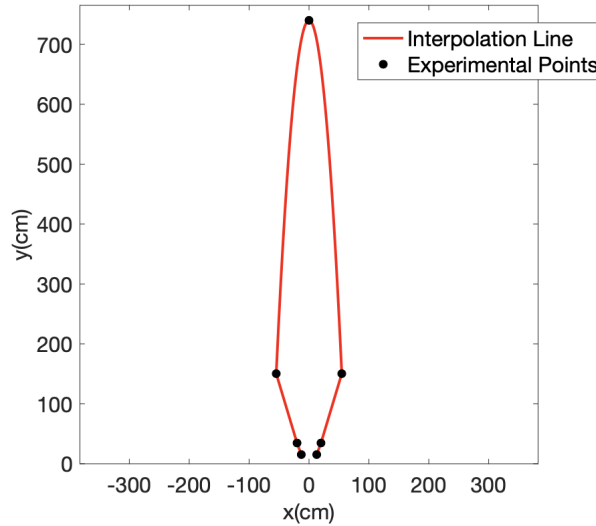


Figure 19: MB1242 empirical beam pattern.

5.2 Laser Rangefinder

Prior to testing, it was necessary to configure the LW20/C laser rangefinder within the flight controller's environment [7]. Once the configuration was done, the performance of the sensor could be assessed through an identical experience to that of the sonar. However, since the laser rangefinder is completely directional, it is not necessary to experiment with off-set obstacles.

In frontal tests, the laser maintained a perfect detection rate before reaching 85 m, as seen in Fig. 20. From this distance onward, the detection rate decreased non-linearly until it reached 100 m (marked as a dashed red line in Fig. 20). Ultimately, the complete range promised in the datasheet was not attained with a perfect detection rate.

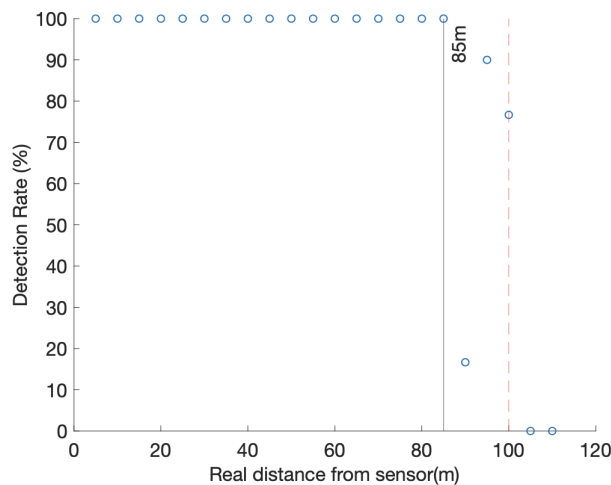


Figure 20: LW20/C detection rate.

These tests also served to prove how much of an impact directionality has on this type of sensor. The wooden board had to be perfectly aligned with the laser rangefinder in order for it to detect it correctly. When translating this to the optimal sensing system designed in Sec. 3, it means the sensor has to be flawlessly aligned with the UAV's longitudinal axis.

Lastly, the average absolute error, plotted in Fig. 21 was mostly between 0 and 50 cm, but increased with the distance from the sensor. The results were not as satisfactory for distances greater than 50m.

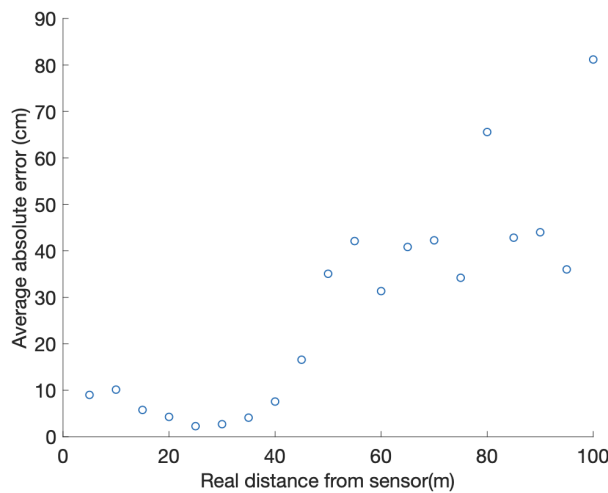


Figure 21: LW20/C average absolute error.

5.3 LIDAR

The chosen LIDAR was the SF45/B model by LightWare [11]. It can be connected to the flight controller's TELEM2 port (TX and RX pins) using a DF13 header. The red and black wires (VCC and GND) were connected with an external power supply and the remaining three wires (blue, white and green) were left unconnected. Figure 22 illustrates the electrical wiring diagram for this configuration.

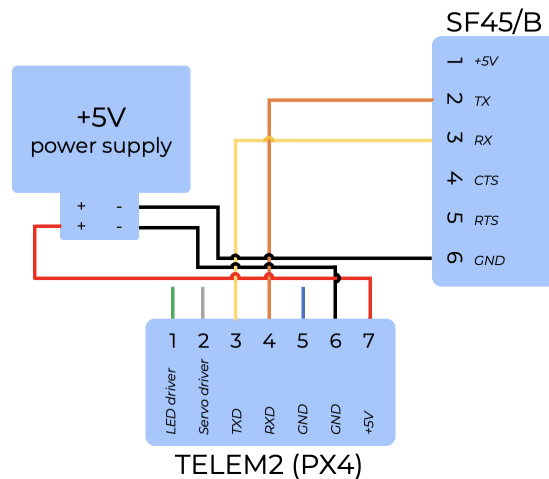


Figure 22: Eletrical wiring diagram for SF45/B.

The LIDAR also comes with a micro USB port that connects to any PC running the LightWare Studio application for visualisation of results, making configuration changes and upgrading the firmware. The sensor should be configured by setting the scanning angle limits to -90° and 90° , the baud rate to 921600 and the update rate to 50 Hz. It is also important to make sure 'scan upon startup' is enabled.

To activate this sensor within the QGroundControl environment, it is necessary to follow the instructions in Sec. 4. After that is assured, the user can access the vehicle setup section and, within the parameters tab, set SENS_EN_SF45_CFG to the desired serial port (TELEM2).

In the bench tests, the angles of the detected obstacle and scanning speed of the sonar were varied. Figure 23 shows the resulting scans from two different experiments: a) the object to detect is in front of the sensor; and b) the object to detect is at 90° in relation to the sensor. In these tests, the forementioned rectangular XPS board (125.5x60cm) was used as target.

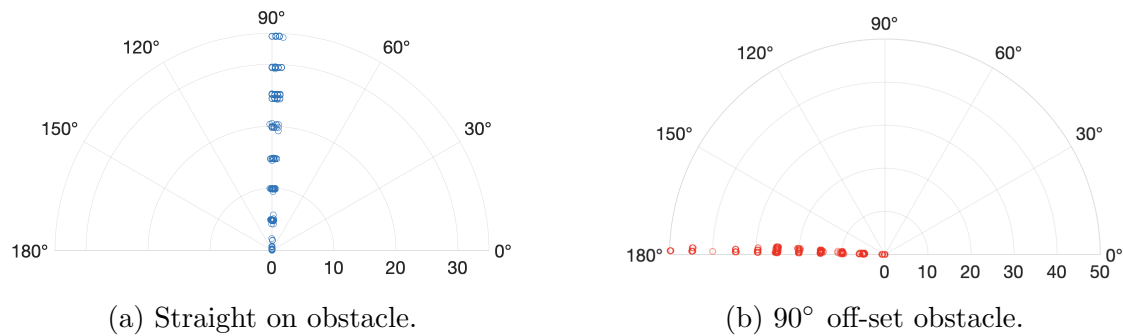


Figure 23: LIDAR bench tests.

On an unobstructed rugby field, the XPS board was positioned at different distances relative to the LIDAR (0 to 50m) and the sensor data was recorded for 30 seconds for each position. As expected, this sensor performed better than the others, maintaining a perfect detection rate through all its range in both experiments, as seen in Fig. 24. However, Fig. 25 shows that the average absolute error was overall lower when the obstacle was aligned with the sensor. This is likely because the LIDAR scans back and forth from -90 to 90° , meaning that for each sweep, it passes twice through $\theta = 0^\circ$ and only once through each limit.

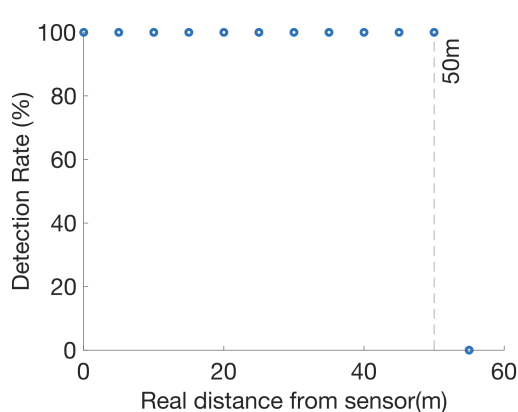


Figure 24: SF45/B detection rate.

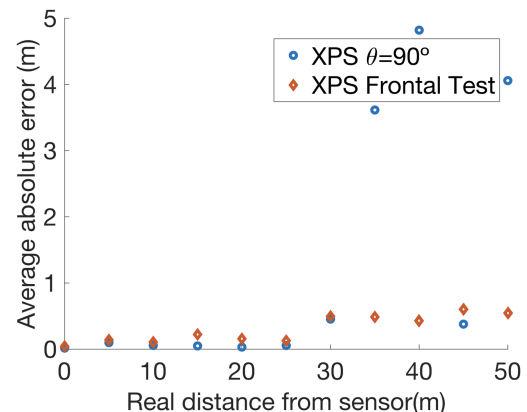


Figure 25: SF45/B average absolute error.

Although the SF45's update rate was set to 50Hz, empirically, it is, on average, 37.2Hz. In LightwareStudio, it is also possible to calibrate the sensor's cycle delay, which is inversely proportional to its scanning speed. The minimum cycle delay (5) corresponds to the maximum scanning speed (6.3 rad/s) and vice-versa. This implies that, by choosing a higher sweep speed and maintaining the angle limits, the arc of circle that is not being detected between each measurement increases. Figure 26 illustrates how the length of the arc traversed varies analytically with the distance to the sensor and the angular velocity. This graphic shows that, although the LIDAR has a 50m range, at the maximum scanning speed, it might not be possible to detect an obstacle less than 8m wide at this distance. When the scanning speed is reduced, the sensor is likely to detect a target of at least 2.2 m at maximum distance. At minimum speed, this stops being relevant within the 50 m range. However, if covering a larger area quickly is more important, sacrificing some visibility at the maximum range might be acceptable. Ultimately, the compromise should be based on the specific needs and constraints of the system.

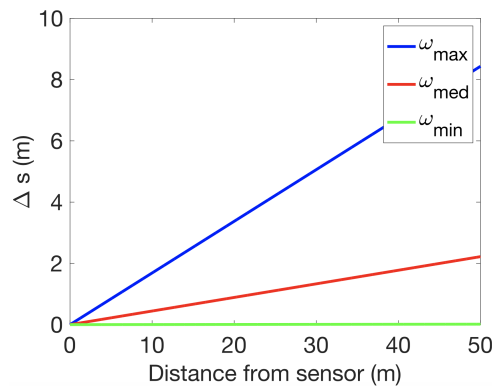


Figure 26: SF45/B undetectable arcs for different scanning speeds.

6 CONCLUSIONS

This work presents a comprehensive solution for enhancing the safety of small fixed-wing UAVs by addressing the critical issue of obstacle detection during flight. A set of select sensors, namely the ultrasonic sensor, laser rangefinder, LIDAR, and RADAR, were identified and further employed in modeling collision detection and avoidance simulations using the potential fields method.

To determine the best combination of sensors and their orientations, these simulations were used in an optimization study. The study revealed that relatively simple detection configurations can yield a high success rate in collision avoidance. While the ultrasonic sensor is found to be inadequate due to its limited range, the laser rangefinder benefits from a long range, but has a restricted field-of-view. On the other hand, both the LIDAR and RADAR prove to be the most promising options, offering not only a substantial range but also a wide field-of-view. Based on the optimization study, the recommended multi-sensor configurations consist of a front-facing LIDAR or RADAR, accompanied by a pair of laser rangefinders pointing sideways at either a 10 or 63 °angle.

To validate the proposed system, the necessary hardware and software were successfully implemented, which allowed for the individual testing of each sensor. The bench tests confirmed the accuracy of the sensors specifications and previous simulations. In the case

of the ultrasonic sensor, the importance of the material and the angular deflection of the obstacle to be detected was highlighted. As for the laser rangefinder, the key factor proved to be directionality. The LIDAR presented less shortcomings, as expected. However, the sensor's parameters (update rate, angular velocity and scan angle limits) directly affected its performance. More specifically, it is necessary to reach a compromise between the LIDAR scan speed and the effective range of visibility.

In the future, these sensors will be integrated into a multi-sensor configuration to be tested on a rover. Subsequently, the system's performance will be evaluated under realistic conditions through flight testing. Overall, this work provides a comprehensive methodology for testing and validation of an optimized multi-sensor system configuration and the proposed system holds great potential for enhancing the safety of small fixed-wing UAVs during flight.

ACKNOWLEDGEMENTS

This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] S. A. H. Mohsan, M. A. Khan, F. Noor, I. Ullah, and M. H. Alsharif. Towards the Unmanned Aerial Vehicles (UAVs): A Comprehensive Review. *Drones*, 6(6), 2022. doi:10.3390/drones6060147.
- [2] Commercial UAV Market Share, Size, Trends & Industry Analysis Report By Type; By End-Use; By Region; Segment Forecast, 2021 - 2028. *Polaris Market Research*, 2021. Accessed: 05/10/2022.
- [3] J. N. Yasin, M.-H. Haghbayan, M. M. Yasin, and J. Plosila. Swarm formation morphing for congestion-aware collision avoidance. *Heliyon*, 7(8), August 2021. doi:10.1016/j.heliyon.2021.e07840.
- [4] Next Generation Air Transportation System (NextGen). <https://www.faa.gov/nextgen>. Accessed: 10/10/2022.
- [5] Title 14 Code of Federal Regulations (CFR) Part 91.113 and RTCA. <https://www.govinfo.gov/content/pkg/CFR-2007-title14-vol1/html/CFR-2007-title14-vol1.htm>, 2007. Accessed: 05/10/2022.
- [6] N. Alturas. Modeling and Optimization of an Obstacle Detection System for Small UAV's. Master's thesis, Instituto Superior Técnico, Lisboa, Portugal, January 2021.
- [7] P. Serrano. Optimization of Obstacle Detection for Small UAVs. Master's thesis, Instituto Superior Técnico, Lisboa, Portugal, June 2022.
- [8] N. Alturas and A. Marta. Modeling and Optimization of an Obstacle Detection System for Small Fixed-wing UAV. In *Aerobest 2021 - ECCOMAS Thematic Conference on Multidisciplinary Design Optimization of Aerospace Systems*, Lisboa, Portugal, July 2021. ISBN:978-989-99424-8-6.

- [9] MaxBotix I2CXL-MaxSonar-EZ Datasheet. <https://maxbotix.com/pages/i2cxl-maxsonar-ez-datasheet>. Accessed: 13/04/2023.
- [10] LightWare LW20 LiDAR sensor Datasheet. <https://www.documents.lightware.co.za/LW20%20-%20LiDAR%20Manual%20-%20Rev%2012.pdf>. Accessed: 13/04/2023.
- [11] LightWare SF45/B product guide. <https://support.lightware.co.za/sf45b/#/specs>. Accessed: 13/04/2023.
- [12] Ainstein US-D1 Data Sheet. <https://ainstein.ai/wp-content/uploads/US-D1-Data-Sheet.pdf>. Accessed: 13/04/2023.
- [13] L. Yang, X. Feng, J. Zhang, and X. Shu. Multi-ray modeling of ultrasonic sensors and application for micro-UAV localization in indoor environments. *Sensors*, 19(8): 1770, 2019. doi:10.3390/s19081770.
- [14] M. Schirrmann, A. Hamdorf, A. Giebel, F. Gleiniger, M. Pflanz, and K.-H. Dammer. Regression kriging for improving crop height models fusing ultra-sonic sensing with UAV imagery. *Remote Sensing*, 9(7):665, 2017. doi:10.3390/rs9070665.
- [15] J. Saunders, B. Call, A. Curtis, R. Beard, and T. McLain. Static and dynamic obstacle avoidance in miniature air vehicles. In *Infotech@ Aerospace*, page 6950. Provo, UT, USA, September 2005.
- [16] F. Fayad and V. Cherfaoui. Tracking objects using a laser scanner in driving situation based on modeling target shape. *2007 IEEE Intelligent Vehicles Symposium*, pages 44–49, 2007. doi:10.1109/IVS.2007.4290089.
- [17] M. S. Grewal, A. P. Andrews, and C. G. Bartone. *Kalman filtering*. Wiley Telecom, second edition, 2020. ISBN:0-471-26638-8.
- [18] M. Skowron, W. Chmielowiec, K. Glowacka, M. Krupa, and A. Srebro. Sense and avoid for small unmanned aircraft systems: Research on methods and best practices. *Journal of Aerospace Engineering*, 233(16), June 2019. doi:10.1177/0954410019867802.
- [19] M. Longbin, S. Xiaoquan, Z. Yiyu, S. Z. Kang, and Y. Bar-Shalom. Unbiased converted measurements for tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):1023–1027, 1998. doi:10.1109/7.705921.
- [20] N. Gageik, P. Benz, and S. Montenegro. Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors. *IEEE Access*, 3:599–609, 2015. doi:10.1109/ACCESS.2015.2432455.
- [21] PX4 Autopilot User Guide. <https://docs.px4.io/main/en/>, . Accessed: 27/10/2022.
- [22] PX4 Building Software. https://docs.px4.io/main/en/dev_setup/building_px4.html, . Accessed: 20/05/2023.

- [23] PX4 Board Configuration (Kconfig). https://docs.px4.io/main/en/hardware/porting_guide_config.html, . Accessed: 20/05/2023.
- [24] QGroundControl User Guide. <https://docs.qgroundcontrol.com/master/en/index.html>. Accessed: 25/01/2023.
- [25] PX4 Loading Firmware. <https://docs.px4.io/main/en/config/firmware.html>, . Accessed: 20/05/2023.



SOFTWARE SUBSYSTEM AS A NEW CONCEPT IN SATELLITE SYSTEM ARCHITECTURE

João P. L. Monteiro^{1*}, Paulo J. S. Gil¹ and Rui M. Rocha²

1: IDMEC
Instituto Superior Técnico
University of Lisbon
Av. Rovisco Pais 1
{joaoplmonteiro,paulo.gil}@tecnico.ulisboa.pt

2: Instituto de Telecomunicações
Instituto Superior Técnico
University of Lisbon
Av. Rovisco Pais 1
rui.rocha@tecnico.ulisboa.pt

Abstract. *Satellite systems increasingly rely on software in order to deliver the intended functionality. While the reliance on software has enabled the miniaturization of space systems, as well as improved performance in various technologies such as image processing or telecommunications, it has also brought up some issues in reconciling hardware and software architecture. As an approach for better integration of hardware and software architectures in modern space systems, we argue that some software modules should be considered independent at the first level of the hierarchical system decomposition. We argue that such software modules are not contained within a more abstract entity featuring hardware and software, which we would typically call a "subsystem". Rather, it is a "software subsystem" which can be developed separately from the processing platform including the devices (sensors, actuators, external memory, etc.) and the processor on which it is deployed, and can be deployed to one or more processing platforms whose hardware, firmware and other software components may be developed by separate contractors. We demonstrate the feasibility of this approach through a case study using a software-intensive university CubeSat. We show that this approach allowed for extensive code reuse, simplified the implementation of subsystem interfaces, and enabled better engineering solutions through increased specialization during the design phase. We also discuss some limitations of the approach, such as increased software overhead and integration issues with off-the-shelf subsystems. We conclude that the concept of abstracted software subsystems can contribute to better satellite system performance by allowing further software specialization in an industry which has traditionally been conservative in this domain.*

Keywords: Satellite, Software, Architecture

1 INTRODUCTION

Spacecraft are complex systems comprising multiple components, or subsystems, which are typically developed by specialized teams from different engineering disciplines. As all complex systems, the behaviours of spacecraft emerge from the interactions between subsystems, which means that the overall system performance is unlikely to be satisfactory if each subsystem is optimized according to domain-specific rules. The development of spacecraft architecture, that is, the definition of which subsystems are used, how they interact, and their individual performances, must involve multidisciplinary optimization if a globally acceptable performance level is to be achieved. The increasing commercialization of the space industry poses a challenge for multidisciplinary optimization by imposing new objectives in the form of non-functional requirements (also called the "ilities"), as does the increasing reliance on software for delivering functionality, which in effect introduces a host of new performance parameters which were hitherto not considered. The challenge in integrating non-functional and software properties in multidisciplinary optimization is compounded by the fact that both issues can be difficult to assess quantitatively.

The quantitative assessment of non-functional properties of systems has often been performed in the systems engineering and engineering design literatures through modularity analysis. Modularity is a measure of subsystem independence, which is linked to properties such as reliability, maintainability, flexibility, etc. The space industry has evolved toward very modular architectures, with most spacecraft featuring some sort of on-board computer for command and data handling, a telecommunications subsystem for transforming digital information to and from radio frequency signals, an attitude control system and a propulsion system for changing the spacecraft's position and orientation towards Earth, and a power handling system for harvesting energy from solar panels, storing it, and distributing it to the other subsystems. As software is increasingly employed for previously hardware-based functionalities, software-specific functionalities (such as digital interfacing) must be replicated across different physical subsystems. This can increase the overall project cost due to the duplication of effort as well as from the increasing complexity of the digital interfaces, particularly when bus-type communications systems (which involve appropriate interfacing between all subsystems) are used. This suggests that the traditional spacecraft system decomposition may no longer ensure adequate modularity.

In this paper, we propose an alternative spacecraft system decomposition, featuring a Software Subsystem (SWSS) as a generic module which has no unique physical implementation, instead being deployed to other physical subsystems but being independently developed. We argue that the application of this concept could lead to advantages such as reduction of overall development effort, ease of system integration (leading, in turn, to early system-level testing), and improved performance stemming from specialization. We describe the concept of a SWSS in Section 3. We then demonstrate the feasibility and desirability of the concept through a case study with a real spacecraft in Section 4. We discuss the implications of the concept for multidisciplinary optimization of spacecraft architectures and conclude the article in Section 6. In the next section, we present some related work on the topic of modularity assessment, its relationship to non-functional properties, and the difficulty in integrating the analysis of hardware and software systems.

2 RELATED WORK

Modularity is a key parameter in evaluating non-functional properties of complex systems, and can be quantitatively measured. Modularity can be defined as the extent to which each system component, or subsystem, can be developed separately from the others. For this to occur, the interactions between subsystem components (from which its functionality emerges) must be significantly stronger than the interactions between components belonging to different subsystems. Modularity analyses are useful for comparing different architectural designs according to how "clustered" their physical components are. These analyses are frequently complemented with the use of algorithms to change a pre-defined component grouping and arrive at an optimal modular architecture [1–3]. New developments in the application of clustering algorithms to system architectures include adding design constraints to the problem such as specifying that two components must be kept apart [4] and accounting for different preferences in system decomposition [5], an issue which arises from the fact that different stakeholders identify different relationships between components in the same system [6, 7].

Multiple metrics have been proposed for modularity analyses, many of which rely on the adjacency matrix of a graph representation of a system to assess the coupling between modules [8]. The adjacency matrix of a graph is a square matrix where the $A_{ij} = 1$ if there is an edge from node i to node j , and 0 if there is no relationship. Graph based analysis allows abstracting the detail of each component in a system and understanding characteristics of the system's structure. Practitioners employ various metrics, usually based on a ratio between intra-module and inter-module interactions, and discussions of modularity analysis often address the adequateness of these metrics in different situations.

There are notable differences in the software and systems engineering literature regarding research on modularity, each with their own interpretation of what is a component and what constitutes an interaction. The software literature focuses modularity assessments on the static structure of the source code repository, with nodes representing folders, files, or functions [9]. Interactions between nodes often represent function calls or file inclusion, or how often files are committed at the same time [10]. The systems engineering literature employs more diverse interpretations for what constitutes a node, either focusing on the physical components or on an abstract functional architecture [11]. Interactions between nodes can represent physical or geometrical connection, information flow, or estimates of "design parameter influence".

While the different approaches enrich the discussion by presenting different views of what constitutes modularity, they make it difficult to compare results. Because there is no overlap between software and hardware approaches, it is difficult to assess modularity in space systems for which functionality depends on both. Some research (from the software literature) also suggests that the abstraction of the hardware architecture can lead to "modularity violations", where engineering changes are required of two components which were theoretically not linked but which share a hardware interface [12]. This presents a research opportunity to study the impact of assessing modularity for an integrated architecture rather than for separate hardware/software architectures.

It is therefore difficult to assess modularity in modern spacecraft, which increasingly derive their capabilities from software. There are also challenges involved in reconciling hardware and software architecture practices [13], making it harder to understand how some sort of "interface" could be developed to "bridge" the gap in modularity assessment.

The remainder of this article proposes the "software subsystem" concept as a possible approach to bridging this gap.

3 SOFTWARE SUBSYSTEMS: DEFINING THE CONCEPT

In this section, we propose a new framework for spacecraft systems architecture which enhances the role of software as a driver of functionality for space missions, describe some opportunities this framework offers, and discuss required changes to development processes to enable its implementation.

We argue that, in modern space systems, some software modules should be considered independent at the first level of the hierarchical system decomposition. By software module, we mean a collection of source code files which, when compiled and deployed on a processing unit, make the hardware behave in a well defined and predictable way. We use the plural "modules" in the sense that there can be multiple collections of source code files, deployed to one or more processing units, each of which independent. What makes a module independent is that it can be developed concurrently with other modules, provided that the interfaces to the processing units and to other software modules, as well as requirements, are defined. The final aspect of our claim, that some software modules can be considered independent at the first level of system decomposition, entails that a software module is not contained within a subsystem. Rather, the module can be developed separately from the processing platform including the devices (sensors, actuators, external memory, etc.) and the processor on which it is deployed, and can be deployed to one or more processing platforms whose hardware, firmware and other software components may be developed by separate contractors. We call such an independent software module a "software subsystem", as illustrated in Figure 1.

To be independent as described above, a software subsystem must retain its configuration when deployed, by which we mean that the collection of source files is not changed regardless of which processing platform it is being compiled for. This is similar to the integration of hardware subsystems, in that the subsystem component parts and their interrelationships do not change when the subsystem is integrated. However, the interaction between hardware and software depends on the specific microprocessor device, as different units have different physical architectures, different interface capabilities, and different ways of making the stored program data interact with the I/O configurations. This mismatch between the specific operation of a microprocessor unit and the abstract nature of a software subsystem must be solved through the implementation of a hardware abstraction layer (HAL) by the team in charge of developing the processing platform. The implementation of a HAL is a typical practice in embedded systems programming, allowing the separation of the higher-level application code from the details of the microprocessor operation. Within the framework of our proposal, the HAL constitutes not only an internal interface defined by the subsystem developer, but also a system-level interface managed by the prime contractor.

In addition to being abstracted from the operational details of the hardware on which it will be deployed, the software subsystem must be abstracted from the specific software functionalities which are not shared across subsystems. Typical spacecraft subsystems, such as attitude control or power distribution units, rely on embedded software in order to deliver their core functionality. This functionality is specific to those subsystems and therefore not implemented in others unless there are redundancy concerns. The focus of

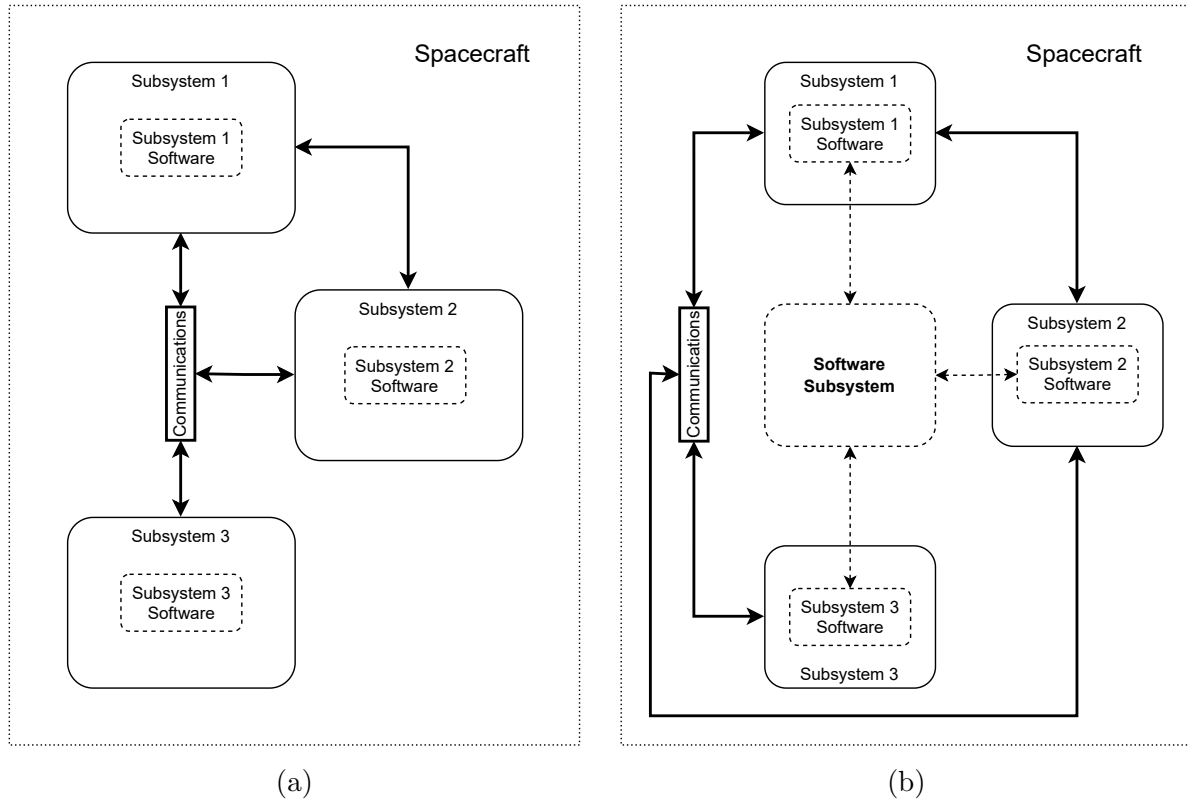


Figure 1: Comparison between a traditional spacecraft architecture (a) and the proposed architecture featuring a Software Subsystem (b).

these software components is often domain specific and deeply intertwined with knowledge of the physical processes they act upon. Accommodating these capabilities in the software subsystem itself would make it more complex and would negate the benefits of modularity. Instead, the software subsystem must be viewed as a service which is available to enhance the capabilities of the subsystem and which is useful for multiple subsystems. Together with the previous point regarding the need for a HAL, this entails that hardware subsystem developers implement both the application-level software which is specific for each subsystem and the low-level software which determines how the microprocessor interacts with other hardware devices.

4 CASE STUDY: ISTSAT-1

In this section, we describe how a software subsystem has been successfully implemented in a real satellite system and compare this system to a traditional development approach. The ISTSat-1 CubeSat [14] is the subject of this case study. We consider that the spacecraft itself, as well as the development approach which lead to its final flight qualification, adequately embody many of the new features of the changing space industry which we discussed in Section 1. The spacecraft was developed using a low-cost, high-risk approach with very limited use of radiation-hardened components and a heavy reliance on off-the-shelf electronics with no flight heritage. An iterative development approach was followed, allowing for continuous refinement of requirements as they were better understood by end users and developers. A lot of effort was put into ensuring that the system is modular (as is typical with CubeSats) and, thereby, partially reusable.

Finally, system capabilities are mostly implemented in software, which was developed as abstractly as possible in order to allow for code reuse.

4.1 ISTSat-1: a traditional system decomposition

The ISTsat-1 spacecraft can be decomposed into the ADS-B payload module and the flight platform. The flight platform carries the technology needed to ensure the reliability and maintainability of the spacecraft during the mission lifetime and to provide the required logistics for the payload module. Because the structural part contributes little to the functionality of the system, we will not discuss it in this work. There are seven different avionics systems, five of which were developed in-house and two of which are commercially available products. The Antenna Deployment Mechanism (ADM) is a commercially available product which is responsible for deploying the VHF/UHF dipole antennas (which are too long to fit unfolded within the 1U CubeSat envelope) upon command. The Solar Panels (SP) are also commercially available products which feature two triple-junction photovoltaic cells each, generating electrical power when in sunlight, in addition to hosting coarse sun sensors (which indicate how the panel is angled towards the sun) and embedded coils which can be used as magnetic torquers.

The following paragraphs provide a small description of the in-house developed spacecraft subsystems. For architectural purposes, the ADM can be considered a component of the Telemetry, Tracking, and Command (TTC) subsystem, whereas the SP can be considered part of the Electrical Power Supply (EPS) subsystem. The global architecture is depicted in Figure 2.

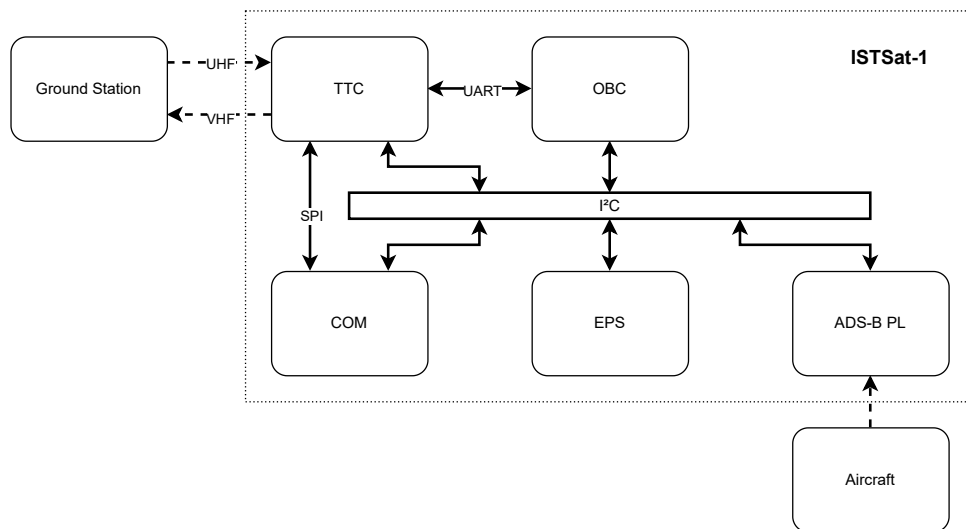


Figure 2: ISTSat-1 architecture. The thick lines represent digital interfaces.

The Payload (PL) subsystem detects aircraft messages encoded and broadcast using the ADS-B standard. When a valid message preamble is detected, it decodes the message and compares the contents against data filters set by the operator, thus allowing only valid messages (for instance, only messages containing information about aircraft position and speed) to be stored. It does not store the message data internally. Rather, it uses the I²C bus, with which it interfaces with all subsystems, to send the data to the COM subsystem, where it is stored. The decoding algorithm is based on a C language implementation of the

dump1090 algorithm originally developed in Matlab. This allowed deploying the algorithm on a ARM Cortex-M4 processor running a real-time operating system rather than a dedicated FPGA board, leading to lower power implementation - which was required due to the limited power generation capabilities of a 1U CubeSat platform.

The On-Board Computer (OBC) implements most of the fault detection, isolation and recovery functionality in the spacecraft, as well as the operational mode management. As the system mode and state manager, it is the OBC which determines which subsystems should be on and off. It is the OBC which periodically requests health information (for instance, temperature data) from all other subsystems and either stores this data or sends it to the TTC to be broadcast. It also estimates where the spacecraft is pointing (relative to the Earth) and drives the magnetic torquers embedded on the SP to create a magnetic dipole which interacts with the Earth's magnetic field, changing the spacecraft attitude. Lastly, it implements part of the digital communications protocol stack, which comprises the logical interface between the spacecraft and the mission control ground station. This means that ground commands sent to any subsystem are "dropped" on the I²C bus by the OBC. It implements an optional reliability layer which is useful when users wish to download stored data. Despite the wide scope of the OBC's functionalities, one of the main requirements guiding its development was that the subsystem should have a very low power consumption, leading to low computational resources.

The Communications Processor (COM) subsystem is a more computationally powerful processing platform which enhances some of the OBC's functionalities, at the cost of increased power consumption. It features an alternative implementation of the communications protocol stack which allows for the use of a higher bitrate digital modulation scheme than that allowed by the OBC. It also significantly expands the spacecraft's data storage capacity, allowing the OBC to forward the housekeeping telemetry data to its memory banks. All ADS-B message data is stored in the COM subsystem, as are the configurable parameters of the PL. As is the case with the OBC, the COM subsystem can create a reliable communications protocol with the ground station to ensure that no data is missing (e.g. due to packet losses) when stored data is downloaded. The COM subsystem, like the PL, is powered off while the spacecraft is in Safe Mode.

The TTC subsystem is responsible for maintaining the radio link with the control ground station, as well as broadcasting digitally modulated packets featuring telemetry data which can be interpreted by any radio amateur operator with the decoding software. It features a modem which handles the modulation function, transforming the digital data stream to be transmitted to the ground station into a baseband analog stream that drives a transceiver and is delivered to the antenna. Conversely, the modem performs the demodulation of the incoming analog signals to a data stream to be delivered to other subsystems. When interacting with the OBC, i.e., when using the lower bitrate modulation scheme, the TTC implements the lowest level of the communications protocol stack. When interacting with the COM, the whole protocol stack is implemented in the COM. The TTC directly interfaces with the ADM, to which it sends the antenna deployment command once the spacecraft is ready (45 minutes after deployment from the launcher). Lastly, the TTC features a Beacon module, a completely independent transmission system with its own modulator and power amplifier, which allows sending short amounts of information encoded in Morse.

The last subsystem, the Electrical Power Supply (EPS), implements all of the power

handling and distribution functionalities of the spacecraft. It implements a Maximum Power Point Tracking (MPPT) algorithm to optimize power draw from the illuminated solar panels, and distributes it to other subsystems as well as to a lithium-polymer battery. When no solar panel is illuminated, it distributes the energy stored in the battery. It receives commands from the OBC through the I²C bus in order to know which modules to power according to the spacecraft operational mode, and it also uses the I²C bus to receive commands from ground (for instance, to reboot a certain subsystem by cutting a specific power regulator). It implements a safety mechanism whereby it reboots a subsystem if the average power consumption suddenly increases above a certain threshold. It also implements a small monitoring function by requesting a keep-alive signal from the OBC. Lastly, in order to ensure safe operation even in very cold environments which can be expected while the spacecraft is in eclipse, the EPS monitors and controls the battery temperature using resistive heaters.

4.2 ISTSat-1 functional chains

The exchange of information between the processing units of the various physical subsystems is a critical aspect of delivering the overall system functionality. For the most part, this exchange of information relies on a low-level digital communications interface such as I²C or SPI to encode data into electrical signals and vice-versa, and some software component, running in each processing unit, which attributes meaning to the received bit frames and dispatches some sort of action within the processor. The following paragraphs provide some examples of how sequences of message exchanges between the processing units are used to deliver functionality. Although there are many different ways of conceptually organizing these examples, we have chosen to broadly separate them between those which are triggered externally (through control ground station commands) and those triggered internally by some sort of timer.

Many spacecraft functionalities involve replying to a command sent from the control ground station. The most basic operations involve issuing a command to one of the physical subsystems and receiving a data frame containing information. One example is issuing a "ping" command, which returns a short acknowledgement message. Other examples include requesting the value of an internal variable (for instance, the value read by a temperature sensor in the OBC board) or requesting a change in the value of a configurable parameters (to which the physical subsystem would reply by confirming that the value had been changed). All the physical subsystems support data requests and configuration changes. The same mechanism is also used for deploying actions which are specific for each physical subsystem. For instance, one could issue a command directly to the EPS to turn the battery heaters on. The operator would not be able to check that the heaters are indeed on without issuing a data request (for instance, checking that the battery current drain had increased), but would receive an acknowledgement that the command has been well interpreted.

In all of these scenarios, the basic exchange mechanism is the same, regardless of the physical subsystem involved. The chain of message exchanges involved in a "ping" command to the EPS is depicted in Figure 3. First, the radio frequency signal sent by the ground station is converted into a packet by the TTC (#1). The TTC then sends the message to the OBC using the UART interface (#2), which "drops" the message on the I²C bus. Note that, at this point, no processing unit "knows" what the command is:

they are just relaying the message. Also, during this process, neither the TTC nor the OBC expect any reply once their part of the message exchange is complete. Finally, the target subsystem interprets the message, performs the intended action, and replies using the reverse procedure (#4,#5,#6).

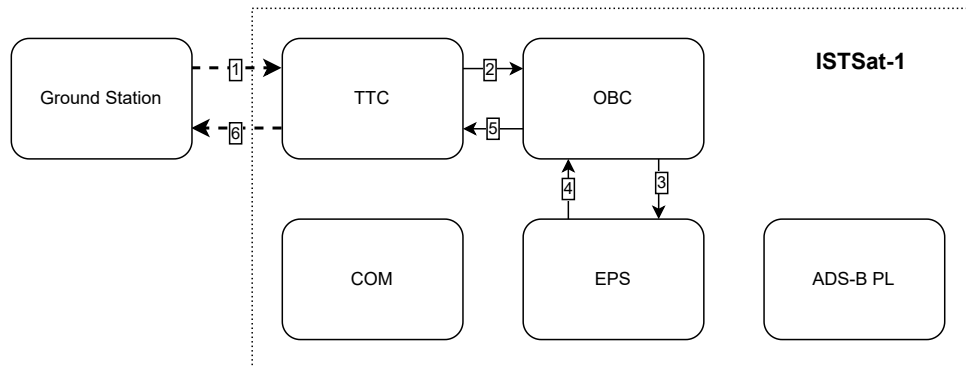


Figure 3: Functional chain involved in "pinging" the EPS subsystem.

4.3 An alternative decomposition of ISTSat-1: the HouseKeeping Subsystem

The interactions described above involve multiple networking functionalities. First, the sender of a command must be able to encode information in such a way that the receiver can decode it and trigger the appropriate response. This is done by establishing a communications protocol which defines the different fields included in a message (i.e. what information is encoded in each group of bits), and by defining a list of available commands and their opcodes for each subsystem. Second, the sender should be capable of handling instances where the receiver does not respond (e.g. by resending the command), and the receiver should handle errors (such as wrong opcodes, possibly stemming from corrupted messages) and handling new commands while a previous one is still under execution. Additional functionalities which are needed to execute only some of the scenarios described above include ensuring a reliable stream of ordered data, as well as securing some commands such that they can only be executed by an authenticated user. While these functionalities could be implemented separately, the ISTSat-1 team decided to define a single codebase which is compiled to each physical subsystem and which interfaces with its application-level and driver-level software. This new software subsystem is the HouseKeeping System (HKS), as depicted in Figure 4.

The HKS implements the functions which each physical subsystem requires in order to logically interface with one another. This means that whenever one subsystem must issue a command to another (for instance, the OBC requesting telemetry data from the EPS), the application-level routine in the "sender" subsystem will call a HKS function to issue a command. This function builds the message according to the type of command and the command parameters to be sent. The HKS then calls the subsystem-specific driver-level software to physically send the command (e.g. using the I²C bus). When the "receiver" subsystem gets the command, it is the driver-level software which calls an HKS function to interpret the command and to dispatch it. The HKS does this by calling a function which is implemented on the application-level subsystem-specific software. The interface between the HKS and the subsystem-specific software is defined as a list of functions, both at application level and driver level, which each subsystem must implement. The function

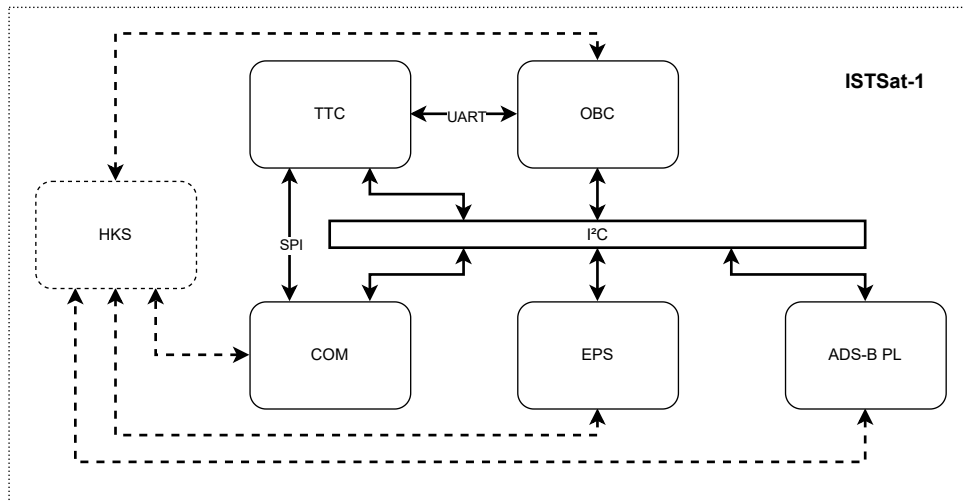


Figure 4: An alternative decomposition for ISTSat-1, in which the HKS provides logical interfacing functionalities which are traditionally implemented separately in each physical subsystem.

arguments are then used in order to differentiate between different processing units and between commands to be executed. The complete list of commands and arguments is provided in a configuration file which is, at all times, accessible to all developers, and constitutes a single source of truth for this information (i.e. once a new command is implemented in a subsystem, others can issue this command just by calling the HKS function). The HKS is compiled to the OBC, EPS, COM, and ADS-B PL from the same codebase, which is placed in the project repository in a shared folder.

5 DISCUSSION

Based on the description provided in Section 4, we consider the HKS to comply with the specification for a Software Subsystem as defined in Section 3. First, all physical subsystems (except the TTC, for which the networking functionalities were implemented separately) rely on the same codebase for compilation of the HKS. The HKS is also independent both from the domain-specific application software and from the low-level hardware interface software for each physical subsystem.

The development of the HKS as a software subsystem allowed engineers to focus on their area of expertise, such as power electronics, attitude determination and control, embedded systems, digital signal processing, etc. This was particularly important for the ISTSat-1 project, for which each subsystem was developed as part of students' MSc dissertation work. Because the networking aspects themselves constitute an engineering speciality, this system decomposition allowed all students to develop technology which was relevant for their specific engineering degrees while spending less time in development activities which would not be as relevant for their evaluation. This "efficiency" is compounded by the fact that because networking is not their area of expertise, they would have spent a disproportionate amount of time implementing that functionality. As a result, and also given the time-bounded nature of MSc dissertation work, it seems reasonable to argue that the overall quality of the system improved due to this system architecture.

In addition to perceived improvements in system quality, the HKS subsystem allowed for early integration of system-level prototypes. In a traditional decomposition, the de-

development of interfacing functions is typically a lesser concern in comparison to the development of highly specialized, domain-specific capabilities. As such, subsystems are seldom ready for preliminary integration before the domain-specific capabilities are fully developed, despite the fact that interactions between subsystems are often the cause of system-level issues. By developing the HKS subsystem concurrently with the domain-specific functions, the team were able to integrate preliminary prototypes of the physical subsystems, which uncovered several integration issues which would have remained unresolved until final system integration. Because of this, the team were also able to perform multiple system-level tests (particularly those which stressed the interactions between subsystems) before all domain-specific capabilities were fully implemented [15], which in turn uncovered more design flaws when they could still be corrected with relatively low cost.

A final advantage of implementing the HKS was the improved flexibility stemming from the easy expansion of command lists and the single configuration file allowing all subsystems to "know" which commands and data variables were available from other subsystems. This allowed for a more agile approach to requirements and interfaces definition, which proved to be crucial for a relatively inexperienced team - as demonstrated by the significantly different command and variable lists at the beginning and at the end of the project. The flexibility of the HKS also allowed expanding on some of its core functions to deliver new capabilities which were not strictly considered at the beginning of the project, such as the capability to store information in non-volatile memory which is available in a different subsystem.

The implementation of the HKS also brought some challenges. The most obvious one is the overhead stemming from the generic nature of the software subsystem, which led to some memory and CPU issues in the OBC, which features a very low power microprocessor. A less obvious one is the difficulty in integrating physical subsystems which do not share the same interface. This was the case with the TTC, for which the networking and digital interfacing capabilities were separately implemented. During development, it was assumed that all processing units would run the same version of the HKS and, therefore, would always be fully compatible with each other in terms of networking. Manually updating the TTC interface to keep up with changes in the other subsystems was demanding and proved to be a source of some issues during integration testing.

Further work is necessary in order to fully validate this concept, as the educational setting of the ISTSat-1 may not be an accurate proxy for the industrial setting in which the concept would be used. Developing the HKS depended greatly on the open communication between the networking and domain-specific engineering teams, which might be harder to ensure when proprietary information is involved. The scope of the engineering work behind each subsystem for ISTSat-1 is also more limited than that for larger spacecraft, in which large teams may be involved in each subsystem. As teams grow larger, the specialization "penalty" stemming from the incorporation of different domains (for instance, networking and attitude control) in a single subsystem may be smaller than in a one or two person team.

6 CONCLUSION

In this paper, we have presented the concept of a software subsystem and demonstrated its feasibility in spacecraft system decomposition by providing the example of the ISTSat-

1 CubeSat. We believe that this decomposition is different, but potentially beneficial for spacecraft projects in that it allows for further specialization of software practices, enhances concurrent development of subsystem capabilities, and allows for early prototype integration, thereby introducing some "agility" into a traditionally sequential project cycle. This could allow for faster development cycles and seems to be a good approach in reconciling hardware and software architecture practices in the space sector.

This work contributes to the literature on the application of multidisciplinary design optimization to spacecraft design by providing a different decomposition approach which could be used to inform modularity studies. Given that modularity is a key predictor of non-functional properties of systems, and given the importance of these properties in the satisfaction of stakeholder expectations for spacecraft performance in a changing space industry, we argue that modularity is an important parameter to account for in multidisciplinary trade studies. The software subsystem concept can be used to generate alternative system architectures, thereby expanding the design space for optimization algorithms to explore.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the contributions of the IST NanosatLab team throughout the development of the ISTSat-1 mission. The authors particularly thank Alexandre Silva, the software engineer who implemented most of the HKS functionality, and who contributed greatly to the ideas discussed in this article. The authors also thank ESA Education for sponsoring multiple test campaigns under the Fly Your Satellite! Program, as well as their continued technical support. The work of P.G. and J.M. was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020. Lastly, the authors wish to acknowledge the financial support of INESC-ID, Instituto de Telecomunicações, IDMEC, and IST in the development of the ISTSat-1 mission.

REFERENCES

- [1] J. W. Herrmann, M. Morency, A. Anparasan, and E. L. Gralla. Evaluating clustering algorithms for identifying design subproblems. *Journal of Mechanical Design, Transactions of the ASME*, 140(8), Aug 2018. ISSN 10500472. doi:10.1115/1.4040176.
- [2] L. Qiao, M. Efatmaneshnik, M. Ryan, and S. Shoval. Product modular analysis with design structure matrix using a hybrid approach based on mds and clustering. *Journal of Engineering Design*, 28(6):433–456, Jun 2017. ISSN 14661837 09544828. doi:10.1080/09544828.2017.1325858.
- [3] T. L. Yu, A. A. Yassine, and D. E. Goldberg. An information theoretic method for developing modular architectures using genetic algorithms. *Research in Engineering Design*, 18(2):91–109, Aug 2007. ISSN 09349839. doi:10.1007/s00163-007-0030-1.
- [4] K. Sinha, S. Y. Han, and E. S. Suh. Design structure matrix-based modularization approach for complex systems with multiple design constraints. *Systems Engineering*, 23(2):211–220, Mar 2020. ISSN 15206858 10981241. doi:10.1002/sys.21518.
- [5] E. S. Suh, K. Sinha, and J. Ahn. Multi-attribute optimization-based system decomposition considering several value chain stakeholder perspectives. *Research in Engineer-*

- ing Design*, 31(4):411–428, Oct 2020. ISSN 14356066 09349839. doi:10.1007/s00163-020-00342-9.
- [6] J. Ahn, M. Choi, and E. S. Suh. Entropy-based system assessment metric for determining architecture’s robustness to different stakeholder perspectives. *Systems Engineering*, 21(5):476–489, Sep 2018. ISSN 15206858 10981241. doi:10.1002/sys.21448.
- [7] E. S. Suh, N. Chiriac, and K. Hölttä-Otto. Seeing complex system through different lenses: Impact of decomposition perspective on system architecture analysis. *Systems Engineering*, 18(3):229–240, May 2015. ISSN 15206858 10981241. doi:10.1002/sys.21294.
- [8] K. Hölttä-Otto, N. A. Chiriac, D. Lysy, and E. S. Suh. Comparative analysis of coupling modularity metrics. *Journal of Engineering Design*, 23(10-11):790–806, Jan 2012. ISSN 14661837 09544828. doi:10.1080/09544828.2012.701728.
- [9] T. Lutellier, D. Chollak, J. Garcia, L. Tan, D. Rayside, N. Medvidovic, and R. Kroeger. Comparing software architecture recovery techniques using accurate dependencies. In *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015*, volume 2, pages 69–78. IEEE Computer Society, Aug 2015. doi:10.1109/ICSE.2015.136.
- [10] L. L. Silva, M. T. Valente, and M. D. A. Maia. Assessing modularity using co-change clusters. In *13th International Conference on Modularity, MODULARITY 2014 (Formerly AOSD)*, pages 49–60. Association for Computing Machinery, Jan 2014. doi:10.1145/2577080.2577086.
- [11] M. E. Sosa, S. D. Eppinger, and C. M. Rowles. A network approach to define modularity of components in complex products. *Journal of Mechanical Design, Transactions of the ASME*, 129(11):1118–1129, Nov 2007. ISSN 10500472. doi:10.1115/1.2771182.
- [12] L. Xiao, M. J. Pennock, J. L. Cardoso, and X. Wang. A case study on modularity violations in cyber-physical systems. *Systems Engineering*, 23(3):338–349, May 2020. ISSN 15206858 10981241. doi:10.1002/sys.21530.
- [13] M. W. Maier. System and software architecture reconciliation. *Systems Engineering*, 9(2):146–159, 2006. doi:10.1002/sys.20050.
- [14] J. P. Monteiro, A. Cunha, A. Silva, C. Fernandes, D. Neves, F. Naf, G. Tavares, J. Freitas, J. Pinto, M. Piedade, N. Ramos, P. J. S. Gil, P. Macedo, R. Afonso, R. Encarnação, R. Ramos, T. Almeida, and R. M. Rocha. Istsat-1, a space-based automatic dependent surveillance-broadcast demonstration cubesat mission. *International Journal of Satellite Communications and Networking*, 40(4):268–293, Jul 2022. ISSN 15420981 15420973. doi:10.1002/sat.1440.
- [15] J. P. Monteiro, R. M. Rocha, A. Silva, R. Afonso, and N. Ramos. Integration and verification approach of istsat-1 cubesat. *Aerospace*, 6(12), Jan 2019. ISSN 22264310. doi:10.3390/aerospace6120131.



A SYSTEM-BASED APPROACH FOR TECHNOLOGY ROADMAPPING ASSISTED BY VISUAL ANALYTICS

Andrea Spinelli, Gustavo P. Krupa and Timoleon Kipouros

Centre for Propulsion and Thermal Power Engineering
School of Aerospace, Transport and Manufacturing
Cranfield University
MK430AL Cranfield, UK

{andrea.spinelli, g.krupa, t.kipouros}@cranfield.ac.uk, <https://www.cranfield.ac.uk/>

Abstract. *System analysis is necessary to identify and characterise the emerging behaviour of multiple components, or sub-systems, synthesised together to fulfill a new function. Although this is well known, it is hugely important to consider a system-based approach for the exploration and understanding of the behaviour of new technology in existing and mature systems. One example is the introduction of a considerable amount of energy storage in the form of batteries in conventional aircraft configurations. We would like to identify how such technology influences the behaviour of the aircraft but also what is the required level of performance of this technology for any meaningful impact on the environmental targets. We performed such analysis using an open case study of a regional aircraft and open models for the simulation of battery behaviour, the design space exploration, and the interactive visualisation of multidimensional data. The results show that considerable advancements are required in all three key aspects of batteries, energy density, durability, and safety. In this study, we have quantified the required energy density above which meaningful trade-off analyses can be performed and such characteristics provide input to technology roadmaps.*

Keywords: interactive visualisation, technology roadmap, aircraft energy storage, design space exploration, visual analytics

1 INTRODUCTION

The aviation industry is experiencing a significant shift towards sustainable and environmentally friendly solutions [1]. In line with this trend, the development of hybrid-electric aircraft has gained considerable attention and investment. With increasing environmental regulations, volatile fuel costs, and heightened customer expectations for greener transportation, airlines and manufacturers are under pressure to reduce their carbon footprint and mitigate environmental impact [2]. These aircraft hold the potential to reduce emissions, improve fuel efficiency, and contribute to a greener and more cost-effective transportation system [3]. Thus, the market for hybrid-electric aircraft is poised for substantial growth in the coming years with market forecasts indicating a significant demand for these aircraft, driven by the need for more sustainable and economically viable transportation [4].

The development of hybrid-electric aircraft presents complex challenges that require a systems-based approach. Traditional methods of technology development and roadmapping may not fully capture the interdependencies and synergies among different technologies and subsystems. To address this need, a holistic methodology is necessary, taking into account the intricate relationships between technologies, system architecture, market drivers, and performance requirements. Such a systems-based approach enables informed decision-making, efficient resource allocation, and improved technology adoption in the context of hybrid-electric aircraft development. Existing studies have highlighted the importance of integrating systems thinking, modelling techniques, and advanced visualisation tools to enhance technology roadmapping processes [5–9].

To demonstrate the practical application of our system-based approach, we have chosen the development of a 50 passenger hybrid-electric regional aircraft as an illustrative case study. The choice of this aircraft is motivated by the work done in the FUTPRINT50 project [10–12]. This example allows us to delve into the specific challenges and considerations of hybrid-electric technology in the context of regional air travel. By focusing on this example, we aim to provide valuable insights into the technology roadmap development process, identification of gaps, and alignment with strategic drivers for hybrid-electric aircraft in the regional aviation market. Hybrid-electric aircraft combines traditional aviation propulsion systems with electric propulsion, harnessing the benefits of both technologies [13]. By integrating electric motors and batteries, these aircraft can reduce reliance on fossil fuels, minimise environmental impact, and enhance operational efficiency. The development of hybrid-electric aircraft represents a critical step towards achieving sustainable aviation and addressing the growing concerns regarding greenhouse gas emissions and offer a good study case for system-based technology forecasting methods.

The development of hybrid-electric aircraft is influenced by a range of strategic drivers that shape the technological landscape and market dynamics. Understanding these drivers is crucial for effective technology roadmapping and making informed decisions. In Figure 1 the 5 key enabling technologies are shown, as also identified in the FUTPRINT50 roadmap.

The advancement of hybrid-electric aircraft is intrinsically tied to breakthroughs in all these technologies and their interactions. More importantly, the advancements in battery technology are fated to play the biggest role in the development of hybrid-electric aircraft [14, 15], and we use this technology as an example to demonstrate our proposed methodology. However, it should be emphasised that we consider this technology as an



Figure 1: The FUTPRINT50 hybrid electric aircraft architecture and the key enabling technologies.

energy storage means on board the aircraft. Indeed, this can be in the form of batteries, but also in other forms, such as PowerPaste by Fraunhofer IFAM [16].

The seamless integration and widespread adoption of hybrid-electric aircraft by airliners depend on the availability of supporting infrastructure. These include establishing charging or refueling stations, implementing battery swapping or fast-charging technologies, and developing suitable electrical grid systems capable of meeting power demands. The readiness of infrastructure and investments in charging networks greatly influence the feasibility and scalability of hybrid-electric aircraft operations. Therefore, a careful technology roadmapping approach should take these infrastructure considerations into account to ensure the feasibility of hybrid-electric aircraft and their integration into existing air transportation infrastructure.

In the next section, we will introduce our system-based approach for technology roadmapping, leveraging systems modelling and visual analytics to address the gaps in energy storage technology and enable the successful implementation of hybrid-electric aircraft.

2 METHODOLOGY

2.1 Systems-based Design Space Exploration

The design space exploration (DSE) methodology used in this study was developed by Spinelli et al. [17] by combining set-based design (SBD) principles with multi-disciplinary optimisation (MDO).

SBD revolves around three core tenets: reasoning on design intervals (sets) and not individual design points, focusing on eliminating unfeasible designs instead of searching for the optimum, and delaying commitment to a design as much as possible [18]. Implementing these principles allows to explore the effects of the design parameters and its implications on downstream preliminary design phases, hence helping avoid pitfalls.

The advantages of this methodology have been shown in reducing design uncertainty from changing requirements [19] and avoiding cost increments due to rework [20]. It also produces a lot of alternatives, which can be useful in designing a family of products for

the same requirements. However, evaluating many designs in parallel incurs higher computational costs, which may not be feasible for high-fidelity methods. Furthermore, the elimination procedure relies on a method for mapping the requirements to the input parameters. The most common quantitative approaches rely on expert rules or handcrafted knowledge in the form of utility functions [21]. While easier to interpret and implement, these approaches may not be robust when developing novel systems where little experience is available.

The SBD framework developed by Spinelli et al. [17] addresses these limitations by utilising a Bayesian interpretation of the requirements and utilising probability to estimate the likelihood a set is capable of producing a satisfactory solution. This approach avoids explicitly defining rules that map the inputs to the quantities of interest of the system, instead it extracts the knowledge directly from the models utilised in the function evaluations. The procedure shown in Figure 2 follows two general steps: the first one, defined as the exploration phase, divides the input parameters into levels and utilises the probabilistic approach to eliminate those combinations which are likely to not return a feasible solution. The second step, named the search phase, extracts the local design points by introducing a MDO problem in each surviving set. This procedure allows to convert the sets in actual local Pareto fronts, which can be then visualised and analysed for extracting knowledge about the system behaviour.

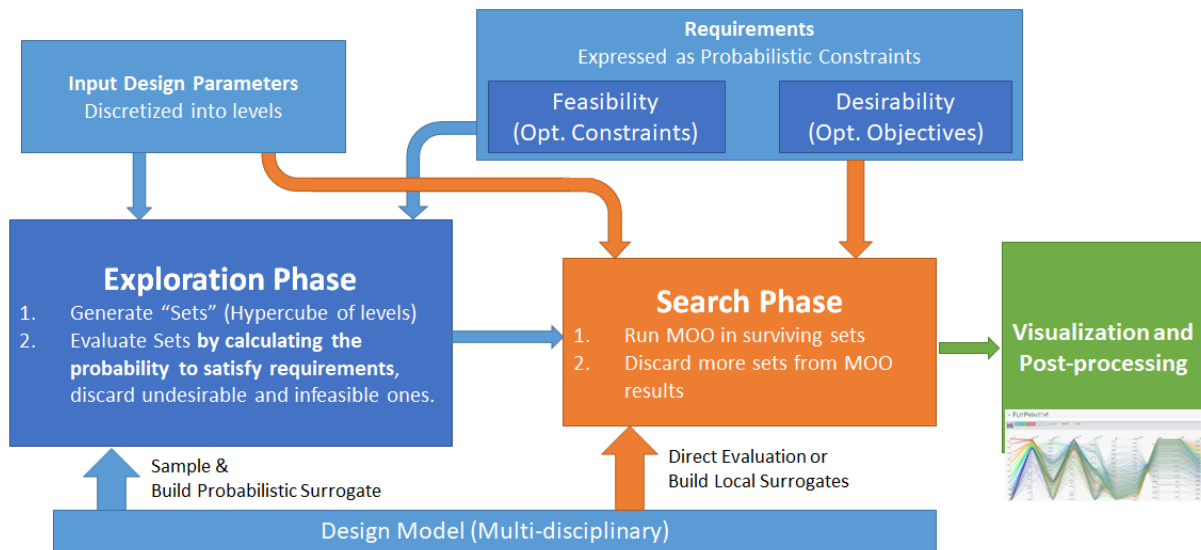


Figure 2: The Set-based Probabilistic Design Optimisation process.

This two step approach enables one to carry out thorough design space exploration without evaluating areas of the design space that would not yield feasible designs. It was shown by the authors that this approach allows to reduce the computational time by 80% [22]. The code is available online in its GitHub repository [23].

By combining the DSE framework described above with a parametric model encompassing the technological scenarios to be explored it is possible to generate data necessary for roadmap definition. The dataset is processed using a visual analytics tool, which allows to highlight information for decision-making purposes.

2.2 Visual Analytics for Data Exploration

Visualisation is the process of converting data into pictures for easier human interpretation. This facilitates the cognitive process of acquiring information and knowledge from the data [24]. Colours, spatial relationships, dimensions and patterns are mapped to the different features of the dataset for extracting and communicating the underlying meaning. Therefore, information can be extracted from the data and synthesised into knowledge that the engineer uses to move the product design forward.

Challenges arise when the number of data dimensions is greater than three. Scatter plots can ideally support up to 5 dimensions by introducing scales for hue, saturation, marker size and marker shape on top of the two axes. In practice the result would be a very cluttered and hard to read diagram.

Parallel Coordinates (PC) diagrams allow one to support many dimensions and provide a birds-eye view of the relationships within the data. PCs are composed of parallel axes for each dimension of the dataset. Each data entry is represented with a polyline that intersects the axes at the value of each dimension. PCs can effectively display data up to 10-15 dimensions [25]. Figure 3 shows an example of a PC plot for the iris dataset.

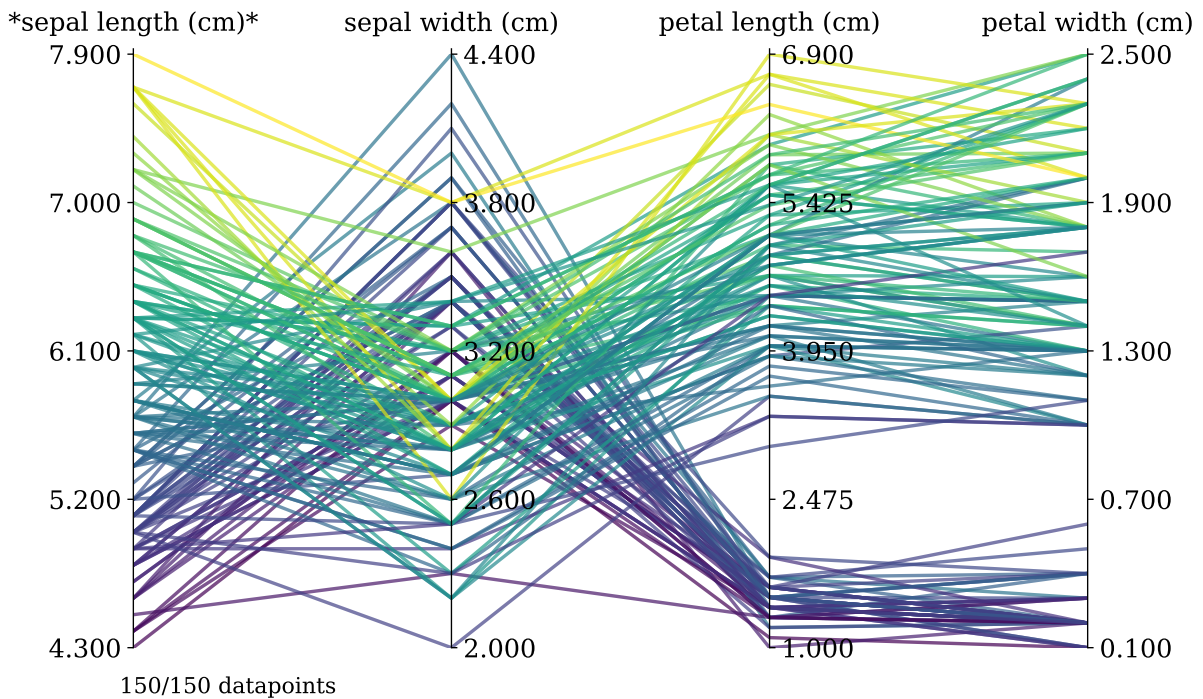


Figure 3: Parallel coordinates plot of the iris dataset.

Parallel Coordinate diagrams are particularly suited for pattern discovery and exploratory data analysis. Inselberg [26] famously published an article where he describes the process as a "multidimensional detective". In essence, PCs shine in their ability to propagate selections from one axis onto all the dimensions. This enables to analyse data more quickly and identify patterns. For example, in Fig. 4 the setosa and versicolor datapoints have been selected and highlighted. The setosa class specifically has a clear separation in petal width and petal length: after selecting these datapoints by limiting the range of petal length between 1 and 2 cm, it is discovered that the sepal length for this

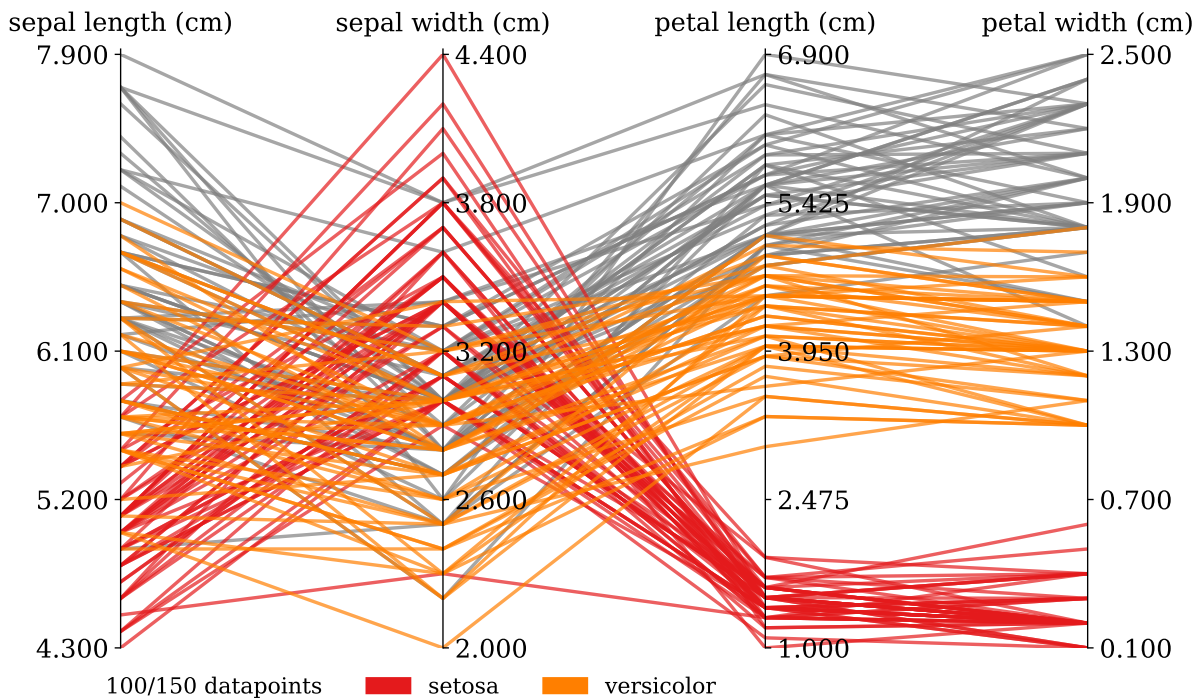


Figure 4: Selections in a PC plot.

class of flowers ranges between 4.3 and 5.8 cm. The same effect can be achieved globally by applying a colour scale reflecting the values one axis on the lines. Fig. 3 shows how the sepal length is correlated with the other dimensions. Other than colour and selections, PCs portray the correlation between each adjacent axis through the disposition of the lines.

For these reasons, PC plots are best suited for interactive visualisation tools. Specifically, the ability to change the order of the axis and select ranges within the axis themselves provide the user with exploratory capabilities that cannot be obtained with scatter plots alone. Kipouros et al. [27] demonstrated an interactive visualisation tool that combines PC and scatter plots. The engineer would use the PC plot as a guiding tool to isolate interesting data points, and the scatter plots as a magnifying scope for understanding the correlations between two specific dimensions or visualising the Pareto front. Inselberg's book on the topic [28] is the definitive resource for implementing and utilising PCs.

A further enhancement of interactive visualisation is the addition of regression and machine learning methods to extract new information on the fly. The combination of these has been named *Visual Analytics*. Piotrowski et al. [29] presented an enhanced tool in which kriging was used to produce new datapoints and support the exploration of hypotheses and 'what-if' scenarios.

In this study, considerations regarding the technological parameters have been extracted using a web-based interactive visualisation tool, implementing both PC and scatter plots [30]. The strengths of both diagrams is enhanced thanks to their integration, with selections propagating from one plot to the other.

3 EXEMPLAR CASE STUDY: ENERGY STORAGE FOR REGIONAL AIRCRAFT

The aim of this case study is to analyse the performance of a battery pack with futuristic technology. The results will provide insights into the behaviour of the energy storage system in the context of a regional hybrid electric aircraft, a turboprop of the ATR-72 class, retrofitted with a parallel hybrid-electric propulsion system.

The propulsion system is a hybrid parallel system, where the gas turbine shaft and the electric motor shafts are connected to the propeller through a gearbox. The battery pack provides power to the electric motor according to the energy management strategy (EMS), which sets the degree of hybridisation (DOH) during the mission. DOH is defined as the percentage of propeller power required to be provided by the electrical powertrain. EMS in this study is defined as a linear function of DOH, one for each mission segment.

In the mission analysis procedure, the battery pack is first sized to match the load required by the EMS. Then an ageing simulation is carried out, assuming 14 flights every week for an entire year of operations. Since it is expected that the capacity of the battery will reduce due to use, which under the original EMS would cause the battery to overdischarge, the EMS is automatically scaled such that the depth of discharge (DOD) of the battery does not exceed 80%.

The gas turbine model predicts fuel consumption by interpolating performance data from a simulated PW-127, while NO_x emissions are captured using the P3T3 method. The battery model consists of a Thevenin equivalent circuit model of the cell dynamics, which are then scaled using the layout of the pack. The ageing of the cells is simulated with a polynomial model which captures both degradation caused by time and charge-discharge cycles. This updates the lumped parameters and the total capacity of the cell model, affecting its dynamics.

Details on the modelling can be found in literature published by the authors [31].

$$\begin{array}{ll}
 \text{Input} & X = [h0_{climb}, h1_{climb}, h0_{cruise}, h1_{cruise}] \\
 \text{minimise} & M_{fuel}, M_{NO_x}, r_{degr} \\
 \text{subject to} & TOM \leq 23,000 \text{ kg } (P_{sat} \geq 50\%) \\
 & e_{battery} = 750 \text{ Wh/kg}
 \end{array} \tag{1}$$

The optimisation problem is formulated in Eq.(1). The objective is to minimise fuel consumption, NO_x emissions and the degradation of the battery pack after one year of operations. This last parameter is defined as the ratio between the fuel consumption after one year of operations and the original fuel consumption with a fresh battery.

Since the aeroplane is not fully re-designed, the take-off mass was constrained to the ATR-72 MTOM of 23,000 kg. In this specific scenario, it is assumed some of the payload mass of the ATR-72 is used for accommodating batteries, hence the number of passengers is reduced to 50. The constraint is marked with a satisfaction probability of 50%.

The mission used for sizing and analysing the propulsion system is the design mission of project FUTPRINT50 [32], a regional flight of 800 km from Edinburgh to Dublin, with an alternate of 200 km to Belfast. The EMS is specified only for the climb and cruise segments, since they have been shown to be the most significant in the reduction of emissions [17]. Hence, the number of input parameters is in total 4, two DOH for

climb and two DOH for cruise. Tab.1 shows the range and number of levels of the input parameters. These in total yield 256 sets to be evaluated.

Table 1: Details on Input Parameters.

Name	Symbol	Range	Number of Levels
DOH at beginning of climb	$h0_{\text{climb}}$	[0,1]	4
DOH at end of climb	$h1_{\text{climb}}$	[0,1]	4
DOH at beginning of cruise	$h0_{\text{cruise}}$	[0,1]	4
DOH at end of cruise	$h1_{\text{cruise}}$	[0,1]	4

Results from the analysis will be compared against the dataset published previously by the authors [31], after normalising the fuel consumption and the emissions against the baseline value (the aircraft evaluated with DOH set to zero in the entire mission). The baseline values can be found in Tab.2.

Table 2: Baseline Quantities.

Take Off Mass	17,792 kg
Fuel Consumption	1242 kg
NO _x Emissions	8.59 kg

4 RESULTS

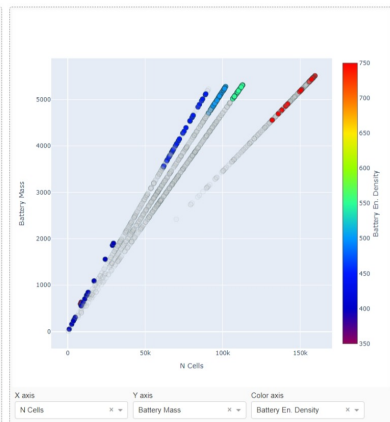
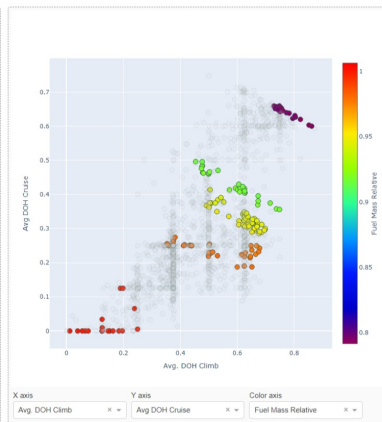
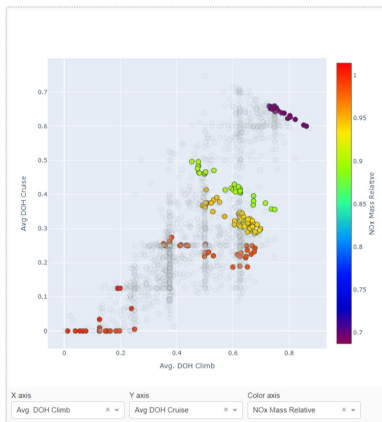
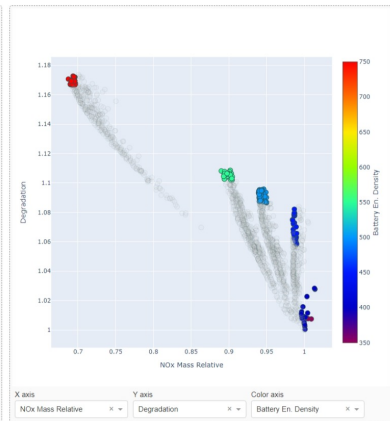
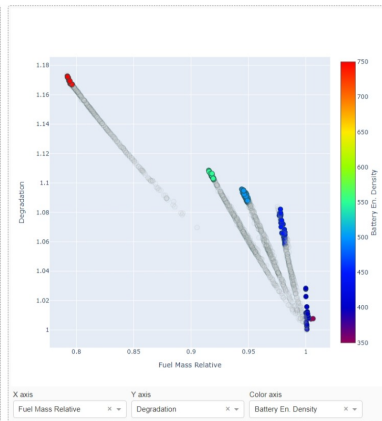
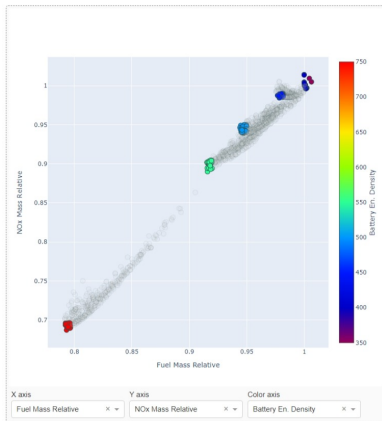
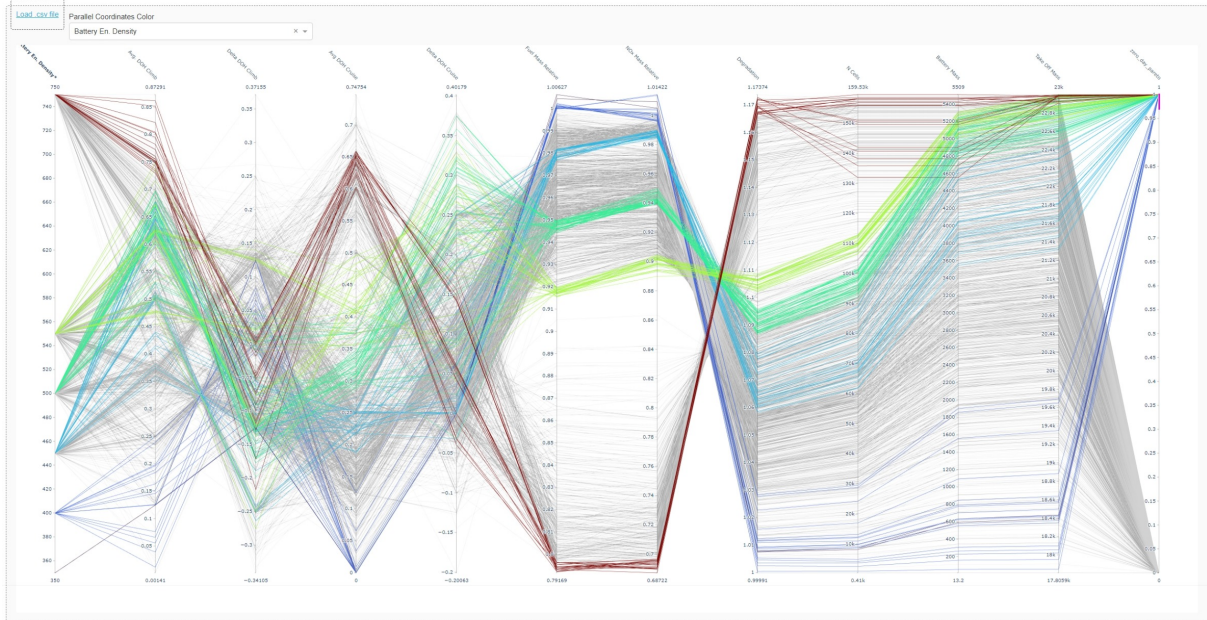
Figure 5 shows the web-based interactive visualisation tool used for analysing the dataset produced in this test case. Data is selected in the PC diagram and propagated onto the six available scatter plots. Each of them can be defined by selecting the columns to be shown in the X, Y, and colour dimension.

The parallel coordinates plot of Figure 6 presents the Pareto front results of the test case design space exploration, integrated with the results previously published [31], such as to highlight the effects of an energy storage system with very high energy density, as the previous study considered only values in the range 350-550 Wh/kg. The visualisation enables us to identify the relationships between the input parameters (EMS), the objectives, and the constraints. The input parameters have been transformed into average DOH and Δ DOH to better highlight the effects of the objectives on the EMS topology. The selections reflect the points with the lowest fuel consumption and NO_x emissions, for all levels of degradation.

Figure 7 compares the three objectives in scatter plot form. As it was evident in Figure 6, the degradation objective is in strong contrast with the other two performance objectives. Indeed, as it was found in previous studies [31], having higher DOH for better performance with a fresh battery caused a short operational life of the battery pack. EMS with lower average DOH fared better but at the cost of smaller improvements in fuel consumption and NO_x emissions.

Another point of note is that the high energy density of 750 Wh/kg enabled higher average DOH compared to previous results, with very high amounts in the climb segment. Consequently, both the fuel consumption and the NO_x emissions have been significantly

FUTPRINT  Decision making environment



V.0.55, Copyright Cranfield University

Figure 5: Decision-making Visualisation Framework webtool.

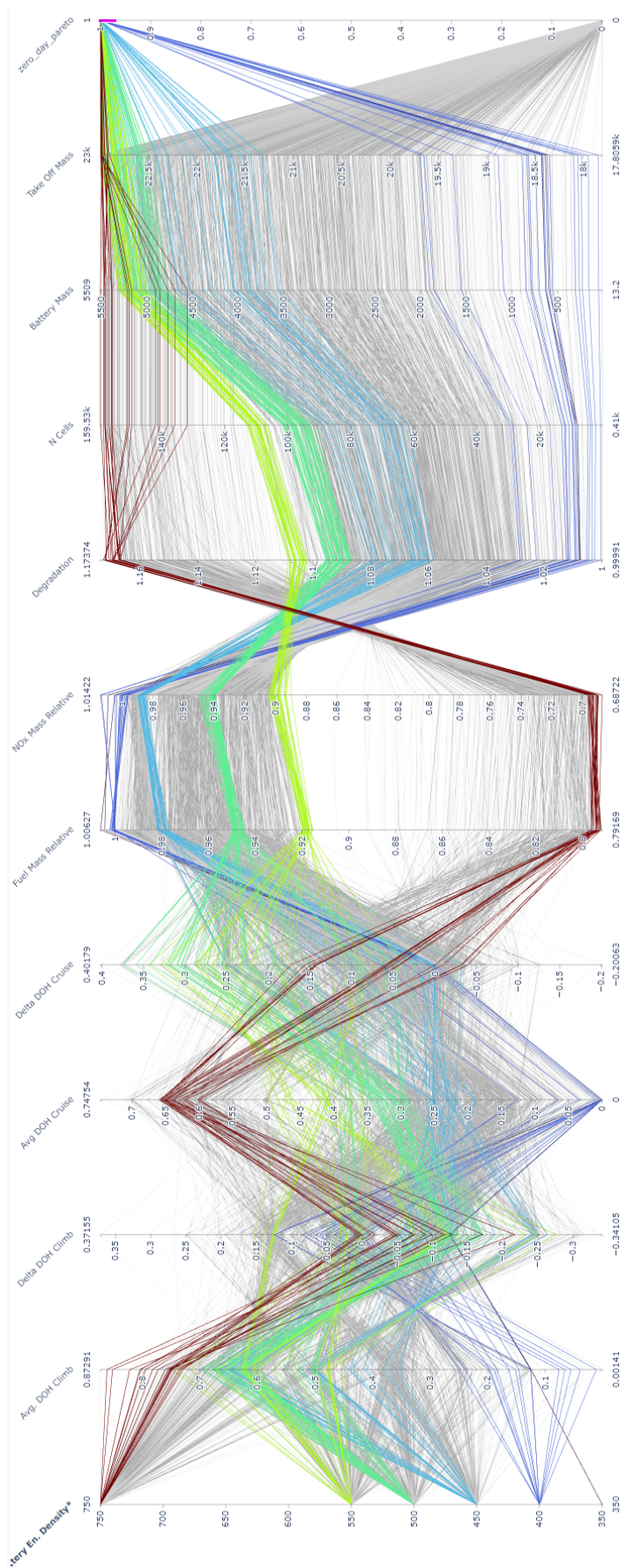


Figure 6: Parallel Coordinates comparison of the test case results and the data from [31].

reduced. On the other hand, the degradation is significantly higher, with the maximum being 17.5%, while the highest of 550 Wh/kg is only 11%. The mechanism causing this was identified in the mass of the battery and the reduction of effective energy density as the capacity of the cells degrade [31]. Nonetheless, with the higher energy density it is possible to achieve better performance for the same value of degradation. For instance, with $r_{degr} = 1.074$, the 750 Wh/kg case achieves a reduction of 10.5% of fuel consumption and 13.6%, while the 550 Wh/kg case yields 6.1% and 8.6% respectively.

Thus, investing in better energy storage technologies would grant much better environmental compatibility, even when taking into account the degradation of the battery packs. Most importantly, the tendency visible from the Pareto front (Figure 7a) suggests a non-linear growth of the reduction of emissions and fuel consumption with the improvement of the technology. Henceforth, higher returns should be expected in investing in projects that aim to significantly improve energy storage, rather than gradually improve it.

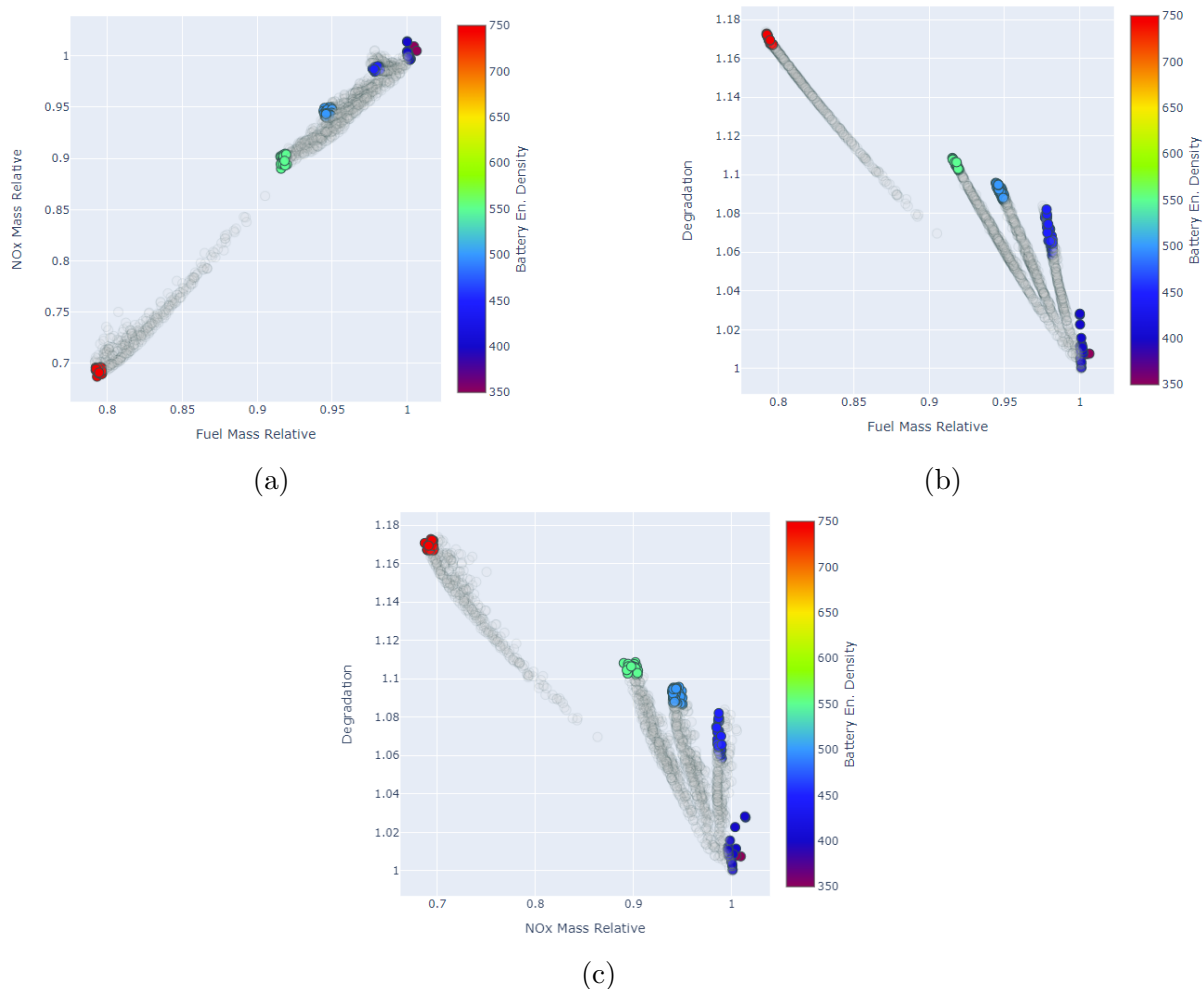


Figure 7: Pareto Fronts of the three objectives, with selections from Fig. 6

The two performance objectives present an indication of topology preference. When fuel consumption is minimised, the climb EMS is descending (high DOH at the beginning of the climb phase), while the cruise EMS increases its DOH as the aeroplane gets lighter.

Andrea Spinelli, Gustavo P. Krupa and Timoleon Kipouros

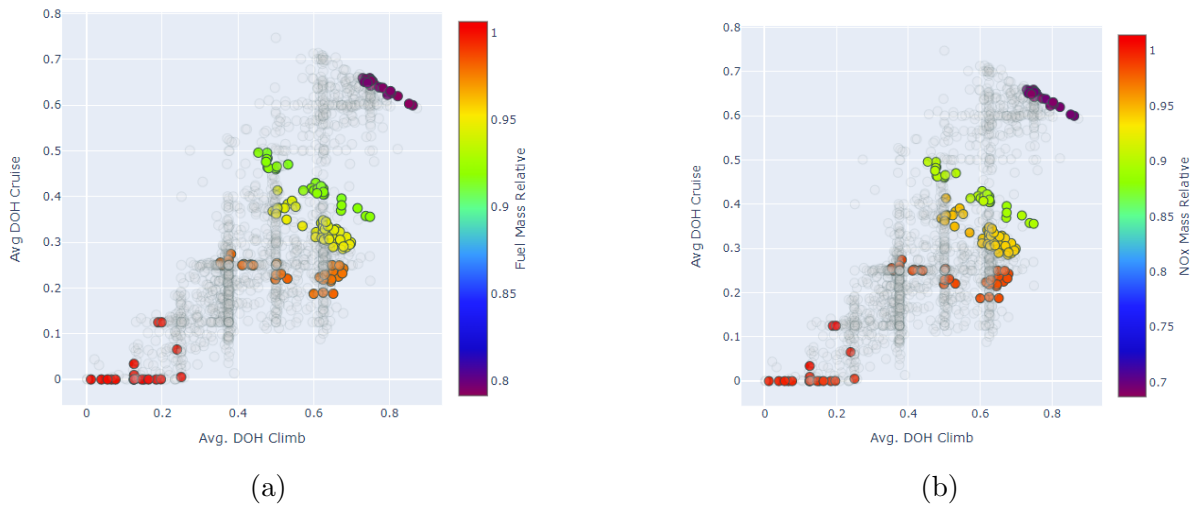


Figure 8: Average DOH of Climb and Cruise for NO_x and Fuel consumption.

Conversely, if minimal NO_x is the priority, the EMS of both mission phases are quasi-constant, with a slight tendency towards decreasing-DOH in climb and increasing-DOH in cruise. This result is coherent with previous findings [17], where NO_x emissions were found to be correlated with the average DOH through the mission. Hence distributing electrical power across the mission would help reduce it the most. This is also shown in Figure 8, reflecting the different sensibilities of the two objectives.

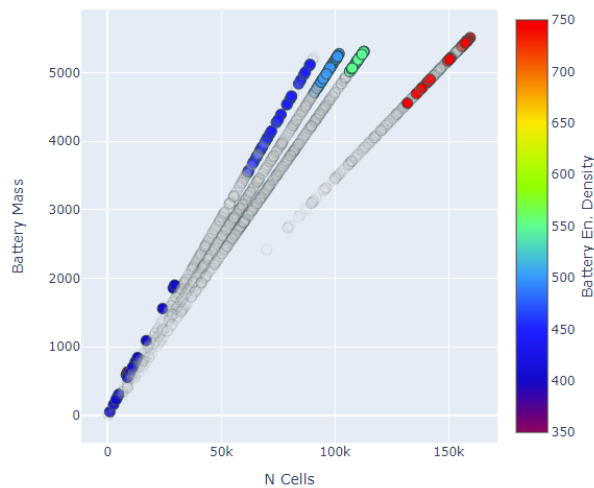


Figure 9: Comparison between total battery mass and number of cells.

Figure 9 compares the number of cells installed in the battery pack versus the total mass of the pack itself. The data points are aligned in lines with different slopes, increasing as the energy density increased. The cause is to be traced in the mass of the individual cell. As the energy density of the pack grows, each individual cell is lighter, hence it is possible to use more of them for the same amount of mass available on board. Furthermore, it enables greater flexibility in arranging the cells without exceeding the maximum take-off weight. In fact, the design points with high energy density also have the highest take-off mass (Figure 6), due to this ability to exploit all the available battery mass before violating the take-off mass constraint.

Finally, the trade-off between climb and cruise DOH appears to follow a consistent linear relationship for the maximum benefit of the emission objectives. This hypothesis was further tested and verified using the embedded Machine Learning model in the interactive visualisation software developed by Piotrowski et al. [29], and shown in Figure 10.

5 LIMITATIONS AND FURTHER WORK

We considered the utilisation of a web-based decision-making software environment, and a Visual Analytics decision-making software, for the analysis of the behaviour of the energy storage system and its impact on a new regional aircraft architecture. The two platforms offered complimentary functionalities for the analysis of a multidimensional data set and to discover further insights. The web interface will be further improved to include uncertainty visualisation supported with the latest Artificial Intelligence algorithms for instant analysis and computational experimentation for the understanding of the impact of new technology in derivative airframe configurations. The analysis will be further extended to consider interactions with other key enabling technologies as shown in Figure 1.

6 CONCLUSIONS

In this work, we combined a set-based design space exploration methodology to analyse the behaviour of gradually increased energy density of batteries. We combined these results with interactive visualisation of multidimensional data in order to discover relationships between all parameters and decision criteria. This analyses led to the expression of hypotheses which were further tested in a Visual Analytics software environment.

We discovered a non-linear improvement in the emission criteria when the energy density is 750 Wh/kg at the pack level. This value expresses a threshold for technology advancements. However, this raises questions about safety, which can further compromise the pack energy density, as well as the degree of degradation. But most of all, even if there will be a technology to achieve these advanced technical characteristics, there will be the question of whether these can be achieved on a scale and at an industrial level.

Finally, we discovered a consistent linear relationship between the DOH at climb and cruise for different energy density values that secure minimum achievable generation of emissions.

To conclude, the proposed methodology can be used to explore interactions of non-conventional aircraft architectures and systems when little or no prior knowledge of their behaviour and interactions is available.

ACKNOWLEDGEMENTS

This project has received funding from Innovate UK under Grant Agreement No 10003388.

REFERENCES

- [1] I. C. A. Organization. Aviation and the environment: A global perspective. 2020. URL <https://www.icao.int/environmental-protection/Pages/default.aspx>.
- [2] I. A. T. Association. Sustainable aviation: A global challenge. 2020. URL <https://www.iata.org/pressroom/pr/Pages/2020-02-14-sustainable-aviation.aspx>.

- [3] A. T. A. Group. The future of aviation: A sustainable vision. 2020. URL <https://www.at2050.org/>.
- [4] W. E. Forum. *Sustainable Aviation: A Global Challenge*. World Economic Forum, Geneva, Switzerland, 2020. URL <https://www.weforum.org/reports/sustainable-aviation-a-global-challenge>.
- [5] A. L. Malinga and R. Oosthuizen. Technology roadmapping and systems engineering—a symbiotic relationship? 2021.
- [6] C. Kerr. The systems approach of strategic roadmapping: Framing challenges and contributions whilst flexing to changing conditions and circumstances. *Systems Research and Behavioral Science*, 2023.
- [7] S. Kim, H. Jang, and B. Yoon. Developing a data-driven technology roadmapping method using generative adversarial network (gan). *Computers in Industry*, 145: 103835, 2023.
- [8] M. G. Moehrl, R. Isenmann, and R. Phaal. Technology roadmapping for strategy and innovation. *Charting Route to Success*. Berlin et al. Springer, 2013.
- [9] O. L. De Weck. *Technology Roadmapping and Development: A Quantitative Approach to the Management of Technology*. Springer Nature, 2022.
- [10] N. Moebs, D. Eisenhut, E. Windels, D. Bergmann, I. Geiß, R. Reis, and A. Strohmayer. Selecting figures of merit for a hybrid-electric 50-seat regional aircraft. In *IOP Conf. Ser.: Mater. Sci. Eng.*, volume 1024, page 012071, 2021.
- [11] D. Eisenhut, N. Moebs, E. Windels, D. Bergmann, I. Geiß, R. Reis, and A. Strohmayer. Aircraft requirements for sustainable regional aviation. *Aerospace*, 8:61, 2021. doi:10.3390/aerospace8030061.
- [12] D. Eisenhut, N. Moebs, E. Windels, D. Bergmann, I. Geiß, R. Reis, and A. Strohmayer. Foundations towards the future: Futprint50 tlars an open approach. In *IOP Conf. Ser.: Mater. Sci. Eng.*, volume 1024, page 012069, 2021.
- [13] M. A. Rendón, C. D. Sánchez R, J. Gallo M, and A. H. Anzai. Aircraft hybrid-electric propulsion: Development trends, challenges and opportunities. *Journal of Control, Automation and Electrical Systems*, 32(5):1244–1268, 2021.
- [14] K. Edström, E. Ayerbe, I. Cekic-Laskovic, R. Dominko, M. Fichtner, A. Grimaud, J. Kumberg, S. Perraud, C. Punckt, and T. Vegge. Battery 2030+: Inventing the sustainable batteries of the future. research needs and future actions. Battery 2030+ roadmap, 2022. URL https://battery2030.eu/wp-content/uploads/2022/07/BATTERY-2030-Roadmap_Revision_FINAL.pdf.
- [15] Y. Lu, X. Rong, Y.-S. Hu, L. Chen, and H. Li. Research and development of advanced battery materials in china. *Energy Storage Materials*, 23:144–153, 2019. ISSN 2405-8297. doi:<https://doi.org/10.1016/j.ensm.2019.05.019>.

- [16] F. IFAM. POWERPASTE - energy storage solution. www, 2022. URL https://www.ifam.fraunhofer.de/content/dam/ifam/de/documents/dd/Infobl%C3%A4tter/POWERPASTE_energy_storage_solution_fraunhofer_ifam_dresden.pdf.
- [17] A. Spinelli, H. B. Enalou, B. Zaghari, T. Kipouros, and P. Laskaridis. Application of probabilistic set-based design exploration on the energy management of a hybrid-electric aircraft. *Aerospace*, 9(3):147, 03 2022. doi:10.3390/aerospace9030147.
- [18] D. J. Singer, N. Doerry, and M. E. Buckley. Application of probabilistic set-based design exploration on the energy management of a hybrid-electric aircraft. *Naval Engineers Journal*, 121(4):31–43, 2009. doi:10.1111/j.1559-3584.2009.00226.x.
- [19] T. A. McKenney, L. F. Kemink, and D. J. Singer. Adapting to changes in design requirements using set-based design. *Naval Engineers Journal*, 123(3):67–77, 2011. doi:10.1111/j.1559-3584.2011.00331.x.
- [20] R. Costa and D. K. Sobek. Iteration in engineering design: inherent and unavoidable or product of choices made? In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 37017, pages 669–674, 2003. doi:10.1115/DETC2003/DTM-48662.
- [21] S. Dullen, D. Verma, M. Blackburn, and C. Whitcomb. Survey on set-based design (sbd) quantitative methods. *Systems Engineering*, 24(5):269–292, 2021. doi:10.1002/sys.21580.
- [22] A. Spinelli, L. Anderson, H. B. Enalou, B. Zaghari, T. Kipouros, and P. Laskaridis. Application of probabilistic principles to set-based design for the optimisation of a hybrid-electric propulsion system. In *IOP Conference Series: Materials Science and Engineering*, volume 1226-1, page 012064. IOP Publishing, 2022. doi:10.1088/1757-899X/1226/1/012064.
- [23] C. University. PDOPT - probabilistic design and optimisation. GitHub, 2022. URL <https://github.com/FutPrInt50/Probabilistic-Design-Optimisation---PDOPT>.
- [24] M. Chen, D. Ebert, H. Hagen, R. S. Laramee, R. van Liere, K.-L. Ma, W. Ribarsky, G. Scheuermann, and D. Silver. Data, information, and knowledge in visualization. *IEEE Computer Graphics and Applications*, 29(1):12–19, 2009. doi:10.1109/MCG.2009.6.
- [25] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the first IEEE conference on visualization: visualization90*, pages 361–378. IEEE, 1990. doi:10.1109/VISUAL.1990.146402.
- [26] Multidimensional detective. In *Proceedings of VIZ'97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pages 100–107. IEEE, 1997. doi:10.1109/INFVIS.1997.636793.
- [27] T. Kipouros, A. Inselberg, G. Parks, and A. M. Savill. Parallel coordinates in computational engineering design. In *54th AIAA/ASME/ASCE/AHS/ASC*

- Structures, Structural Dynamics, and Materials Conference*, page 1750, 2013. doi:10.2514/6.2013-1750.
- [28] A. Inselberg and B. Dimsdale. Parallel coordinates. *Human-Machine Interactive Systems*, pages 199–233, 2009.
- [29] W. Piotrowski, T. Kipouros, and P. J. Clarkson. Enhanced interactive parallel coordinates using machine learning and uncertainty propagation for engineering design. In *2019 15th International Conference on eScience (eScience)*, pages 339–348, 2019. doi:10.1109/eScience.2019.00045.
- [30] C. University. PDOPT - probabilistic design and optimisation. GitHub, 2022. URL <https://github.com/FutPrInt50/Decision-Making-Environment>.
- [31] A. Spinelli, G. P. Krupa, and T. Kipouros. Set-based design space exploration to investigate the effect of energy storage durability on the energy management strategy of a hybrid-electric aircraft. In *AIAA SCITECH 2023 Forum*, 2023. doi:10.2514/6.2023-0837.
- [32] N. Moebs, D. Eisenhut, E. Windels, J. van der Pols, and A. Strohmayr. Adaptive initial sizing method and safety assessment for hybrid-electric regional aircraft. *Aerospace*, 9(3):150, 2022. ISSN 2226-4310. doi:10.3390/aerospace9030150.



AIRBUS - PIONEERING SUSTAINABLE AEROSPACE

António Teixeira Da Costa

VP Single-Aisle Marketing
Airbus

Toulouse, France

antonio.t.da-costa@airbus.com, <https://www.airbus.com/>

Abstract. *Airbus is a global aerospace pioneer, operating in the commercial aircraft, helicopter, defence and space sectors. It has built on its strong European heritage to become truly international. It has constantly innovated since its inception, bringing novelties such as the two-crew cockpit, fly-by-wire or composite structures. Looking into the future, the aerospace sector faces a new challenge which is that of sustainable flying. Airbus is embracing this challenge through an industry-wide multi-prong approach that is based upon: fleet renewal with the latest generation aircraft, introducing disruptive technologies, improving the air transport operations and infrastructure, introducing Sustainable Aviation Fuel (SAF), and applying carbon offsetting and capture techniques. The combined effect of these measures is expected to bring the air transport sector to net zero carbon emissions by 2050. From a future aircraft design standpoint, disruptive technologies are being developed across all domains to improve aircraft weight and aerodynamics, as well as developing radically new propulsion systems. Several novel propulsion technologies are entering the testing phase that hold promise for future aircraft applications, such as distributed hybrid propulsion and open fan engines. The energy sources are also being disrupted to remove the need to rely on fossil fuels, with two major pathways emerging, namely SAF that allows up to 85% reduction in carbon emissions across the lifecycle, and hydrogen that could reach up to 100% reduction. These developments can only be brought to life with the intellectual and innovation power of highly skilled professionals, therefore Airbus is actively recruiting the future team members to pioneer sustainable aerospace for a safe and united world.*

Keywords: aerospace, next generation, disruptive technologies, carbon emissions, future aircraft design, sustainable aviation fuel, propulsion technologies



WING STRUCTURAL DESIGN FOR A MAME UAV USING HIGH-FIDELITY NUMERICAL TOOLS

Vítor M. T. Silva¹, Nuno M. B. Matos² and André C. Marta¹

1:IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
{vitortavaressilva8, andre.marta}@tecnico.ulisboa.pt, <http://mdo.tecnico.ulisboa.pt>

2: Research and Development
Tekever UAS
Tekever
2500-750 Caldas da Rainha
nuno.matos@tekever.com, <http://www.tekever.com>

Abstract. *With the rapid growth of the UAV market, the search for more efficient solutions promotes a huge competitive advantage for manufacturers. With the implementation of optimization techniques and the use of high-fidelity analysis in aircraft design, it is possible to develop better solutions. This work addresses the desire of a leading UAV manufacturer to improve its fleet to remain competitive in the surveillance UAV market. For this, a structural analysis tool using the finite element method is demonstrated, which is then used as part of a structural optimization framework. For this demonstration, static analyses of the wingbox of an existing UAV model, with a CFRP material with different lay-ups in certain areas of the model, are carried out for cruise and 4g load case, obtaining results of deformation and failure of this wing. These results help to identify possible weaknesses of the wing, as well as evaluate how the wingbox structural behaviour changes. The goals of this work include the validation of the numerical design framework using available experimental data and the study of alternative wing structural solutions. The results of the experimental and computational analyses presented slight differences. This was the expected behaviour due to model simplifications, which allowed for the the framework to be validated and proven useful. Three new optimal wingbox solutions were found, having a theoretical mass reduction of about 50%, while respecting a safety factor of 1.5. The first was optimised without displacement constraints and the other two had a maximum allowed displacement. These two differ on the optimization starting point to check for possible local minima, which were found.*

Keywords: optimization, design framework, adjoint method, finite element method, composite materials, fiber orientation

1 INTRODUCTION

With the continuous development and technological improvements of the Unmanned Aerial Vehicles (UAV) industry, different sizes and configurations were found useful to cover a wide range of missions. Nowadays, these vehicles are equipped with various electronic equipment, such as different cameras and sensors, depending on their intended purpose, and turning these in highly complex systems. Some use cases of these Unmanned Aerial Systems (UAS) and their advantages include: inspection of photovoltaic plants, to reduce the time of manual inspections [1]; precision agriculture, for weed mapping and management, vegetation health and growth monitoring, irrigation management and crops spraying [2]; urban environment and management, to give real-time monitoring of traffic, road conditions as well as building observation [3]; disaster hazard and rescue, to quickly and accurately respond when necessary as well as post-disaster assessment and emergency response in remote places; and maritime monitoring, to control oil spills, gas pipes, whales and marine life, tracking fishing boats as well as controlling illegal immigration.

Various applications have been brought up and many more are yet to be developed. Investments in the UAS industry have been a trend for the past couple of years, with many companies, mainly start-ups, directing their funds to explore new and original applications to be competitive. Since 2000, more than 300 start-ups entered the UAS market, focusing on hardware, support services and operations [4]. They have gathered more than 3 billion USD to develop the industry with new applications.

In order to remain competitive in this growing market, the demand for highly efficient and optimized UASs increases.

2 TEKEVER'S AR5 UAS

Tekever is a company founded in 2001 that manufactures and operates their UAS mainly for surveillance missions. An objective was set to optimize their most advanced Medium Altitude Medium Endurance (MAME) UAS - the AR5 (Fig. 1) - using high-fidelity tools. This paper is the starting point of this project, introducing the numerical tools and proving the concept.



Figure 1: Tekever AR5

The main AR5 specifications are resumed in Tab. 1.

For take-off and landing specifications, this UAS needs a track. As a drawback, it can not take-off with crosswinds. Regarding the payload, one advantage is that, for rescue

Table 1: Some characteristics of the Tekever’s AR5 [5]

Wing span	Cruise speed	MTOW	Payload capacity	Endurance	Communications range
7.3 m	100 km h ⁻¹	180 kg	50 kg	12 h*	Unlimited (SATCOM)

* Expected endurance with inflating raft

missions, that may require fast action on site, the AR5 can transport an inflating raft. However, this added weight might affect the total endurance, and reduce total operating time from 16h to 12h. Finally, since it uses Satellite Communication (SATCOM) for communication with the operator, there is no range limit for operating, which is really advantageous.

Since the performance evaluation of the configuration come usually from estimations with empiric correlations, very simplified calculus or, in later stages, computational solid mechanics (CSM) for a structural case, it would benefit from the creation of some automated steps. In [6], Grose explains that the competitiveness of a company is high if: it can provide fast responses to consumer needs by reducing development or manufacturing times; the aircraft is produced with the least possible costs; and if it requires low maintenance, indicating a good final product quality. Using optimization as a tool to improve in all these aspects, the design traditional process can be adapted by having an optimizer directly and automatically changing some pre-established design variables. By combining this with a judgment of an experienced engineer, the final product should also have a superior quality.

3 STRUCTURAL ANALYSIS AND OPTIMIZATION

3.1 Finite Element Method

For the structural analysis presented in this paper, the FEM method is used due to its high capabilities of modelling complex geometries and its high use in the aeronautic industry for structural applications.

For thin structures, like the wing’s skin, 2D shell elements are used to model the middle plane of the skin [7].

In this work, the elements used in the analysis are the 4-node quadrangular elements based on a Mixed-Interpolation of Tensorial Components (MITC) approach, which avoid shear and membrane locking [8, 9]. It is important to note that a modelling error is introduced which makes these not suitable for shell structures with high thickness and curvature radius.

For these elements, the basis theory is the classical first order deformation theory [9]. In this, the transverse plane does not remain perpendicular to the mid plane after deformation including shear strains in model. With it, the variables to be determined in this formulation are the mid-plane displacements and the rotations about the x and y axis. To add the rotation about z (also called "drilling degree of freedom"), a penalty approach is used [10, 11]. Then, with laminate constitutive equations, the stresses are obtained from the strains.

3.2 Material Characterization and Modelling

The structures to be analysed are composed of Carbon Fiber Reinforced Polymer (CFRP) plies that, when stacked, form the laminate. Since each ply is composed of unidirectional fibers in a matrix medium, it presents an orthotropic behaviour, with 9 independent properties that define this material constitutive law:

$$\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \\ \varepsilon_6 \end{Bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{Bmatrix} \quad (1)$$

with E_i as the Young's Modulus for the i th direction, ν_{ij} as the ratio between transverse shear in the j th direction to the axial strain in the i th direction with an applied stress in the i th direction (Poisson's ratio) and G_{ij} as the shear modulus for the ij th plane [9].

To model the composite material that composes the wingbox, a smeared approach is used [12]. This method describes the composite structure using fractions of plies (f_{θ_i}) and their respective angles (θ_i). It is necessary to provide the elastic and strength properties of each ply as well as the total thickness of the laminate. With these, the global stiffness matrix is weighted with each ply fraction of a certain orientation. A visual diagram of the geometric parameters necessary for this model is presented in Fig. 2.

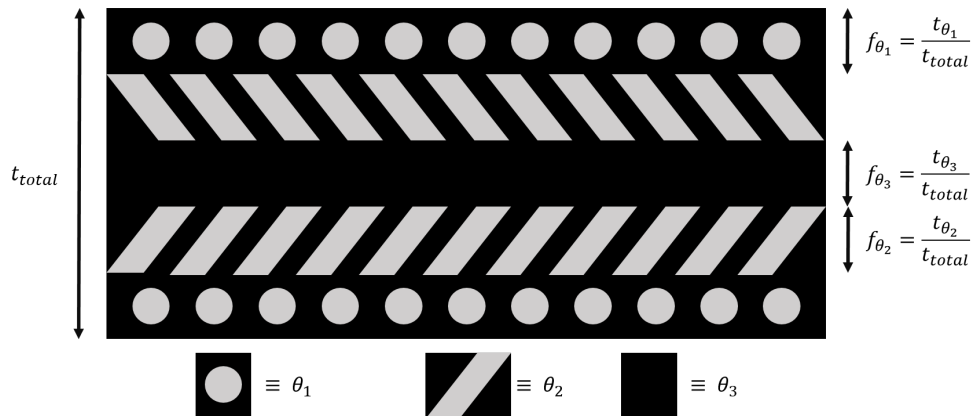


Figure 2: Composite representation with the necessary parameters to define the smeared model

To predict failure of composite materials, it is necessary to characterize the composite strength for different load conditions (axial, transverse and shear tensile stresses), which are reflected in their longitudinal, transverse and shear tensile strength parameters.

In this work, the *Tsai-Wu* Criterion is used and defined as

$$F_{TW} = F_1\sigma_1 + F_2\sigma_2 + F_{11}\sigma_1^2 + 2F_{12}\sigma_1\sigma_2 + F_{22}\sigma_2^2 + F_{66}\sigma_{12}^2, \quad (2)$$

where σ_i is the stress component in the i th direction in the principal material coordinates of each ply (1 to 3 are normal stresses and 4 to 6 shear stresses) and F_{ij} are coefficients depending on the material normal and shear strengths [9]. Note that, when this F_{TW}

failure index is equal or above 1, material failure is expected and no stress components are present in the 3rd direction of space since the shell elements assume that $\sigma_z = \sigma_3 = 0$.

3.3 Optimization Techniques

The general structural optimization problem can be stated as

$$\begin{array}{ll}
 \text{Minimize} & f(v, u_1, \dots, u_{n_l}) \\
 \text{w.r.t.} & v, u_1, \dots, u_{n_l} \\
 \text{governed by} & R_i(X^N(v_G), v_M, u_i) = 0 \\
 \text{subject to} & g_i(v, u_i) \leq 1
 \end{array} \tag{3}$$

where $f(v, u_1, \dots, u_{n_l})$ is the objective function, $g_i(v, u_i)$ is a constraint vector for the i th load case, up to a total of n_l , $x = (v_G, v_M)$ are the design variables, divided as geometric and material design variables, respectively, $X^N(v_G)$ are the nodal locations, u_i are the state variables for the i th load case and R_i are the finite-element residuals.

to compute gradients, either of the objective function or the constraint functions with respect to the design variables. These are very computationally expensive to calculate and many methods have been developed to make them more efficient to compute. Kennedy & Martins [13] and Kenway et al. [14], for example, used a gradient-based algorithm for an aerostructural optimization of a wing and a wide-body transport aircraft, respectively, due to the large number of design variables (thousands). The most important aspect in these cases was that the derivatives were obtained by the adjoint method. Another use for this type of gradient-based methods is given by Werter & De Breuker [15] by using a globally convergent method of moving asymptotes (GCMMA) to minimize a wing weight with respect to some lamination parameters and the laminate thickness. One of the key aspects in this case was that the derivatives were obtained using the direct method and with the use of analytically obtained sensitivities of the objective and constraints' functions with respect to the design variables.

In the context of this work, the gradient-based optimization seems the most favourable approach since the number of design variables is high and the functions of interest are smooth. One of the most commonly used approaches to solve the optimization problem is recurring to a sequential quadratic programming method. One great example is the Sequential Least Squares Quadratic Programming method (SLSQP) which uses the Han–Powell quasi-Newton method with a BFGS update of the B-matrix and an L1-test function in the step-length algorithm [16]. This method approximates the Lagrangian to a quadratic function and linearizes both equality and inequality constraints [16, 17]. A studied performed by Zhoujie et al. [18] showed that, for a multi-dimensional Rosenbrock function, the SLSQP algorithm performed the best among several other gradient-based and -free methods, converging the fastest, with the least amount of function evaluations. Not only that, but both this algorithm and the Sparse Nonlinear Optimizer (SNOPT) performed the greatest when obtaining the minimum C_D when changing the wing twist or shape. This results were especially good using the adjoint method for the derivatives. This SLSQP method is commonly used in aerodynamic shape optimization [19, 20], airfoil shape optimization problems [21] as well as aerostructural problems [22].

3.3.1 Adjoint Method

To compute gradients, either of the objective function or the constraint functions with respect to the design variables, the chosen method and the one used by the optimization framework, is the adjoint method. Since these gradients are very computationally expensive to calculate, many methods have been developed to make them more efficient to compute. Kennedy & Martins [13] and Kenway et al. [14], for example, used a gradient-based algorithm for an aerostructural optimization of a wing and a wide-body transport aircraft, respectively, due to the large number of design variables (thousands), which is bigger than the number of constraints' functions. On the other hand, if the situation was the opposite, the direct method used by Werter & De Breuker [15] to minimize a wing weight with respect to some lamination parameters and the laminate thickness, would be the most efficient. One of the other key aspects, besides the use of the direct method in this case, was the use of analytically obtained sensitivities of the objective and constraints' functions with respect to the design variables that help gradient-based optimizations to be faster.

The derivatives of an objective function f and the residual of the governing equation $R(x_n, y_i(x_n))$ can be given by the chain rule as

$$\begin{aligned} \frac{df}{dx_n} &= \frac{\partial f}{\partial x_n} + \frac{\partial f}{\partial y_i} \frac{dy_i}{dx_n} \\ \frac{dR}{dx_n} &= \frac{\partial R}{\partial x_n} + \frac{\partial R}{\partial y_i} \frac{dy_i}{dx_n} = 0. \end{aligned} \quad (4)$$

where x_n are the design variables and y_i the state variables. Using a reduced-space (or nested) approach [23] to the problem, the structural analysis of the model is solved repeatedly in each optimisation iteration. This means that the derivatives in Eq. (4) are necessary in each iteration. With variational calculus, the partial derivatives from the first expression can be easily obtained. However, the total derivative $\frac{dy_i}{dx_n}$ is computationally expensive since it requires solving the structural model, after imposing a small perturbation in x_n , to obtain y_i . To tackle this problem of solving the total derivatives, the derivative of the residuals in Eq. (4) can be re-written as

$$\frac{\partial R}{\partial y_i} \frac{dy_i}{dx_n} = -\frac{\partial R}{\partial x_n} \Leftrightarrow \frac{dy_i}{dx_n} = -\left[\frac{\partial R}{\partial y_i}\right]^{-1} \frac{\partial R}{\partial x_n}. \quad (5)$$

Replacing this in Eq. (4) of the function f ,

$$\frac{df}{dx_n} = \frac{\partial f}{\partial x_n} + \frac{\partial f}{\partial y_i} \left[\frac{\partial R}{\partial y_i}\right]^{-1} \frac{\partial R}{\partial x_n}. \quad (6)$$

Using

$$\psi^T \equiv \frac{\partial f}{\partial y_i} \left[\frac{\partial R}{\partial y_i}\right]^{-1} \quad (7)$$

$$\left[\frac{\partial R}{\partial y_i}\right]^T \psi = \frac{\partial f}{\partial y_i} \quad (8)$$

where ψ is called the adjoint vector. Solving the equivalent Eq. (8), also called adjoint equation, to obtain ψ , it can then be replaced in Eq. (5) and the derivative of the function f found. This procedure to compute derivatives is called the adjoint method. From Eq.

(8), it is possible to conclude that in order to obtain the adjoint vector, this equation needs to be solved as many times as the number of functions f that there are in the optimisation problem. So, this problem complexity scales with the number of functions.

3.4 CSM Analyses' Steps and TACS Framework

From the CAD file containing the geometry information of the wingbox to be evaluated up to the output results of the finite element analysis, and the posterior optimization, a sequence of steps need to be followed and different software used, as presented in Fig. 3.

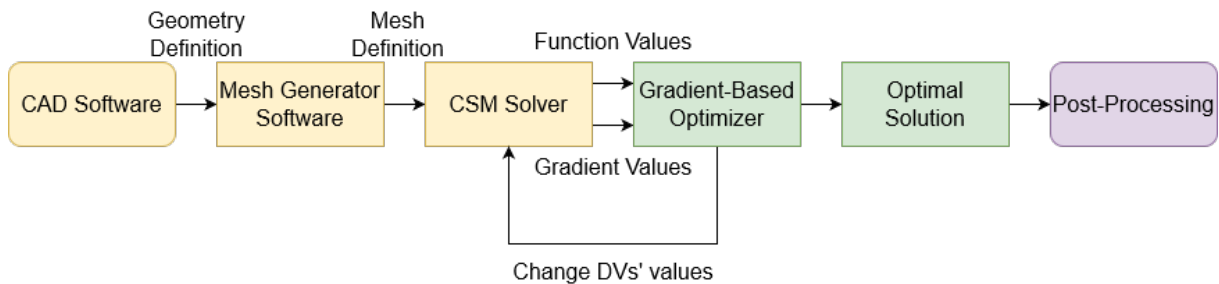


Figure 3: Flowchart of the structural optimization framework with highlight in yellow of the structural analysis components

Firstly, the geometry defined in the CAD needs to be exported as a file with a specific compatible format to be read by a mesh generator software. This geometry definition file contains information about the nodes location, elements identification and nodes which define them and the type of elements used. This is fed into the CSM software so that the finite element model can be used with that discretization.

The CSM software used is TACS [8], an open-source FEM solver. In TACS, load cases are set to condition the problem before it is solved and boundary conditions imported from the mesh definition file. In every problem iteration, the TACS software gives function values and gradients to a gradient-based optimizer, which then iterates until the optimum solution for the intended problem is found. A flow chart representing the TACS workflow is presented in Fig. 4.

4 WING STRUCTURE ANALYSIS

4.1 AR5 Wing Definition

The baseline and start of the optimisation process of the AR5 is its wing, since it is the main component of the UAV. The wingbox can be parametrized in several ways. To do this, some groups within the structure are created, sharing the same parameters, to later, in the optimization process, these varying equally within the group. In Fig. 5 the groups can be observed in different colours as well as the internal wingbox structure and reference frame used throughout its analysis. The parameters that define each group are total thickness, type of materials, their proportions and the ply angles (parameters from Fig. 2).

The AR5 wingbox is composed sandwich composite components with different thicknesses and fractions of core and shell material. A representation of these sandwich components is shown in Fig. 6.

The core consists on a low weight rigid foam which gives the necessary stiffness to

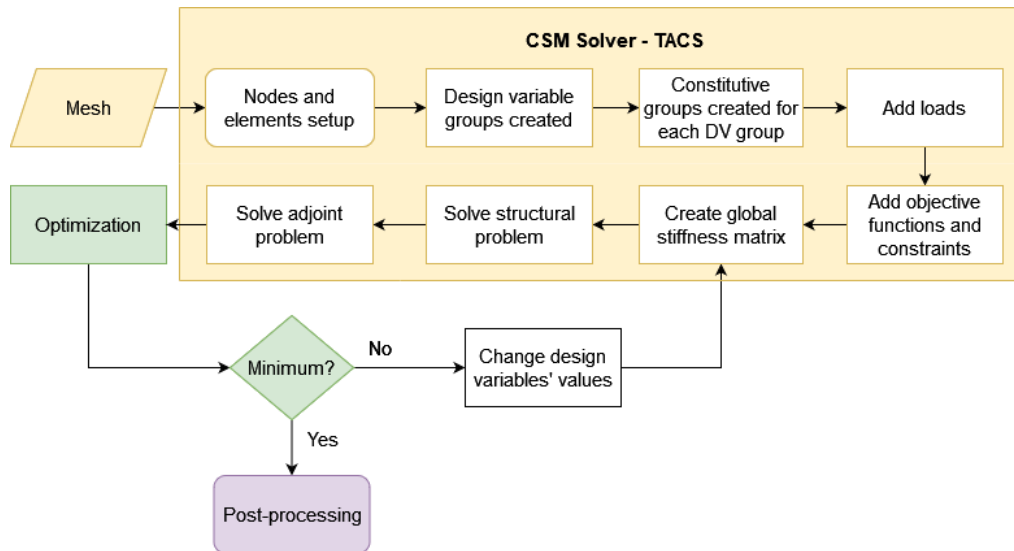


Figure 4: Flow chart of the steps to solve an optimization problem

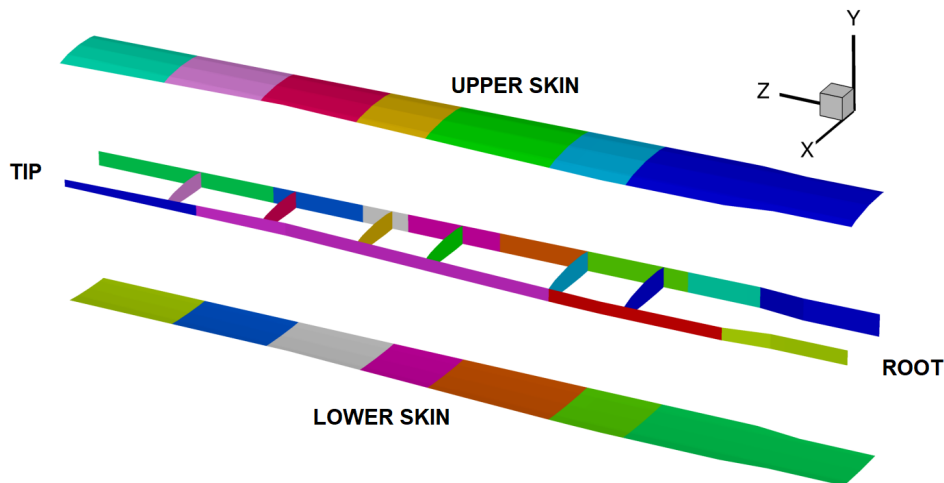


Figure 5: Representation of the design variable groups

the wing, with very low weight. Different versions and proportions of this core are used depending on the wingbox section and a detailed description cannot be provided due to confidentiality restrictions. The physical and mechanical properties used are summarized in Tab. 2. It is important to note that, only the in-plane properties were the necessary manufacturer’s properties since in the shell formulation, the transverse strain is neglected. This means that, since the foam core properties in-plane are independent of direction, this material can be modelled as isotropic.

Table 2: Foam core properties

Density
Tensile modulus in the plane
Shear modulus
Tensile strength in the plane

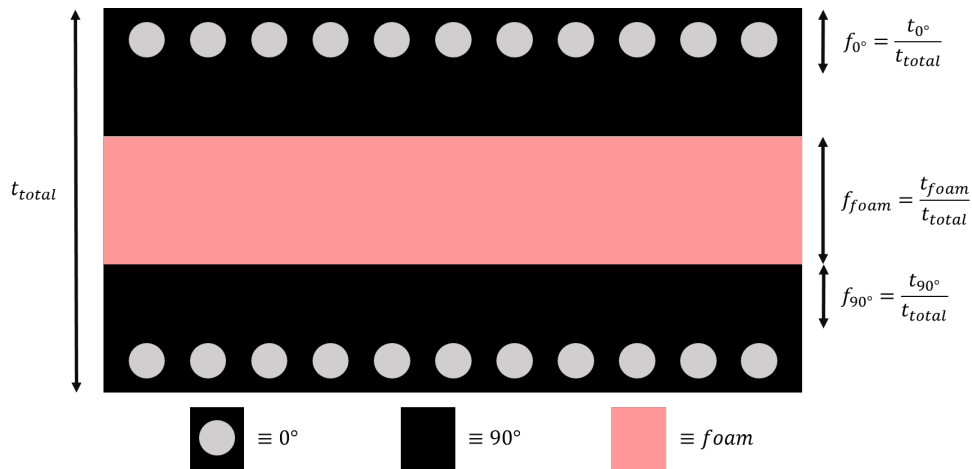


Figure 6: Representation of the sandwich composite (CFRP with black matrix and grey fibers and foam core foam in pink)

Table 3: Density and physical properties of CFRP ply

Density*
Young’s modulus (Fiber Direction)*
Young’s modulus (Transverse Direction)*
Shear modulus
Poisson ratio
Tensile strength (Fiber Direction)
Compressive strength (Fiber Direction)
Tensile and compressive strength (Transverse Direction)
Shear strength

* Obtained from rule of mixtures

For the shell, a CFRP layup is used with plies of 0° and 90° as a starting point. A summary of the properties obtained can be seen in Tab. 3.

To define the material composed of both the foam and CFRP, present in the wingbox, the constitutive model explained in 3.2 is used. In TACS, however, a complete and detailed model of this material is not possible, mainly due to the fact that the core is fully enclosed by the CFRP, in contrast with the layup in Fig. 6 and some properties are not exactly representing the real material. These are the major limitations in terms of material that were encountered and a probable cause for different results between a real test and a computational analysis. Nonetheless, the analysis is performed and an optimization can be carried out later, based on this.

Finally, to acknowledge the possible differences between the mass calculated from the structural model and the real mass of each component, measured by the Tekever’s quality control team, the error of the model relative to the measured values is given in Tab. 4.

Table 4: Error of model mass relative to measured mass of AR5 wingbox components

	Lower Skin	Upper Skin	Spars	Ribs	Total
Model to real difference [%]	+19.1*	+26.1*	-30.9	-50.0	-1.7

* Extrapolated value

It is possible to observe that, the calculated total mass is slightly underestimated in the model. Firstly, for the skin mass, an estimation was needed to compare the real and model values since due to the simplifications needed for the wingbox meshing, the skin panels that are computed are only present between the spars. So, with those panel's mass, an extrapolation was made to obtain the estimated weight of the real model skin and it is shown in Tab. 4. This extrapolation was simply made considering the chordwise position of the spars and would be accurate if the skin panels were straight and parallel to the chordwise direction. Since there are a few geometric or material properties' differences, some discrepancies are present in the model. For example, spar's beam profile in the model is simply rectangular, however, in reality, this is not the case. This means that a good amount of material is not considered hence, resulting in a underestimation of the mass in the model. Finally, it is important to note that, overall, the difference is not high (about 2% error) and does not affect the wing's rigidity directly.

4.2 Mesh Convergence Study

Some model simplifications and limitations were encountered Nonetheless, the structure is composed by 6 ribs, 2 spars and the skin panels between these. The simplifications include the trimming of the ribs' leading edge at the front spar, the addition of a rear spar section close to the tip and the skin panels are only present, between the spars (no leading or trailing edge skin panels).

To decide on the mesh refinement level, a mesh convergence study was carried out with 5 meshes, as summarized in Tab. 5. A representation of mesh 1 can be seen in Fig. 7 where the others are a subdivision of this one, where each element is divided by 4 (except number 5 due to a memory error that could not be resolved).

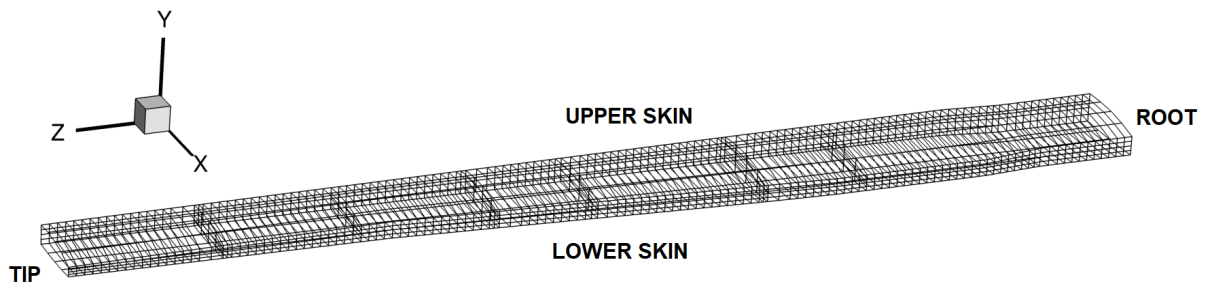


Figure 7: Mesh of the AR5 wingbox

Table 5: Convergence study

Mesh	Number of elements	Degrees of freedom	Max disp. (Tip Rear Spar - USkin) [mm]	Diff.	Stress (Root Front Spar - LSkin) [Pa]	Diff.	RAM [GB]	Time [s]
#1	2,272	13,476	0.274	-	60,490	-	0.1	30
#2	9,088	54,180	0.273	0.15%	62,450	-3.14%	0.3	238
#3	36,352	217,380	0.273	-0.09%	64,280	-2.85%	1.0	378
#4	145,408	870,948	0.266	2.60%	66,089	-2.74%	4.9	410
#5	284,672	1,706,532	0.266	-0.01%	66,753	-1.00%	8.8	466

To choose the most efficient mesh to obtain the structural results from and use during the optimization process, the criteria was: it had to be converged, so a deviation on the monitored values between meshes shouldn't be above one percent; and it should be the one that takes the least time and memory to solve. Looking at the Tab. 5, it was concluded that mesh 4 met all criteria.

As a final regard, these results were obtained with a computer using a 16-core processor with 4.5 GHz base frequency and 128 GB of RAM.

4.3 Design Framework Validation

To validate this numerical design framework, experimental data was used from a static wing bending test provided by Tekever. Weights were added in the lower skin with the wing upside down and the tip displacement measured.

Analysing with TACS, the maximum deflection obtained was about 0.17 m across the tip. The original and deformed shapes can be observed in Fig. 8.

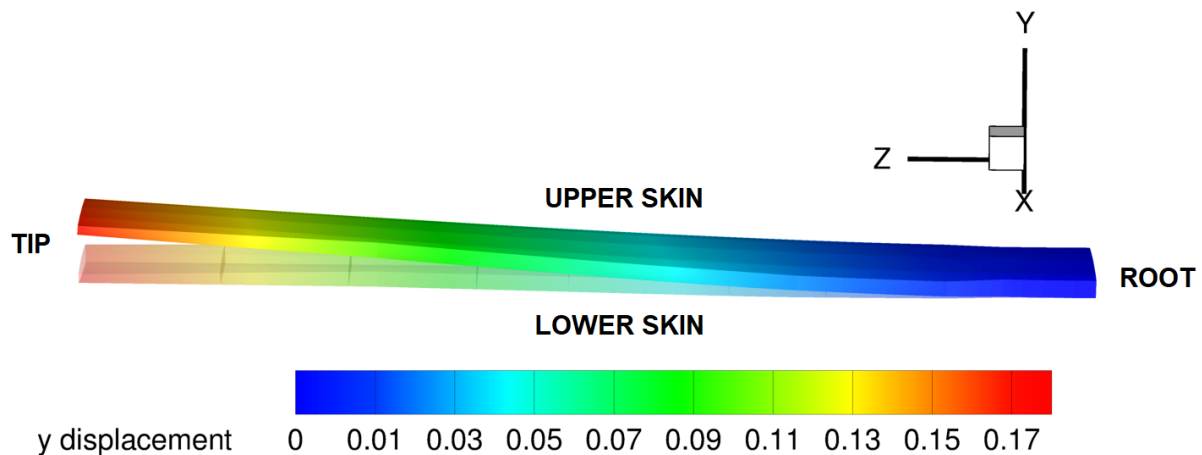


Figure 8: Wingbox's displacement in y under the fixed loads condition

According to the ground test values, the measured displacements, at the tip, for the leading edge and trailing edge were 0.12 m and 0.17 m, respectively. Firstly, the order of magnitude of the computational results is pretty similar which is a good indicator that the model is behaving as expected. Secondly, the fact that there is no significant difference between the leading and trailing edge from the computational results is due to the spar section close to the tip not existing in reality, so the trailing edge has lower stiffness and a higher displacement in the bending test. Finally, the computational results reproduce overall an higher displacement and this is the expected behaviour since, with the geometry approximations explained in Sec. 4.2 to produce the mesh, some wing elements were not added, making the wing less stiff and have a larger deformation. It can be concluded that the framework produced valid and coherent results.

4.4 Analysis of 4g Maneuvre

Since the maximum allowed loads within the flight envelope of the AR5 is a 4.0g manoeuvre, this is the case studied. For this condition, the most important results to gather are the failure index and displacement, to make sure it is an acceptable amount.

The failure index results are shown in Fig. 9. They are satisfactory and well below

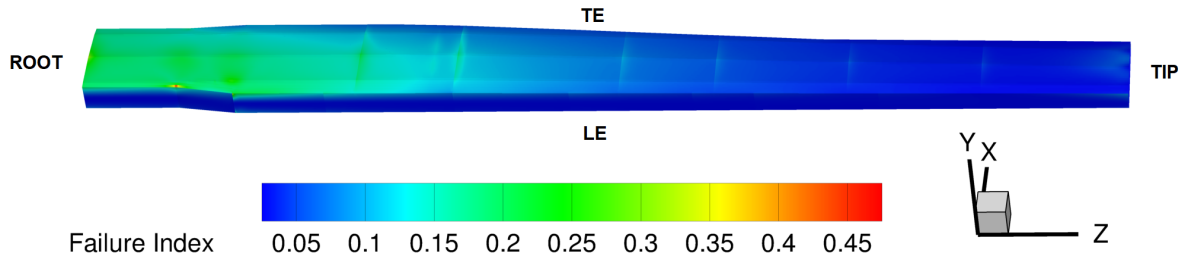


Figure 9: Wingbox failure index under the 4.0g condition

unity, being the maximum observed of 0.47 meaning that the structure has a safety factor of 2.1 and a large margin for improvement.

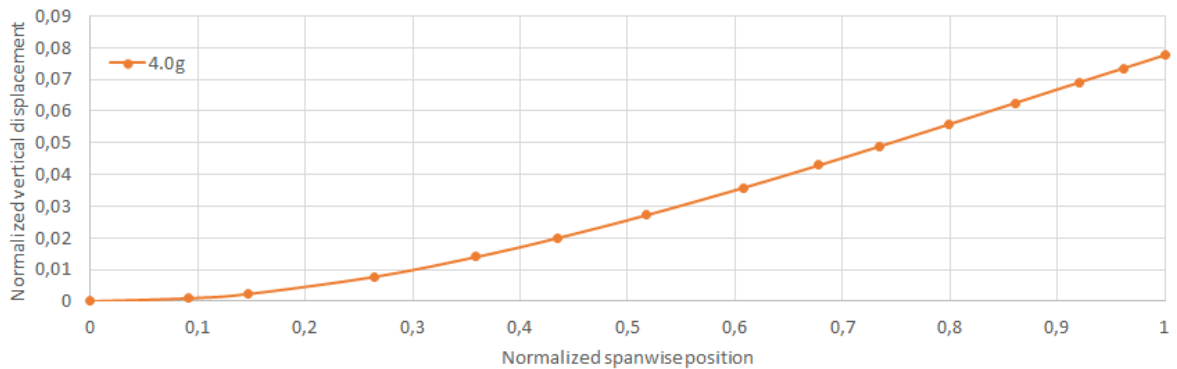


Figure 10: Vertical displacement at the front spar

Regarding the normalized vertical displacement that was plotted across the wingbox front spar in Fig. 10, the maximum obtained at the tip corresponds to about 8% of half span (0.264 m). This indicates that the deflection is not too large relative to the wing size and it is acceptable.

5 WING STRUCTURAL OPTIMIZATION

5.1 Baseline Problem Formulation

Tekever's goal is to minimize the structural mass and increase the efficiency of the AR5, meaning that the objective function to this problem is the total wingbox mass. This includes the ribs, spars and upper and lower skin,

$$m_{\text{total}} = m_{\text{ribs}} + m_{\text{spars}} + m_{\text{upper skin}} + m_{\text{lower skin}} \quad (9)$$

To control this mass, the variables allowed are some of the parameters that describe the wingbox design variable (DV) groups. In this case, the ones chosen are the thickness (t_i) of the i th DV group and the orientation (θ_{ij}) of the j th CFRP ply from the i th DV group. This last design variable does not have a direct impact on the weight but allows the wing to change its stiffness to possibly lower the thickness of the CFRP and still withstand the aerodynamic loads.

Some inequality constraints are set: adjacency constraints, to keep a thickness difference ($|t_i - t_k|$) between a certain i th DV group and its adjacent groups (k th DV groups)

under a certain maximum; failure constraints, in this case, the failure index given in Sec. 3.2 by the Tsai-Wu criterion, to ensure that the wing withstands the loads. Since it is not possible to obtain a gradient of a function that retrieves the maximum failure indexes across all points in which it is calculated, the Kreisselmeier-Steinhauser (KS) aggregation function [24] is used to convert the discrete nature of the maximum failure index function into a continuous and smooth function, allowing for a gradient-based optimizer to use it,

$$\text{KS} = \frac{1}{\rho} \ln \left(\sum_{l=1}^{n_g} \exp(\rho g_l) \right), \quad (10)$$

where ρ is a tolerance/weight parameter, n_g the number of failure constraints and g_l the l th failure constraint function.

With this in mind, the baseline optimization problem for this wingbox structure can be formulated as

$$\begin{aligned} & \text{minimize} && m_{\text{total}} \\ & \text{with respect to} && t_i \\ & && \theta_{ij} \\ & \text{subject to} && \text{KS} \leq 1 \\ & && |t_i - t_k| \leq \Delta_{\text{max}} \end{aligned} \quad (11)$$

According to the manufacturing requirements and methods employed, the boundaries for each DV group is defined as: minimum of 1.0 mm for both skins and spars and 0.25 mm for the spars; to give more freedom to the optimizer to possibly reinforce certain areas of the wingbox, a maximum of 100 mm was set. For the adjacency constraint, Δ_{max} was defined as 5 mm.

The optimization algorithm used was the sequential least squares programming algorithm (SLSQP), briefly presented in Sec. 3.3. Convergence is assumed for the optimal solution when the difference between solutions is smaller than 10^{-6} . To ensure that the optimization process stops if no minimum is found and it does not run indefinitely, a maximum of 500 iterations are allowed.

5.2 Baseline Optimization

Fig. 11 shows the history of the objective and constraints in which the objective function exhibits a converging behaviour, reinforcing the fact that the optimisation found an optimal solution.

Another interesting aspect to note in Fig. 11 is the failure constraint has visible upper spikes that match with the mass lowering spikes, which happens when the optimiser tries to lower the mass, violating the failure constraint.

Fig. 12 compares the initial case with the found optimised solution having the different wingbox components' thicknesses. Across the entire wingbox, the thickness is lower in the optimal solution and, as expected, the higher and lower values are located near root and tip of the wing, respectively.

Regarding the failure index shown in Fig. 13, the optimised wingbox has more regions with a higher failure index, due to the overall decrease in the thickness and the wing being less oversized.

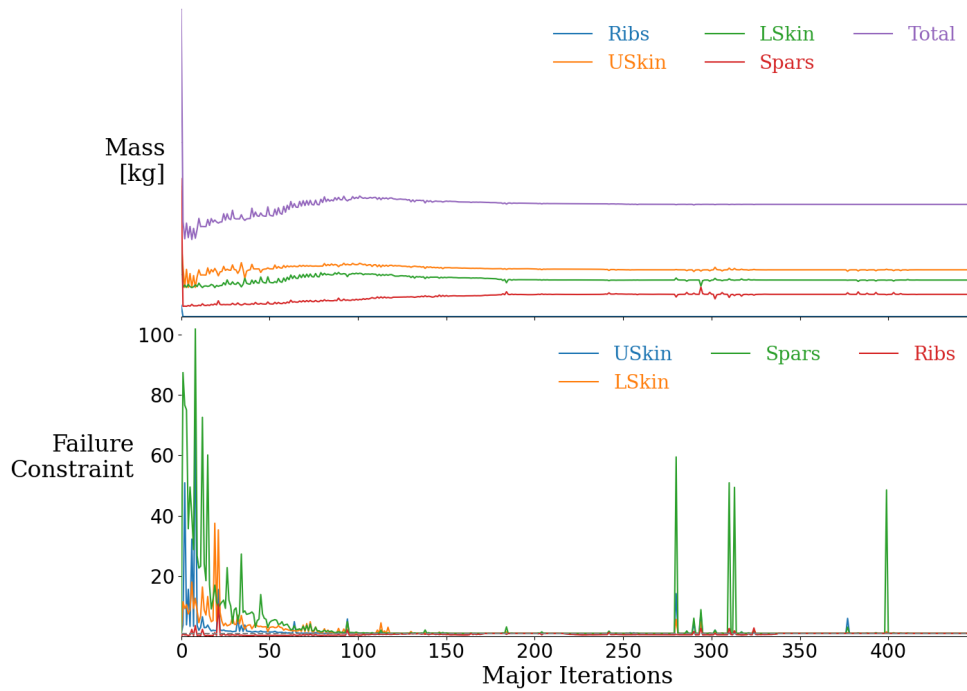


Figure 11: Objective function and failure constraint value across all optimization iterations

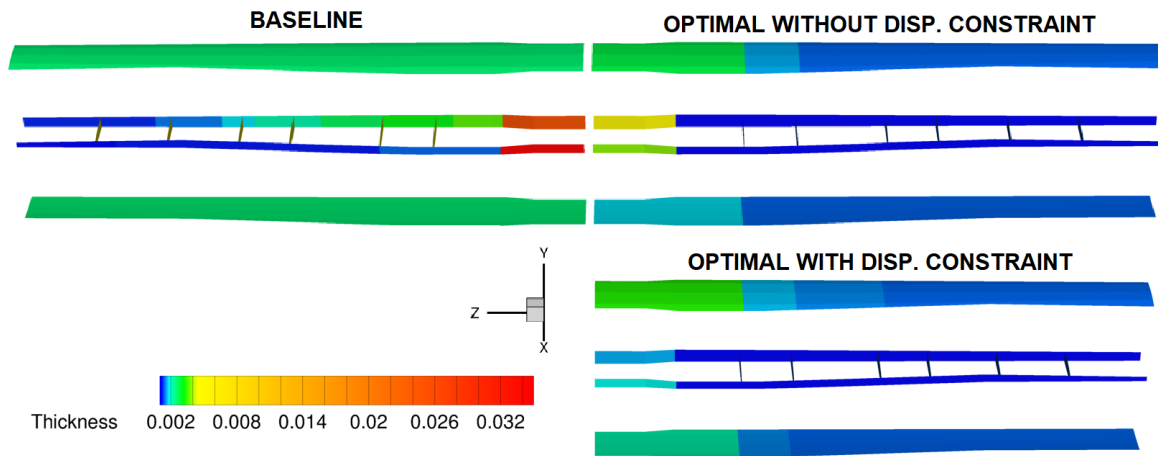


Figure 12: Comparison between initial component thicknesses (left) and optimised solution found (right)

With the thinning of the wingbox components and without buckling constraints present in this optimization, it is expected that the wing displacement increases. In fact, the optimized wingbox, for this 4g load condition, presents a maximum displacement of 0.61 m and its deformation is shown in Fig. 14.

5.3 Baseline Optimization with Displacement Constraint

To prevent such large displacement, another KS function, aggregating all displacements from a specific wingbox region is used, in order to more easily control the maximum displacement of that specific component. Similarly to the failure function, this displacement

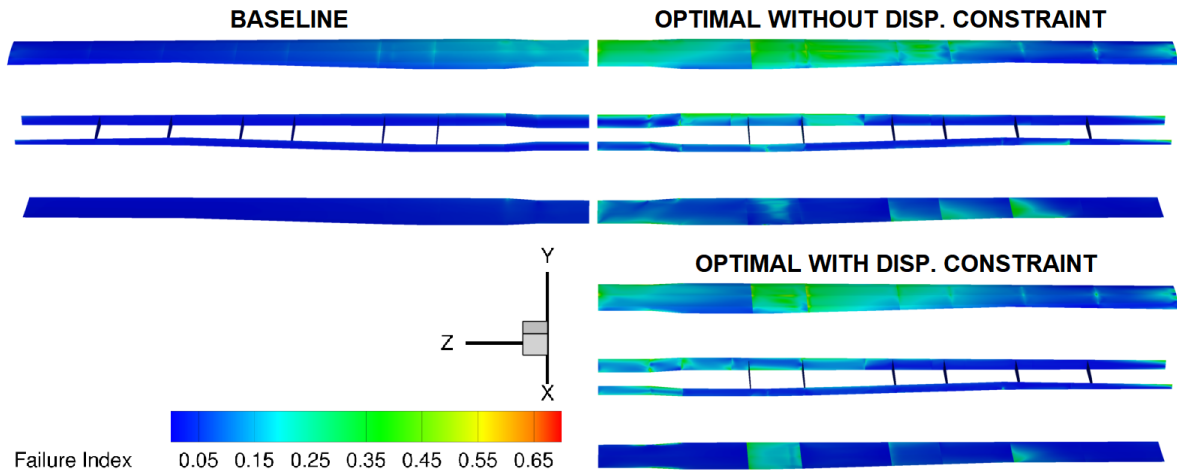


Figure 13: Comparison between initial component thicknesses (left) and optimised solution found (right)

constraint requires a KS function since the maximum function can not be differentiated. In the present case, the tip of the wing is the most problematic location so the constraining KS function uses the displacements in that region.

Having performed the optimization adding the inequality constraint, $KS(\text{disp}) < 0.4 \text{ m}$, a similar plot to Fig. 11 was obtained, confirming convergence of the solution and constraints. By plotting the front spar normalized displacements against the spanwise position (Fig. 14) it is possible to assess that the constraint was successful at lowering the maximum allowed displacement.

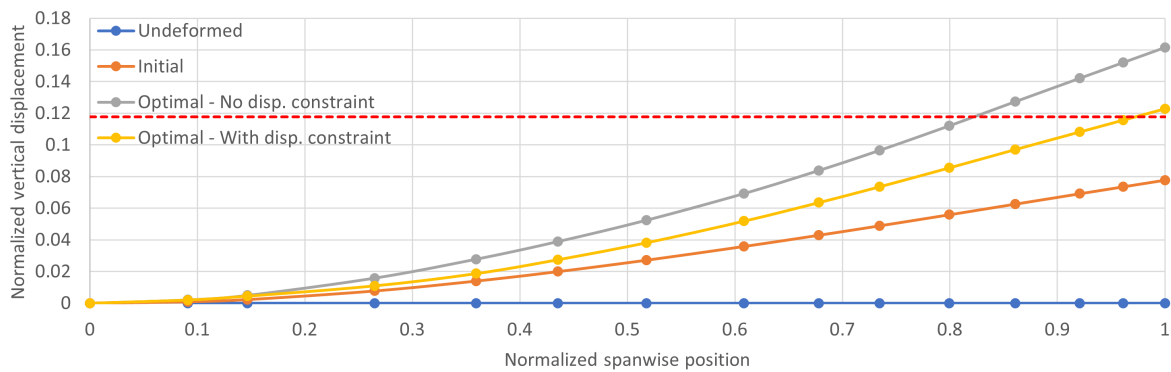


Figure 14: Front spar y deflection under the 4g condition

To ensure this new constraint was respected, it is interesting to see that, comparing the thickness of the components from previous optimal solution and the newer one (Fig. 12), the thickness increased, particularly near the root and for both the upper and lower skins. In contrast, there is a reduction in thickness near the root for both spars. This indicates that the root panels of the skin are more effective at increasing the wing bending stiffness and reducing the maximum displacement and it is possible to mitigate the weight gain by thinning the spars. Although making the spars thinner supposedly decreases their rigidity, the gain from the skin panels is greater and the overall weight is reduced this way.

From Fig. 13 and the information in Tab. 6, it is possible to conclude that the failure inequality constraint remained active, implying that the optimized structure under the 4g condition corresponds to the thinnest possible solution that still withstands the critical aerodynamic loads.

Using the optimal solution obtained from case 1 as the optimization starting point leads to a new solution, which corresponds to a different local minimum.

It is relevant to compare the results against the AR5 baseline analysis quantitatively, to better understand the enhancements these optimizations provide to the AR5 wing. In Tab. 6 is a summary of these results.

Table 6: Summary of relevant optimization results

	Mass difference	Maximum disp. [m]	Maximum failure index
Baseline	0%	0.273	0.47
Optimal - No disp. constraint	-55.6%	0.613	0.79
Optimal - With disp. constraint	-52.4%	0.473	0.74
Optimal - Different baseline	-54.4%	0.471	0.58

It is possible to conclude that these optimizations provide a mass reduction larger than 50% of the original baseline. As expected, with the displacement constraint, the reduction is smaller since, to increase the rigidity of the wing to respect the constraint, thicker sections were necessary. The third optimization case revealed to be important since by having the starting point as the optimal solution from the first optimization case, a better result could be found keeping the displacement constraint. The maximum displacement for the third solution was about 12% of the wing's half span, similar to the results in Fig. 14 for the second case, which is considered acceptable. Finally, regarding the possible wingbox failure, a safety factor of 1.5 was used so that all optimizations would have a maximum failure index of about $\frac{1}{1.5} = 0.67$. The third optimization is well within the limit having a maximum failure index of 0.58 and a 1.78 safety factor.

A full description of the final values of the design variables for each wingbox DV group is present in Tab. 7 for the three optimization cases (Case 1/Case 2/Case 3). The 0° ply orientation is defined by direction 1 of the local reference axis for each wingbox component: for both skins, spars and ribs, it corresponds to the z , y and x directions in Fig. 7, respectively.

It can be concluded that, across the span, the thickness decreases except for the rear spar although it is not too much (about 0.1 mm more than the previous panel). Furthermore, although the local minima produced a similar displacement response, it has a very distinct composition in terms of ply orientation, allowing the wingbox to satisfy the constraints with a higher safety factor (lower failure index).

Regarding the ribs, these are the parts that suffer the most thickness changes since their baseline value is 6 mm and, across all optimizations, the value is reduced to the minimum 1 mm. This might indicate that the baseline value is greatly oversized. Although it might be true, it cannot be concluded since no buckling constraints were added and the wingbox ribs main function is to increase the buckling strength.

One final note about the plies orientation and a main concern and physical constraint, is that if this wing was to be manufactured, since the angles of the different DV groups have large disparities between them, it would not be possible to manufacture it with unidirectional fiber cloths. Using them, would lead to non continuity of the fibers along

Vítor M. T. Silva, Nuno M. B. Matos and André C. Marta

Table 7: Design variables’ values of the optimal solutions

Upper Skin	1 (Root)	2	3	4	5	6	7 (Tip)
t [mm]	2.8/3.3/3.0	1.4/1.4/1.3	1.0/1.2/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0
θ_1 [°]	-1.1/-7.8/-3.6	-1.4/-5.2/-2.6	-4.8/-2.4/-2.4	-20.1/-1.9/-3.6	3.0/-2.7/-1.9	-1.7/-1.7/-0.1	9.5/7.9/6.0
θ_2 [°]	90.0/19.6/90.0	90.0/27.8/90.0	32.8/90.0/-2.5	24.3/90.0/0.1	87.8/90.0/-0.6	88.1/90.0/90.0	87.9/90.0/90.0

Lower Skin	1 (Root)	2	3	4	5	6	7 (Tip)
t [mm]	1.7/2.2/2.1	1.0/1.2/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0
θ_1 [°]	-1.8/-5.7/-5.3	-14.4/3.2/-16.3	-4.1/-8.5/10.3	7.9/1.3/3.5	16.6/0.7/6.3	43.2/30.5/-21.4	52.5/58.4/62.8
θ_2 [°]	56.8/14.0/18.0	22.0/90.0/19.6	20.8/13.8/-4.6	84.9/89.7/-0.3	89.9/77.4/-4.3	90.0/90.0/19.8	52.5/58.3/63.2

Ribs	1 (Closer to root)	2	3	4	5	6 (Closer to tip)
t [mm]	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0	1.0/1.0/1.0
θ_1 [°]	-17.3/-55.6/-77.0	-20.2/3.4/13.8	-20.7/6.6/7.2	-0.5/0.0/-0.2	-5.5/-4.2/-26.6	4.4/-1.6/-24.0
θ_2 [°]	36.4/75.5/45.9	16.3/90.0/10.1	80.1/89.4/82.5	89.8/90.0/90.0	88.4/88.9/86.1	90.0/89.4/82.9

Front Spar	1 (Root)	2	3	4	5	6	7	8 (Tip)
t [mm]	5.1/1.4/3.0	0.4/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.3
θ_1 [°]	20.8/-11.1/1.7	20.7/-61.1/-30.6	10.5/-23.8/-70.4	-40.1/-3.1/-68.9	9.8/-1.8/0.4	-0.5/-2.5/-37.2	-0.3/-6.3/-62.3	0.8/1.4/1.7
θ_2 [°]	90.0/86.5/90.0	90.0/5.3/27.3	90.0/10.8/5.7	90.0/90.0/7.9	61.3/68.4/73.1	77.1/80.9/78.4	74.8/79.8/72.6	83.0/81.1/78.4

Rear Spar	1 (Root)	2	3	4 (Tip)
t [mm]	3.9/1.9/3.3	0.3/0.3/0.3	0.3/0.3/0.3	0.3/0.3/0.4
θ_1 [°]	-26.7/-28.6/-11.3	5.8/-26.9/-61.0	-9.6/-72.8/-69.9	-19.1/-18.2/-11.7
θ_2 [°]	72.9/35.6/45.0	90.0/40.7/40.9	43.3/58.4/75.5	89.3/88.2/86.3

the span which is extremely non desirable, as they create weak points. This means that, alternative methods of manufacturing would be needed or the addition of constraints to make sure the ply angles would be the same across the groups from the different wingbox components.

6 CONCLUSIONS

A alternative wingbox structural solution for the AR5 wing was obtained, leading to an expected reduction of weight of about 50% while maintaining its structural rigidity, with a 1.5 safety factor. It is also important to note that, with further refinement of the computational model, both geometrically, with a more detailed mesh, and in terms of material characterization, with closer to reality mechanical properties, this methodology and high-fidelity framework proved to be a powerful tool to create optimal structural solutions.

One of the major drawbacks from the results presented in this work is the fact that it is not feasible to manufacture the optimal wingbox by traditional means, such as manual layup, since there is a large disparities between ply angles of adjacent DV groups. However, to manufacture this optimal solution, tow-steering technology could be used. The other major limitation from the obtained results is the fact that no buckling constraints were included.

As future work, it is suggested the implementation in this framework of constraints that impose continuity of the DV groups’ ply angles across the wingbox components (each skin and spar), as well as a new or improved material constitutive law to substitute the smeared approach, that would allow more control of the layup as well as the use of buckling constraints.

ACKNOWLEDGEMENTS

The authors would like to thank Tekever for supporting the production of this article by offering all the needed data to successfully produce the analyses here shown. This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] S. Gallardo-Saavedra, L. Hernández-Callejo, and O. Duque-Perez. Technological review of the instrumentation used in aerial thermographic inspection of photovoltaic plants. *Renewable and Sustainable Energy Reviews*, 93:566–579, 2018. ISSN 1364-0321. doi:10.1016/j.rser.2018.05.027.
- [2] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis. A review on uav-based applications for precision agriculture. *Information*, 10(11), 2019. ISSN 2078-2489. doi:10.3390/info10110349.
- [3] H. Yao, R. Qin, and X. Chen. Unmanned aerial vehicle for remote sensing applications—a review. *Remote Sensing*, 11(12), 2019. ISSN 2072-4292. doi:10.3390/rs11121443.
- [4] P. Cohn, A. Green, M. Langstaff, and M. Roller. Commercial drones are here: The future of unmanned aerial systems. *McKinsey & Company*, 2017. URL <https://www.mckinsey.com/industries/travel-logistics-and-infrastructure/our-insights/commercial-drones-are-here-the-future-of-unmanned-aerial-systems>.
- [5] Tekever official website - AR5, May 2023. URL <https://www.tekever.com/models/ar5/>.
- [6] D. Grose. Reengineering the aircraft design process. In *5th Symposium on Multidisciplinary Analysis and Optimization*, Panama City Beach, FL, U.S.A., September 1994.
- [7] K. K. Rumayshah, A. Prayoga, and M. A. Moelyadi. Design of high altitude long endurance uav: Structural analysis of composite wing using finite element method. *Journal of Physics: Conference Series*, 1005(1):012025, April 2018. doi:10.1088/1742-6596/1005/1/012025.
- [8] G. J. Kennedy and J. R. Martins. A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elements in Analysis and Design*, 87:56–73, 2014. doi:10.1016/j.finl.2014.04.011.
- [9] J. N. Reddy. *Mechanics of laminated composite plates and shells: theory and analysis*. CRC Press, 2003.
- [10] T. J. Hughes and F. Brezzi. On drilling degrees of freedom. *Computer Methods in Applied Mechanics and Engineering*, 72(1):105–121, 1989. doi:10.1016/0045-7825(89)90124-2.

- [11] D. Fox and J. Simo. A drill rotation formulation for geometrically exact shells. *Computer Methods in Applied Mechanics and Engineering*, 98(3):329–343, 1992. doi:10.1016/0045-7825(92)90002-2.
- [12] J. R. R. A. Martins, G. Kennedy, and G. K. Kenway. High aspect ratio wing design: Optimal aerostructural tradeoffs for the next generation of materials. In *52nd Aerospace Sciences Meeting*, National Harbor, Maryland, January 2014. doi:10.2514/6.2014-0596.
- [13] G. Kennedy and J. R. R. A. Martins. A comparison of metallic and composite aircraft wings using aerostructural design optimization. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, Indiana.
- [14] G. K. W. Kenway and J. R. R. A. Martins. Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. *Journal of Aircraft*, 51(1):144–160, 2014. doi:10.2514/1.C032150.
- [15] N. Werter and R. De Breuker. A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation. *Composite Structures*, 158:369–386, 2016. doi:10.1016/j.compstruct.2016.09.044.
- [16] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. Martins. pyoptspase: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. doi:10.21105/joss.02564.
- [17] J. R. R. A. Martins and A. Ning. *Engineering Design Optimization*. Cambridge University Press, 2021. doi:10.1017/9781108980647.
- [18] Z. Lyu, Z. Xu, and J. R. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *The 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, July 2014. ICCFD8-2014-0203.
- [19] L. Zhou, J. Huang, Z. Gao, and W. Zhang. Three-dimensional aerodynamic/stealth optimization based on adjoint sensitivity analysis for scattering problem. *AIAA Journal*, 58(6):2702–2715, 2020. doi:10.2514/1.J059136.
- [20] G. Yang and A. D. Ronch. Aerodynamic shape optimisation of benchmark problems using su2. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, January 2018. doi:10.2514/6.2018-0412.
- [21] I. Gibert Martínez, F. Afonso, S. Rodrigues, and F. Lau. A sequential approach for aerodynamic shape optimization with topology optimization of airfoils. *Mathematical and Computational Applications*, 26(2), 2021. ISSN 2297-8747. doi:10.3390/mca26020034.
- [22] A. El Ibrahim, S. Abdulkarim, and I. Göv. Aero structural optimization for sailplane wing in preliminary design. *Journal of Advances in Technology and Engineering Research*, 4, 02 2018. doi:10.20474/jater-4.1.5.

- [23] R. T. Haftka. Simultaneous analysis and design. *AIAA Journal*, 23(7):1099–1103, 1985. doi:10.2514/3.9043.
- [24] S. Sethi, A. Striz, S. Sethi, and A. Striz. On using the kreisselmeier-steinhauser function in simultaneous analysis and design. In *38th Structures, Structural Dynamics, and Materials Conference*, Kissimmee, FL, U.S.A., April 1997. doi:10.2514/6.1997-1289.



WING AERODYNAMIC DESIGN FOR A MAME UAV USING HIGH-FIDELITY NUMERICAL TOOLS

Rúben S. Gameiro¹, Nuno M. B. Matos² and André C. Marta¹

1: IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
{rubengameiro, andre.marta}@tecnico.ulisboa.pt, <https://mdo.tecnico.ulisboa.pt>

2: Research and Development
Tekever UAS
Tekever
2500-750 Caldas da Rainha
nuno.matos@tekever.com, <https://tekever.com>

Abstract. *The UAV market is currently very competitive, with the frequent launch of new products and a wide range of solutions already available, forcing manufacturers to explore the design space faster and more efficiently than in the past. A cost effective approach is to develop growth versions, improving an existing product with new technologies and design tools. Some of these tools include RANS based high fidelity computational fluid dynamics methods and discrete adjoint gradient-based optimization, which will be used in this work on a numerical design framework to explore the aerodynamic shape optimization of a wing, as part of the development of a growth version of a MAME UAV for a leading Portuguese manufacturer. A comprehensive aerodynamic analysis of the current UAV wing will be performed, followed by an optimization procedure to minimize drag subject to a prescribed lift coefficient constraint. To that end, two different starting geometries will be considered and parameterized with common design variables, including twist and chord distribution, sweep and airfoil shape. New optimized geometries for different sets of design variables will be obtained with a significant drag coefficient reduction from the starting geometry. The optimized geometries will approach an elliptical lift distribution, although not exactly considering the trade-offs needed between skin friction and induced drag. Despite the fact that the results obtained here are not considered the final design, as more shape parametrizations and design variables are yet to be explored, they provided a good insight on how the different parametrizations are handled by the design optimization framework and considerations that should be taken.*

Keywords: Optimization, adjoint method, aircraft design, computational fluid dynamics, free-form deformation, geometric parametrization

1 INTRODUCTION

UAVs have been around for more than a century [1]. Initially, developments went hand in hand with the recent aviation industry, following a technology-driven approach which allowed a fast evolution, benefiting from the emerging technologies in all disciplines (aerodynamics, structures, flight mechanics, etc.) [2].

As speeds reached by manned aircraft started to increase significantly during the 60's and the 70's, more focus started to be given to aerodynamic design, starting with airfoil design and quickly evolving to wing design [3]. At this time, CFD was used primarily to obtain a better mission performance by increasing range and speed. However, with the oil crisis in the 70's, this old approach to aircraft design started to shift into a customer driven approach which is still maintained nowadays, meaning that aircraft design started to be a balance between available technology and the benefits it would bring to the customer, taking into account the required constraints [4]. Satisfying the customer needs also implied a reduction in development time so that the manufacturer could respond in time to the current market needs.

During this shift, CFD underwent very significant advances, with the solution of the viscous flow around a 3D geometry of the whole aircraft using RANS equations and turbulent models eventually becoming standard [3]. With the evolution in computational power and code scalability, it became feasible to use such models on the conceptual design phase [5–7]. CFD brought several advantages to the design process, making it much more efficient. However the project of a new vehicle can still take years, with several iterations between the different disciplines, such as structures, aerodynamics and propulsion, often on an intuition and trial and error basis to assess different possible vehicle shapes [3].

Nowadays, the UAV market is experiencing fast growth and dynamic developments, which naturally attracts investors attention. Notably, the industry received a total investment of *US*\$6.96 billion last year [8]. This growing market has also become highly competitive, with the influx of new market players resulting from the substantial investments, companies seeking external funding must differentiate themselves by offering innovative and unique products.



Figure 1: Tekever AR5 UAV. Source: Tekever.

In this context, this work explores automatic aerodynamic shape optimization using high fidelity methods, in particular the MACH-Aero framework, developed by the MDOLab [9]. The focus is on the development of a growth version of the Tekever AR5 UAV, shown in Figure 1. Several different parametrizations will be tested and associated with common design variables which, together with a prescribed lift coefficient constraint

and a drag coefficient reduction as an objective, will form the optimization problem in study. For that, the aerodynamic analysis tools associated with the framework are used first to perform an analysis of the current AR5 wing. After that, optimization problems with different sets of design variables will be set up and the results analyzed and commented.

In the future, this project will evolve to a multidisciplinary optimization (MDO) framework, where the aerodynamic and structural optimization will be performed together.

2 FRAMEWORK OVERVIEW

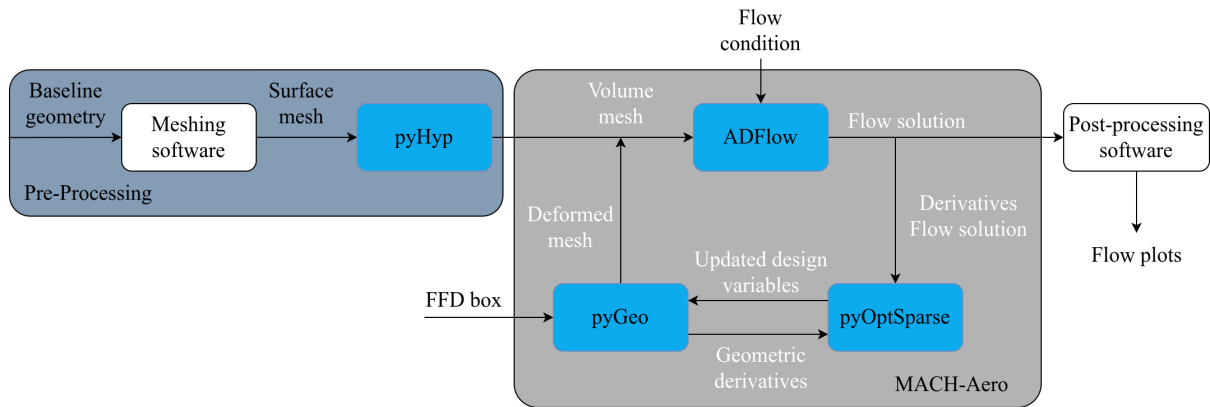


Figure 2: MACH-Aero optimization framework.

Figure 2 schematizes the process followed to optimize the aerodynamic shape of a wing. The blue shaded boxes represent the modules developed by the MDOLab and the gray box encompasses the MACH-Aero framework. The white text represents parameters that are automatically updated by the MACH-Aero framework, and the black text represents parameters that require user input.

The MACH-Framework is composed by the following modules: *ADFlow*, a structured multi-block 3D CFD solver [10] which also solves the adjoint method to compute the derivatives [11], *pyGeo* a geometry manipulation tool that was specifically built for multidisciplinary optimization applications and allows the manipulation, parametrization and constraint handling of the geometric shape [12] and *pyOptSparse*, which handles the constrained nonlinear optimization problem [13].

2.1 Meshing

The first step is to create a baseline geometry which will serve as a starting point for the optimizer. Considering that this geometry will be an input to the MACH-Aero framework, two main options are available for generating the geometry: Using *pyGeo*, that allows to create simple geometries by lofting surfaces between a provided set of airfoils in a code-based way, or using a typical CAD (Computer Aided Design) or geometry generation software, like OpenVSP [14], which was the main tool used in this work to obtain the starting point for the optimizer. The geometry should then be exported to a neutral CAD format that is accepted by the mesh generator. IGES (International Graphics Exchange Standards) or STEP (Standard for the Exchange of Product Data) are two formats widely accepted by most meshing software.

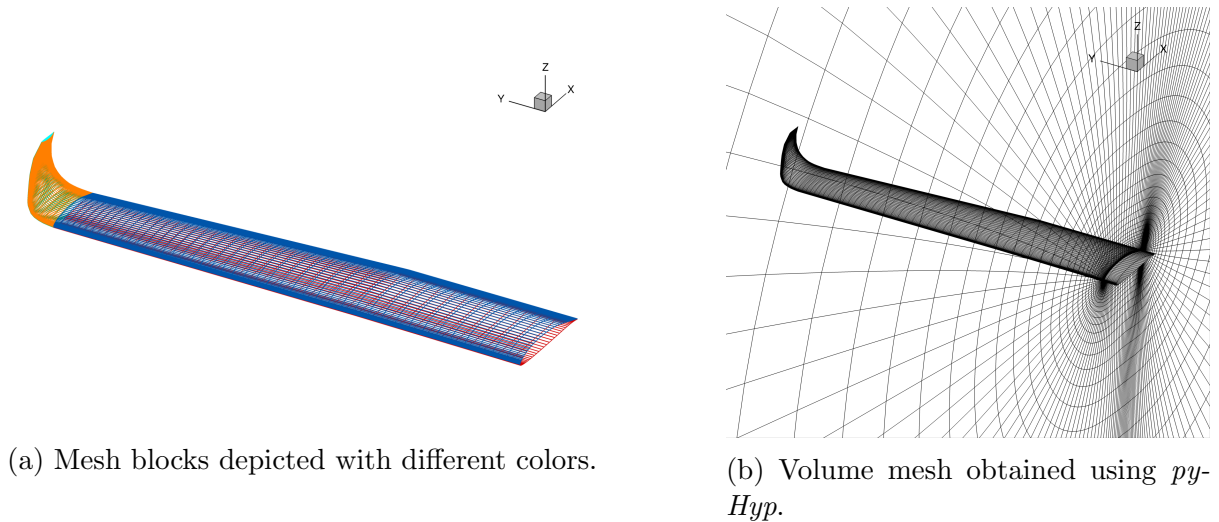


Figure 3: Used mesh for the Tekever AR5 wing.

From the geometry CAD file, a surface mesh must be obtained. For that, it is possible to choose from a variety of different software. However, the chosen grid generator should be able to generate a multi-block and/or overset 2D mesh, as those are the grid types accepted by the flow solver, and output the resulting grid to a CGNS (CFD General Notation System) file, in ADF [15] or HDF5 format, supported in more recent versions of CGNS [16].

The most popular options are ICEM CFD[®] and Pointwise[®]. Some open source solutions for mesh generation, such as GMSH [17] are also widely available. However, the first requirement limits their usage in this work. Figure 3a shows an example of a surface mesh with multiple blocks used in this work.

Having a surface mesh in the appropriate format, an hyperbolic volume mesh is then extruded using *pyHyp*, an hyperbolic mesh generator that also automatically applies the necessary boundary conditions for a wing if desired [18]. Figure 3b shows a volume mesh obtained with *pyGeo*, as well as the respective surface mesh. To generate a volume mesh from a surface mesh, *pyHyp* requires the user to provide a few parameters: the first layer height, the total height of the volume mesh and the number of layers to extrude. Both the surface and the volume mesh should, naturally, be sufficiently refined, and so, grid independence studies must be carried out before starting the aerodynamic analysis and optimization procedure.

2.2 Flow model

To compute the flow, *ADFlow* includes both inviscid models (Euler) and viscous models, which solve the RANS equations with different turbulence models available, including Spalart-Allmaras, Wilcox $k-\omega$, $k-\tau$, Menter SST $k-\omega$ and $v2-f$. In this work, it is intended to optimize the wing for viscous flow, so RANS models will be used.

Regarding the turbulence model, comparisons for external flows around an Onera M6 wing concluded that the Spalart-Allmaras and SST $k-\omega$ turbulence models gave the closest results to the experimental data [19] and using the DPW7 CRM configuration shown that both the $k-\omega$ and Spalart-Allmaras models yield great results [20]. The main drawback of the Spalart-Allmaras model is for separated flows, performing well for external flows

otherwise [21].

The flow solution using *ADFlow* with Spalart-Allmaras turbulence model was also compared with two other solvers and to experimental data for the CRM wing, and successful validation of the *ADFlow* code was achieved [22].

It is important to note that only the Spalart-Allmaras turbulence model has been differentiated in the *ADFlow* code. This fact, together with previous observations of the good results obtained with Spalart-Allmaras for external flow computations makes it the most suited for the optimization of a wing. Spalart-Allmaras is a linear eddy viscosity model, and thus, uses Boussinesq assumption for the constitutive relation [23]. Furthermore, it is considered a low Reynolds number, and in [24] it was concluded that an $y+$ value of approximately 1 was enough to obtain numerical uncertainties of friction resistance coefficients smaller than 1%.

The flow conditions are defined and passed to *ADFlow* by instantiating the *AeroProblem* class from *baseclasses*. Multiple formulations could be used, but for this work, the flow condition will be defined using the angle of attack, α , flight altitude, h , and the Mach number, Ma of the undisturbed flow. From these parameters the complete flow state information is obtained from the 1976 U.S. Standard Atmosphere [25]. Furthermore, wing reference area and chord should be passed as inputs to the *AeroProblem* for the computation of the aerodynamic coefficients.

2.3 Geometric parameterization

Using the MACH-Aero framework, there are two main options for geometry parametrization: CAD-based and free-form deformation based, which uses a box that completely embeds the surface mesh and that will be referred in this work simply as the FFD box. The surface mesh nodes are then mapped to the FFD box with an $\mathcal{R}^3 \rightarrow \mathcal{R}^3$ mapping, determined by performing a Newton search. After that, surface mesh nodes can be deformed by performing deformations on the FFD box nodes, given their mapping. The surface mesh deformations are then used to perturb the volume mesh using an hybrid algebraic-linear-elasticity mesh perturbation scheme [12] due to its resulting high grid quality and low computational effort. Figure 4 shows the result of using such scheme to deform the surface mesh of a wing, where it can be seen that the deformation of the FFD box deformed the embedded wing.

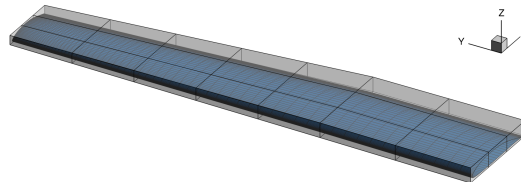


Figure 4: FFD box used to deform a wing surface mesh.

CAD-based approaches parameterize the geometry directly and so the design variables for the optimizer are directly related to parametric CAD variables, which also makes it easier to use the optimized geometry. However, the FFD approach is generally easier to setup and use with an already existing geometry and has freedom to parametrize multiple design variables easily, so it was the chosen approach.

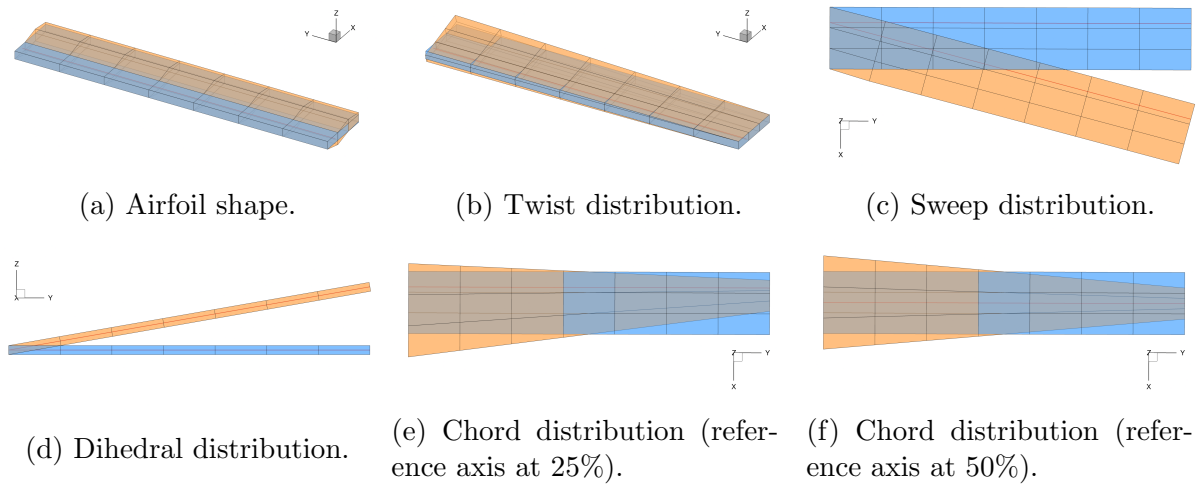


Figure 5: Deformations to the FFD box associated with common design variables.

The nodes of the FFD box may be displaced individually in any spatial direction, and will, in that case, be known as local design variables. However, *pyGeo* also allows to create relations between node displacements in such a way that they are all affected by a smaller set of design variables, allowing the geometry parameterization with global design variables. This second approach is useful as the reduced number of design variables saves computational resources during the optimization whilst allowing for the creation of functions that can represent more intuitive deformations such as taper or twist. An example of a local design variable with vertical displacements is given in Figure 5a where the parameterization was set up such that all sections of the FFD box would mimic the same deformations.

To create a global design variable using *pyGeo*, in addition to the FFD box, an axis is necessary. The axis can be created by specifying its direction and relative position in the FFD box using the fraction of two specified directions. For a typical wing optimization problem, the user should specify the direction that follows the wingspan, and the fraction of the vertical and streamwise position of the axis, which will be, for most practical cases, 0.5 and 0.25 respectively, representing the quarter chord of the wing. It is important to note that the wing should be centered within the FFD box. The reference axis will be used to project all the FFD nodes into it using an user defined direction for the projection. The points will then become rigidly linked to the reference axis, and so, any deformation on the FFD axis will have an effect on several FFD points at once [26]. A deformation on the FFD axis may be of three types: displacement, rotation and scaling.

The first type of deformation is the displacement of the points to a new location, which may be used to create dihedral or sweep, for example, as Figures 5d and 5c show. *pyGeo* also allows rotations and scaling in any one of the three directions x , y and z . The most obvious application of a reference axis rotation would be to create a twist distribution, depicted in Figure 5b, but it is also useful to keep the airfoil sections perpendicular to the reference axis and, thus, effectively keeping the same airfoil by applying rotations to the reference axis points across the axis of rotation. This effect can be clearly visible in Figures 5d and 5c. The scaling function can scale the FFD box points in a specified direction. Figures 5e and 5f show an example of scaling functions being applied to create a chord distribution. Besides the obvious scaling in the streamwise direction, a scaling

in the vertical direction was also applied to maintain the airfoil thickness to chord ratio constant. While Figure 5e shows a chord distribution for a reference axis at 25% of the FFD box, Figure 5f shows a chord distribution for a reference axis at 50%.

The design variables are not necessarily defined by a single number and can have as many degrees of freedom as the number of points in the reference axis. It is even possible to parametrize distributions of the variables with smaller sets of variables: A linear distribution of the chord or twist can be parametrized with only the slope as the design variable.

2.4 Constraints

pyGeo can also be used to set geometric constraints. It is possible to add different types of constraints, being the most common ones the minimum thickness constraints, to ensure that there is enough room for structural components; the minimum volume constraints, to guarantee enough internal volume to carry a specified amount of fuel; the curvature constraints, typically used to ensure manufacturability; and LeTe constraints to avoid shearing twist at the leading and trailing edge when local design variables are present.

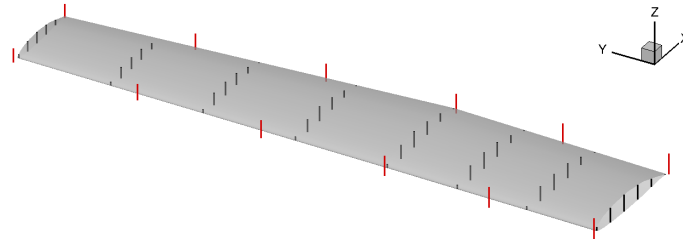


Figure 6: Minimum thickness and LeTe constraints defined using *pyGeo*.

Minimum thickness and volume constraints may be specified with either an absolute value or a relative value in relation to the baseline design. Figure 6 shows an example of the used minimum thickness (represented in black) and LeTe (shown in red) constraints. They are only enforced at certain user-specified locations and are independent of the FFD nodes. When the geometry is deformed, the change of location of those constraints is also handled automatically by *pyGeo*, which also takes care of the needed operations to use them as constraints in the optimization problem.

In this work, only minimum thickness constraints and LeTe constraints will be used. Minimum volume constraints are not required, considering that the UAV in study does not store the fuel in its wings.

2.5 Optimization problem

With both constraints and design variables defined in *pyGeo*, they need to be passed to *pyOptSparse* for the optimization process. Variables from the aero problem, defined using the *baseClasses* can also be passed as optimization variables and a particularly important example of that is the angle of attack. When adding the angle of attack as a design variable, attention must be paid to its relation with twist, given that a geometric rotation of the root wing section has an equivalent effect to the angle of attack. For this

reason, twist at the root is usually kept fixed and not added as a design variable. This also ensures that the shape of the wing mounted at the fuselage is conserved, and thus, the same principle may be applied to the other design variables.

Non-geometric constraints can also be created by user defined functions. The most important non-geometric constraint when optimizing for cruise flight is the lift constraint, which should equal a prescribed lift coefficient at cruise condition and ensure that it is possible to sustain flight. In a similar way, it is possible to define the function or combination of functions to be used as the optimization objective. In this case, the optimization objective will be the drag coefficient minimization.

With the design variables, usually defined by *pyGeo* parametrizations, constraints from both the geometry and the AeroProblem and the objective function passed to the optimization problem, the optimization set-up is effectively complete and ready to run. Given that *pyOptSparse* sets up the optimization in a problem oriented manner, a problem statement in the standard form is easily translated to the optimization framework.

When running, the CFD solver will compute the solution for the baseline geometry and the adjoints, which will then be fed back to the optimizer. The optimizer will use this information to update the value of every design variable and fed them back to *pyGeo*, which will deform the volume mesh as described before. The new volume mesh will be solved by *ADFlow* again, and when performing multiple function evaluations in the same optimization iteration, the solver will take the final solution of the last call as a starting point, speeding up iterations. This process will be repeated until convergence is achieved.

pyOptSparse can use different optimizers, including both gradient based and gradient-free ones. Both gradient-based and gradient-free optimizers have been used in the past for aerodynamic shape optimization. Comparative studies of both methods were performed in [27] and [28], with similar conclusions between them: Both can converge to a solution, but gradient-free methods are considerably more computationally expensive, which the difference increasing dramatically as the number of design variables increases. Examples of gradient-based have been performed where both refinement and exploratory studies were done using both inviscid and viscous models for the optimization of a wing using the MACH-Framework and the SNOPT optimizer and where a more complex problem of multi-component aerodynamic optimization for a wing propeller coupling was also solved using gradient-based methods within the OpenMDAO/MPhys framework [29, 30].

Although in this work a smaller set of design variables will be used initially, the number will be increased later stages, and so, the gradient-based methods were the best option. Furthermore, the functions to be used are smooth and C_1 -continuous, as the previously presented papers have shown.

The SLSQP (Sequential Least Squares Quadratic Programming) was used due to its open source nature and robustness. It has been used in the past for aerodynamic shape optimization [31, 32]. The SLSQP algorithm solves constrained nonlinear optimization problems, which corresponds to the nature of the problem being studied in this work, using the Han-Powell quasi-Newton method with BFGS update of the B-matrix and an L1-test function in the step length algorithm [33].

To compute the gradients needed for the optimization algorithm, given that in aerodynamic shape optimization problems the outputs are often reduced to the drag coefficient but multiple design variables are used as inputs, the adjoint method is the most suitable option for aerodynamic shape optimization and is the one used in the MACH-Aero

framework.

3 INITIAL GEOMETRY ANALYSIS

3.1 Operating conditions

As stated in Section 1, the Tekever AR5 wing is to be optimized. For this reason, the first step is to assess the performance of the current. Its cruise conditions are summarized in Table 1.

Table 1: Performance parameters of Tekever AR5 [34].

Cruise speed	U_∞	100km/h
Cruise altitude	h	1000ft
Maximum Take-off Weight	$MTOW$	180kg
Endurance	E	12h

The analysis should be performed for a specific angle of attack, which can be determined by optimization with a single design variable and a prescribed lift coefficient as constraint or with a secant method provided by *ADFlow* to find the initial angle of attack for a certain lift coefficient.

Tekever provided the wing lift coefficient $C_{L_{wing}} = 0.8932$, considering the projected area of the AR5 wing, $S_{wing} = 2.1691m^2$, and the cruise conditions presented in Table 1.

Despite being automatically calculated by the *AeroProblem* class, it is important to estimate the Reynolds number to understand what kind of simulation will be performed and choose the models adequately. Reynolds number is given by

$$Re = \frac{\rho U_\infty c_{MAC}}{\mu}, \quad (1)$$

where c_{MAC} is the mean aerodynamic chord, U_∞ is the free-stream velocity, defined by the cruise speed (Table 1) and air density, ρ and dynamic viscosity, μ , which can be computed for the cruise altitude using the ISA Standard atmosphere. Using all the described values, it is possible to obtain a Reynolds number of $1.1e6$. This is a moderate Reynolds number, and it indicates that turbulent flow will be dominant. The effects of viscous forces will be much smaller than the ones caused by pressure forces, but they should not be ignored.

Mach number is another dimensionless parameter that should be calculated, not only because it provides an insight into the kind of flow to be expected, but also because it is an input for the *AeroProblem*. It is given by

$$M = \frac{U_\infty}{\sqrt{\gamma RT}}, \quad (2)$$

where γ is the adiabatic constant, R is the gas constant and T is the temperature, calculated also with the ISA Standard atmosphere for the cruise altitude, and with a value of $286.169K$. This results in a Mach number of 0.08164 , which is very low, so no transonic effects are to be expected.

Table 2 summarizes the main parameters used for the CFD simulation. Other parameters were kept at their default values [35].

Table 2: CFD solver parameters used for the simulation.

Discretization	Central plus scalar dissipation
Equation type	RANS
Equation mode	Steady
Turbulence model	Spalart-Allmaras
Turbulence order	First order

For the stopping criteria, either a maximum of 10000 iterations or a convergence of the residuals L2-norm to $1e-6$ in relation to the residuals obtained in the first iteration will be used.

3.2 Geometry

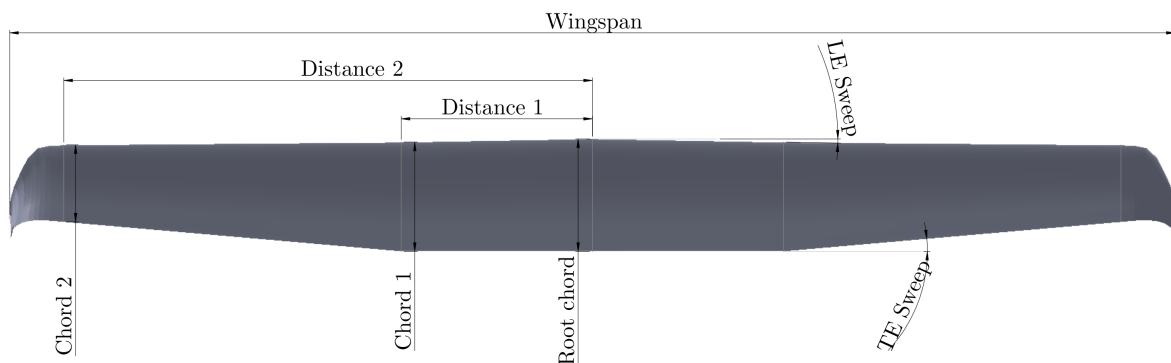


Figure 7: Planform of the AR5 wing.

The planform of the current AR5 wing is shown in Figure 7 and the main design variables associated are presented in Table 3. From distance 2 onward, the wing is composed of a winglet which will not be described here, as it will not be considered in the optimization phase of this work, but it effectively increases the real wingspan whilst keeping the projected wingspan constant and reduces the induced drag, as it will also be shown later. Table 3 also includes the surface area, as well as the projected area of this geometry, which will be kept constant for the initial geometries generated for the optimization. A custom airfoil is used.

Table 3: Some geometric parameters of Tekever AR5 wing.

Wingspan	b	7.565 m
Projected wingspan	b_{proj}	7.246 m
Root chord	c_{root}	0.70 m
Projected area	S_{proj}	2.169 m ²

3.3 Grid convergence study

A grid convergence study was performed, with an approximate uniform refinement ratio of 1.15 between grids by simulating the Tekever AR5 wing at an angle of attack of 1.5° . Sufficient grid refinement will also be important for optimization, namely chord

distribution optimization, during which the mesh in the streamwise direction will get compressed or enlarged together with the geometry, effectively changing its refinement level. This effect will be particularly relevant at the wingtip, where induced drag is expected to be captured.

For those reasons, the used grids all have a greater element clustering near the leading and trailing edge in the streamwise direction, as well as near the tip in the spanwise direction. Since no wall functions were used, the Spalart-Allmaras turbulence model requires an $y^+ \approx 1$, which was satisfied.

Figure 8 shows the evolution of the drag coefficient, as well as the required computational wall time for the different meshes tested. It is possible to observe that the drag coefficient is converging as the grids are refined. The increase in the wall time needed is also noticeable, with the more refined grid taking more than three hours to converge running on two cores of a CPU with a clock speed of $4.5GHz$. A linear increase in the needed RAM memory is verified, at around $2GB$ per $100,000$ elements. Based on this data, it can be concluded that the mesh with about 1.1 million elements is sufficiently refined, as the relative drag coefficient error to the previous iteration is only 0.36% and the computational time needed is less than half of the more refined grid.

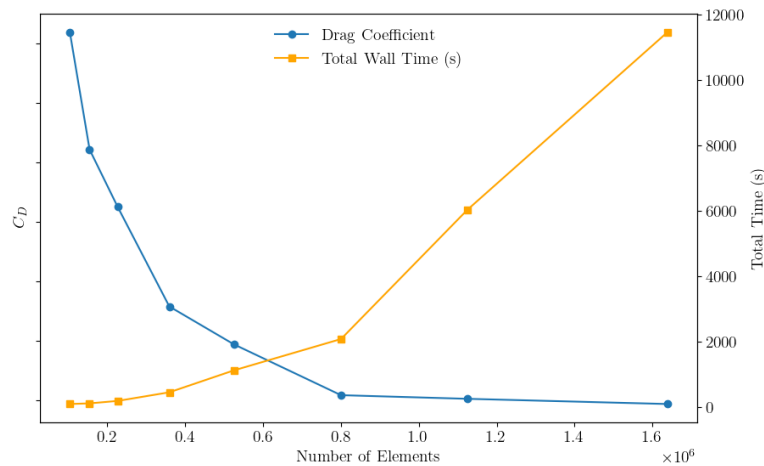


Figure 8: Grid convergence study.

3.4 Domain size study

The influence of the domain size in the solution was also evaluated. For this, several grids with different extrusion distances from the surface normal were tested. However, a coarser surface mesh was used in order to save computational resources, which will increase as the extrusion distance increases. The volume mesh was extruded from the surface mesh using *pyHyp*, which uses a geometric progression to calculate the marching distance of each new layer of cells, given by

$$\Delta d_1(1 - q^{n_k - 1}) - d(1 - q) = 0, \quad (3)$$

in which Δd_1 corresponds to the marching distance of the first layer from the surface, q is the geometric progression ratio, n_k is the number of elements in the off-wall direction

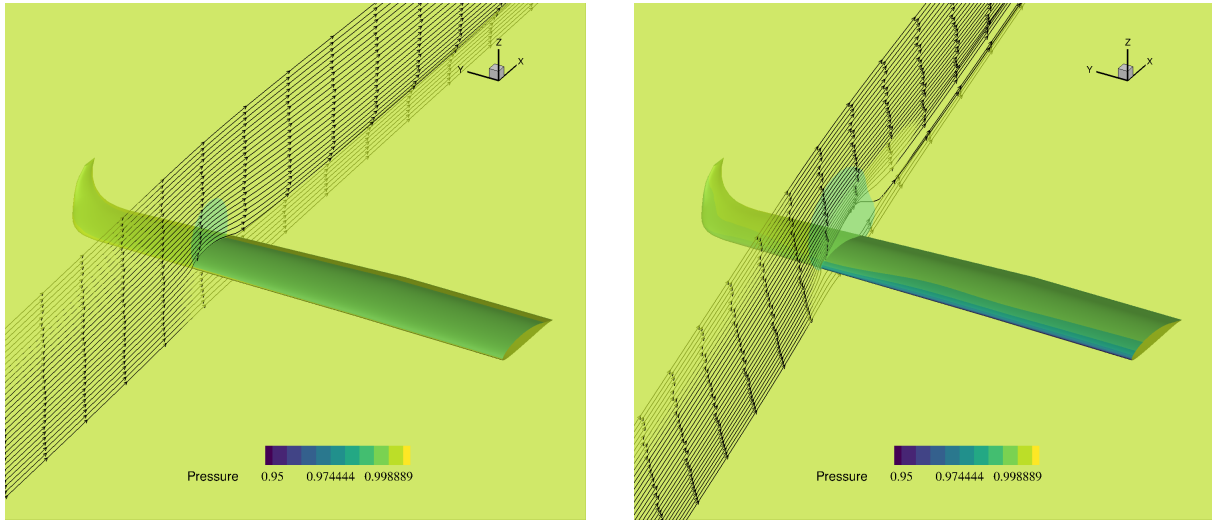
(a) $\alpha = 2.2^\circ$.(b) $\alpha = 20^\circ$.

Figure 9: Pressure distribution and streamlines for different angles of attack.

and d is the total marching distance. Here, the influence of d is studied. However, the geometric progression ratio needs to be kept constant to ensure that the results are not affected by the mesh refinement in the off-wall direction, so n_k is also be changed, according to Equation 3 and q will be kept at around 1.187. The domains are measured in terms of mean aerodynamic chords, and the results are presented in Table 4.

Table 4: Domain size study results

Chords	Nr. elements	Error C_L (%)	Error C_D (%)
5	467,200	-	-
10	496,400	0.23	8.35
15	511,000	1.14	1.89
20	525,600	0.02	0.93
25	540,200	0.10	0.73

It can be observed that the domain size has a bigger impact on the drag than on the lift coefficient. It is also possible to conclude that for 20 chords, the errors for both coefficients are under 1% which was considered enough for this work. Thus, this was the domain size used to perform the aerodynamic analysis.

3.5 Cruise performance

The pressure distribution over the surface can be observed in Figure 9a for an angle of attack of 2.3° , which was found to be the one that produced the prescribed wing lift coefficient for cruise conditions. It is possible to observe that on the upper surface, pressure is higher closer to the leading edge and in the winglet region and in this condition, the flow remains attached to the wing. Figure 9b, on the other hand, shows a flow distribution for a more extreme angle of attack, where it is clearly possible to see the flow separation on the suction side of the wing. In both figures it is possible to see the trailing edge vortex forming.

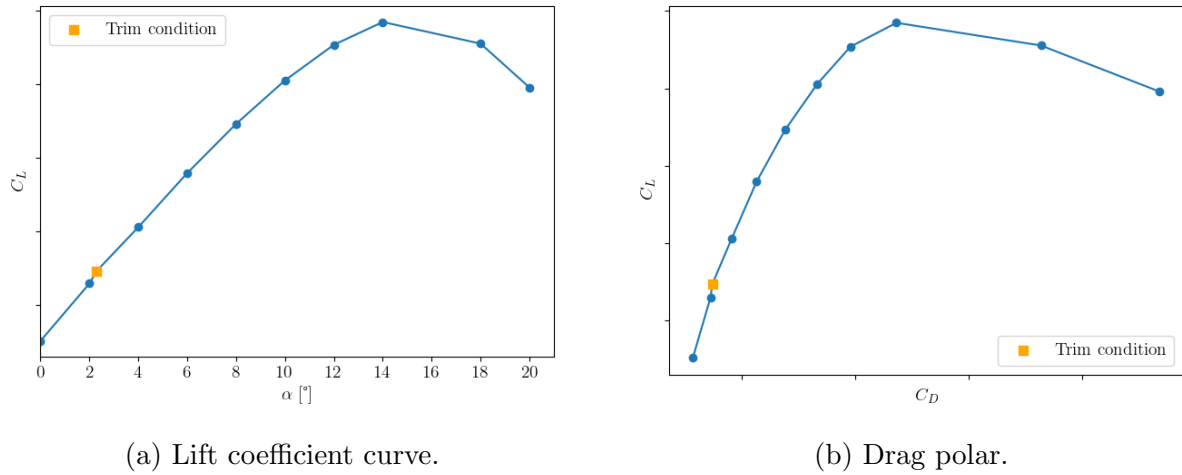


Figure 10: Wing performance for angles of attack between 0° and 20° .

To observe the behavior of the wing with angle of attack, a study was performed and the results presented in Figure 10a. It is possible to see that C_L and C_D increase continuously with angle of attack until around $\alpha = 14^\circ$ and the variation is linear until around $\alpha = 8^\circ$, after which boundary layer separation starts to occur. In the linear region, $C_{L\alpha} = 0.075/^\circ$. A drag polar is also presented in Figure 10b, where efficiency can be analyzed. In this case, it would be higher for the smallest angle in the tested range, of 0° , and the maximum would possibly be for negative angles of attack, outside the tested range, decreasing with increasingly higher angles of attack. The orange square represents the trim operating condition, and it is possible to see that the wing is not operating at its maximum efficiency.

4 WING OPTIMIZATION

4.1 Optimization problem formulation

The formulation of the full optimization problem in standard form and the bounds considered for each design variable are presented in Table 5.

Table 5: Formulation of the full optimization problem in standard form.

		Quantity	Lower bound	Upper bound	Units
minimize	C_D	1	-	-	-
w.r.t.	α	1	0	15	$^\circ$
	γ (twist)	7	-15	15	$^\circ$
	c (chord)	8	15	150	%
	Λ (sweep)	1	0	10	$^\circ$
	airfoil shape	24	-0.05	0.05	m
subject to	$C_{L_{cruise}} = C_{L_{prescribed}}$	1	0.8932	0.8932	-
	$S_{proj} = S_{prescribed}$	1	2.1691	2.1691	m^2
	t (thickness constraint)	24	90	200	%

An FFD box with dimensions $12 \times 8 \times 2$ (streamwise, spanwise and vertical direction, respectively) and a reference axis at 25% of the chord will be used. The parametrizations

for the different variables will be done as discussed in Section 2.3: Twist will be associated with a rotation of each section around the reference axis in the spanwise direction, chord will be parametrized with a scale relative to its initial value in the streamwise and vertical direction (with the same value for both) in relation to its initial value, sweep will be parametrized with a displacement in the streamwise direction of every reference axis point in a linear way followed by a rotation of each control point around the vertical direction and airfoil shape will be parametrized with displacements of the local FFD points in the vertical direction. As discussed before, root twist will be kept at 0° so it will only have seven design variables.

Most limits were defined in such a way that the optimizer would have enough freedom. However, in some cases, like the airfoil shape and the chord lower bound tighter limits had to be imposed to avoid further problems in the volume mesh perturbation resulting in an unusable grid. The thickness constraints were set to span the whole wing and enforced at four spanwise points and six streamwise points. They are important mostly for the airfoil shape optimization and ideally, the wingbox for the required wing structure would be used to set them. As this work deals with the uncoupled aerodynamic shape optimization, it was arbitrated that, at each point, the thickness could not be less than 90% of the initial value.

Regarding the convergence criteria, a successful optimization is finished if either the convergence accuracy is smaller than $1e-6$ or the number of iterations reaches 500. It was verified that on successful optimizations the desired convergence accuracy criterion was met well before the maximum number of iterations.

4.2 Initial geometry characterization

The optimization procedure requires the definition of an initial geometry. That could be the Tekever AR5 current wing, however, given the parametrization method chosen (using FFD volumes), it would be hard to accurately define the design variables in such a way that they would match meaningful wing design parameters. For this reason, two simpler initial geometries were created: one, shown in Figure 11a which is a planification of the Tekever AR5 wing, which had the winglet removed and was then subjected to three main modifications: wingspan extension to match the projected wingspan of the original wing; a change in the trailing edge sweep of the outer section in order to keep the same projected area as the original wing; a removal of the existing twist; the other, shown in Figure 11b is a simple rectangular wing with the same projected wingspan and area as the Tekever AR5 wing and where the original airfoil was replaced with a symmetric NACA 4-series airfoil with the same thickness-to-chord ratio as the original wing. Figure 11 also shows the control points used with each geometry (blue circles) and the reference axis (shown in red).

The rectangular wing utilized a simple parallelepiped-shaped box, while the simplified Tekever AR5 wing employed a more complex FFD box that conforms to the surface geometry. This choice was made due to the non-symmetric nature of the airfoil. In both cases, the reference axis is positioned at 25% and 50% in the streamwise and vertical directions, respectively.

A mesh with a similar refinement level of the one obtained in 3.3 and around 900,000 elements was used for both wings. Given the geometric differences on the planform of both wings, they are not expected to create the same lift at the same angle of attack.

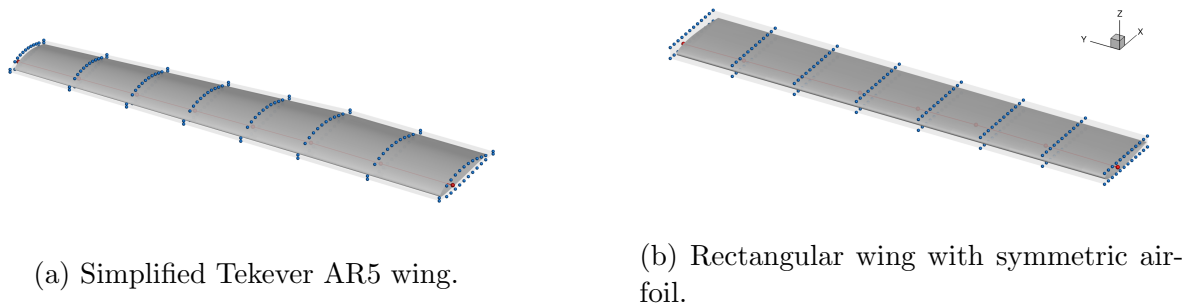


Figure 11: Wings used as a starting point for the optimization problem.

For this reason, an optimization considering only angle of attack as a design variable was performed. Given that the optimizer has only one degree of freedom and one constraint (a fixed lift coefficient), it will only be able to satisfy the constraint, and further drag minimization will not be possible. This allows, however, to find the trim angle of attack needed for both wings to maintain flight at the prescribed $C_{L_{wing}}$ and the corresponding C_D .

4.3 Results for the rectangular wing

The results obtained departing from the rectangular wing and considering different individual design variable optimization subproblems are presented in Table 6 and will be briefly analyzed next.

Table 6: Optimization results for the rectangular wing as starting geometry.

Case		α	C_D
Starting geometry		10.25°	Reference
Twist	Linear variation	12.89°	−2.60%
	3 FFD sections	11.57°	−3.06%
	7 FFD sections	11.86°	−3.21%
Chord	Linear variation	10.05°	−4.15%
	4 FFD sections	10.02°	−5.15%
	8 FFD sections	10.03°	−5.21%
Twist (7 sections) + Chord (8 sections) + Sweep		10.03°	−5.18%
Airfoil	Constant (24 Control points)	6.46°	−6.17%
	Variable (24 × 5 Control points)	5.82°	−8.55%

Twist optimization

The drag coefficient reduces with the increased degrees of freedom provided to the optimizer, however, differences between three and seven FFD sections are very small. Furthermore, less FFD sections usually led to smoother geometries, which may be desirable. The linear twist variation also poses itself as an interesting option, given that the drag reduction difference between it and the best option is 0.61% and it may be easier to manufacture.

Figure 12 compares the lift distribution of the obtained solutions. It can be seen that the distributions tend to the elliptical one as the number of degrees of freedom increase, but do not match it, having a slightly lower variation of the twist angle across the wingspan. A reason for this is the fact that separation started being observed near the trailing edge for such high incidence angles. This is an effect that could not have been predicted with optimization considering inviscid flow, to which the elliptical distribution is optimal.

It is also possible to note that the optimized solution for three and seven FFD sections is in fact very close, explaining the small differences in the drag coefficient observed. It should be noted that for the case with three FFD sections, the optimizer required only 18 function evaluations, whilst for the seven FFD sections case it required 79, representing a very significant difference in computational cost.

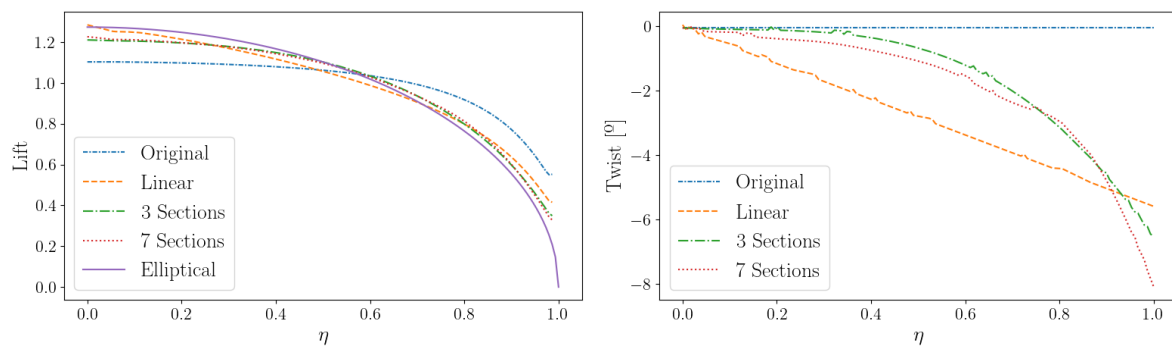


Figure 12: Comparison between lift distribution for the different twist distributions.

Compared to other variables, it is also possible to observe that the twist distribution let the angle of attack be higher than the initial geometry. This is expected, given that all the obtained twist distributions applied a washout, reducing the incidence monotonically across the whole wingspan and, thus, reducing the angle of attack of those sections.

Chord optimization

A similar pattern to the twist distribution was observed regarding the number of degrees of freedom, with the results improving with their increment but with the four and eight FFD section cases yielding very similar results. Here, the difference between the linear variation and the one with eight FFD sections is 1.06%.

The optimized wings feature lower drag coefficients than the ones obtained with twist distribution. One factor that may contribute to this is the additional degree of freedom that chord has at the root, despite having a constraint on the projected area to allow a fair comparison between different geometries. The linear chord variation is effectively equivalent to a trapezoidal wing with taper $\lambda = 0.358$.

Lift and chord distributions for the three cases are shown in Figure 13, where it can be seen that the distribution is indeed closer to elliptical than the one obtained with twist optimization. Regarding the four and eight FFD sections cases, similarly to the twist case, the lift distributions are very close to each other.

Figure 14 compares the suction side pressure distribution of the optimized wing with eight FFD sections with the original one. The elliptical shape and lift distributions ob-

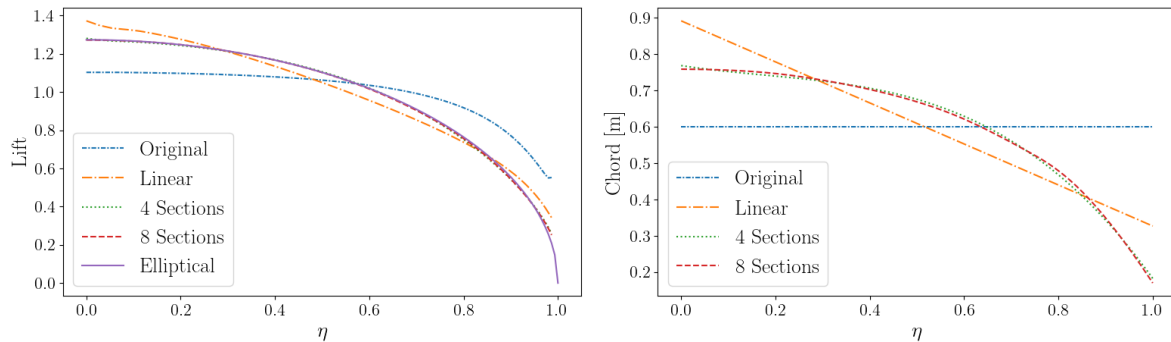


Figure 13: Comparison between lift distribution for the different chord distributions.

tained are evident, with the most visible differences near the wingtip.

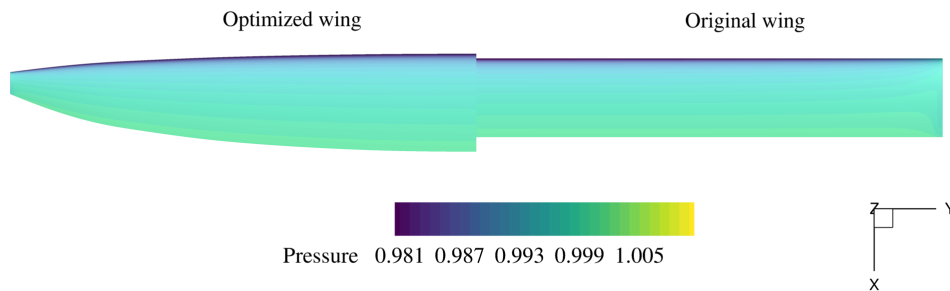


Figure 14: Comparison of the pressure distribution in the suction side between the original and the optimized wing.

Twist + chord + sweep optimization

A combined case with all global design variables was also considered and a very similar drag coefficient to the chord optimization case with eight FFD sections was obtained.

Airfoil optimization

The airfoil optimization considering a constant airfoil on the three-dimensional wing, allowing the capture of induced drag effects, yielded the second best result for drag reduction. Starting from a symmetric NACA 4-series airfoil, the optimizer took 113 function calls and 28 gradient calls to converge to a cambered airfoil, where thickness was mostly taken to the minimum value allowed by the constraint. The obtained airfoil profile is outlined in orange in Figure 15a. With this, the required angle of attack for the prescribed wing lift coefficient was reduced by 3.8° . Another significant change was the distribution of pressure and viscous drag, with the viscous drag only representing 10.5% of the total drag of the optimized geometry, compared to 21.5% on the original wing. Figures 15b and 15c compare the obtained pressure distribution for both airfoils at a section at the middle of the wing and near the tip, respectively. Considering the middle section, it can be seen that the pressure peak on the suction side was greatly decrease on the optimized airfoil, with a smoother recuperation along the chord.

The highest drag reduction was obtained for a variable airfoil across the wing. For this case, a different FFD box was used with only five sections in the wingspan direction in

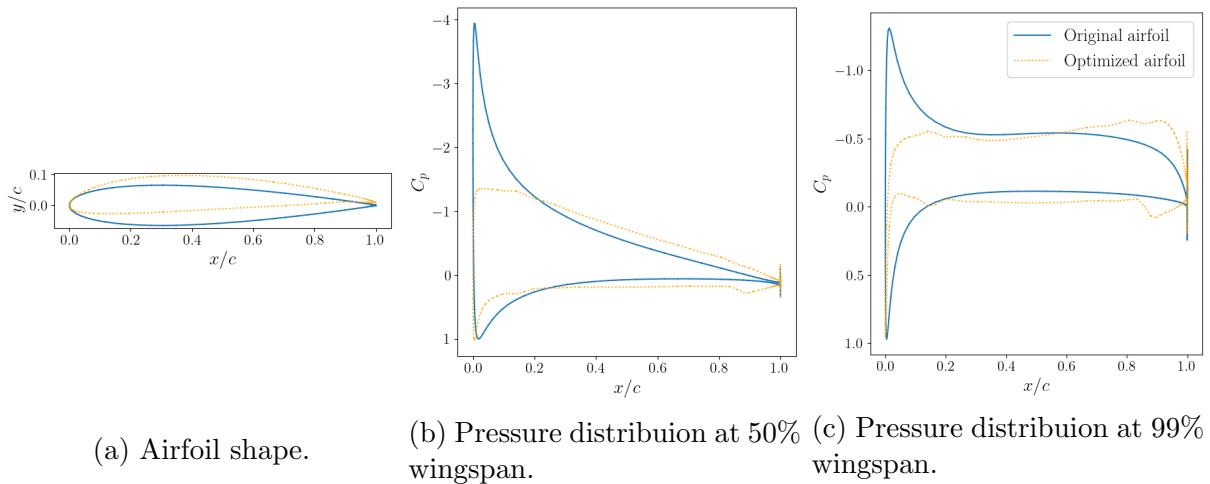


Figure 15: Comparison between the original airfoil and the optimized one.

order to reduce the number of design variables and thus, the computational time needed. Figure 16 shows the airfoils obtained for a section near the root, a section at the middle of the wing and a section near the tip. As expected, the sections near the root and at the middle of the wing are similar, given that a symmetry boundary condition was applied at the root and wingtip effects are not felt yet at those regions, although it has a slightly higher camber near the root. The most notable difference occurs near the wing tip, where the airfoil closely matches the original one, with barely any camber.

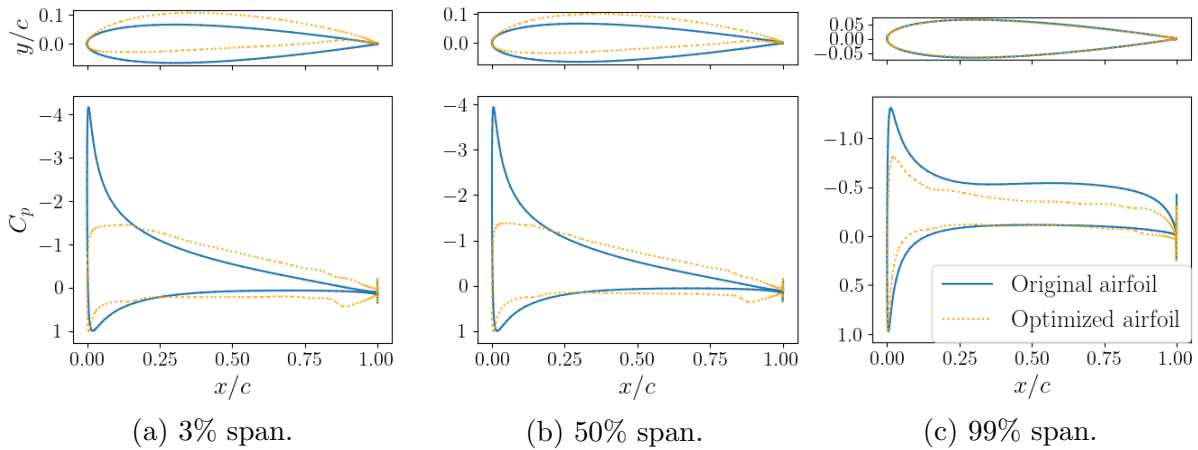


Figure 16: Comparisof the pressure distribution in the suction side between the original and the optimized wing at three sections.

To reach the optimized wing, 128 function calls and 35 gradient calls were needed, representing an increase of only 13% and 25% to the constant airfoil case. It should be noted, however, that the gradients naturally took longer to compute in the variable airfoil case, given the higher number of design variables.

4.4 Results for the simplified Tekever AR5 wing

The summarized results of the optimization using the simplified Tekever AR5 wing as starting geometry geometry are found in Table 7.

Table 7: Optimization results for the simplified Tekever AR5 wing as starting geometry.

Case	α	C_D
Starting geometry	4.03°	Reference
Twist	Linear 2 sections	3.96° –0.39%
	7 FFD sections	4.56° –0.80%
Chord (Linear 2 sections)	4.07°	–0.11%
Twist (7 sections) + Chord (2 linear sections)	4.09°	–1.16%
Airfoil (24 × 8 Control points)	4.59°	–2.54%

Twist optimization

Two cases were defined: A linear variation and a variation with full freedom on the seven FFD sections. Given the the nature of the original geometry, that can be defined by two separate sections, the linear case was defined in two separate sections. To define them, the optimizer was able to change the twist angle at the junction between both sections and at the wingtip. From these values, two separate linear distributions are defined.

It is clear that the initial geometry is much closer to the elliptical lift distribution than the rectangular wing and for this reason, the drag reductions obtained are much smaller. In both scenarios, whether we consider a linear twist distribution defined in two linear sections or seven FFD sections, the optimizer successfully approached an elliptical lift distribution. Notably, the case involving seven FFD sections resulted in nearly twice the drag reduction compared to the linear twist distribution defined in two sections. This significant improvement can be observed in Figure 17, where the lift distribution of the seven FFD sections case is closer to elliptical.

Chord optimization

The parametrization for the chord followed a similar approach to that of twist. However, it was adjusted to ensure a constant taper at the inner section by enforcing the same scaling factor applied to the first two FFD sections, representing the first degree of freedom of the optimizer. The second determined the scaling factor at the wingtip and a linear distribution was then defined between the junction between sections and the wingtip. Due to the constraint imposed by the projected area, the optimizer only had one effective degree of freedom, which determined the taper, λ , for the outer section of the wing. This case was the only configuration tested, in order to maintain the original wing design philosophy. The optimizer produced a different geometry, with the final scale factors being 1.09 and 0.76, but the resulting drag reduction for this particular case was minimal.

Twist + chord optimization

A combined optimization case with full freedom to the seven twist design variables and the taper of the outer section was also considered and, unlike the rectangular wing, the results were better than any of the singular cases, obtaining a drag reduction of more than 1%. The obtained scale factors for chord distribution were 1.13 and 0.67, and a twist distribution was also applied, with its lowest value, at the wingtip, being -4.14° . This

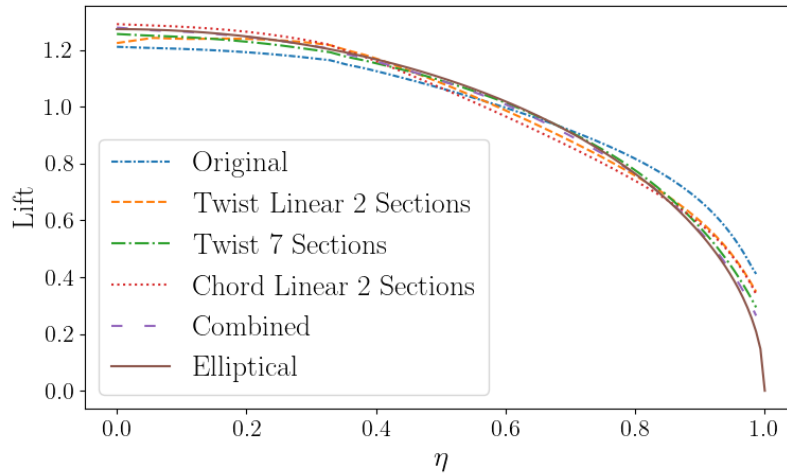


Figure 17: Comparison between lift distribution for the different parametrizations.

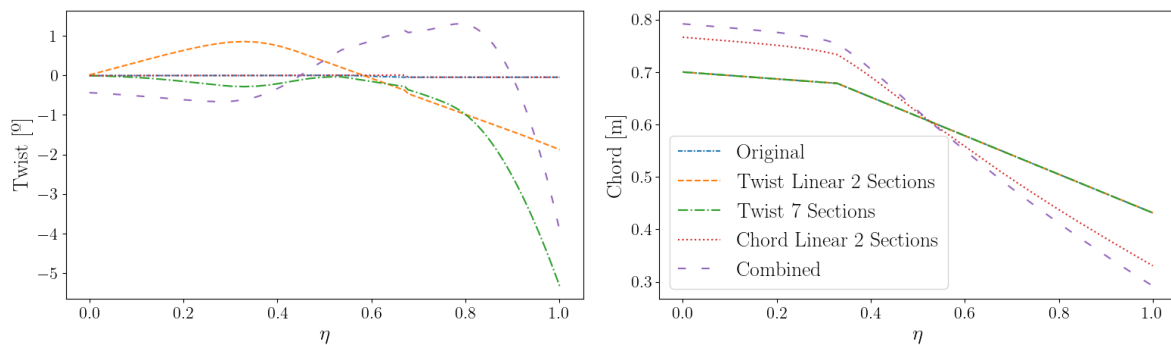


Figure 18: Comparison between chord and twist distribution for the different parametrizations.

results in a lift distribution almost elliptical, as Figure 17 shows. In Figure 18, it is possible to see the twist and chord distribution obtained for this and the other parametrizations tested, starting from the simplified Tekever AR5 wing geometry.

Airfoil optimization

Finally, a variation in airfoil shape along the wingspan was also optimized. Similar to the rectangular wing case, this optimization yielded the greatest reduction. However, the reduction achieved was significantly smaller than that of the rectangular wing, which was expected considering that the starting profile for this case was already optimized for the specified flight condition.

Figure 19 provides a comparison between the original airfoil profile and the optimized one at three different sections along the wingspan. The comparison reveals that the modifications in the initial sections mainly involved a decrease in thickness and a slight reduction in camber. Interestingly, at the wing tip, the optimizer converged to a symmetric airfoil, resembling the case of the rectangular wing with a symmetric airfoil geometry as a starting geometry.

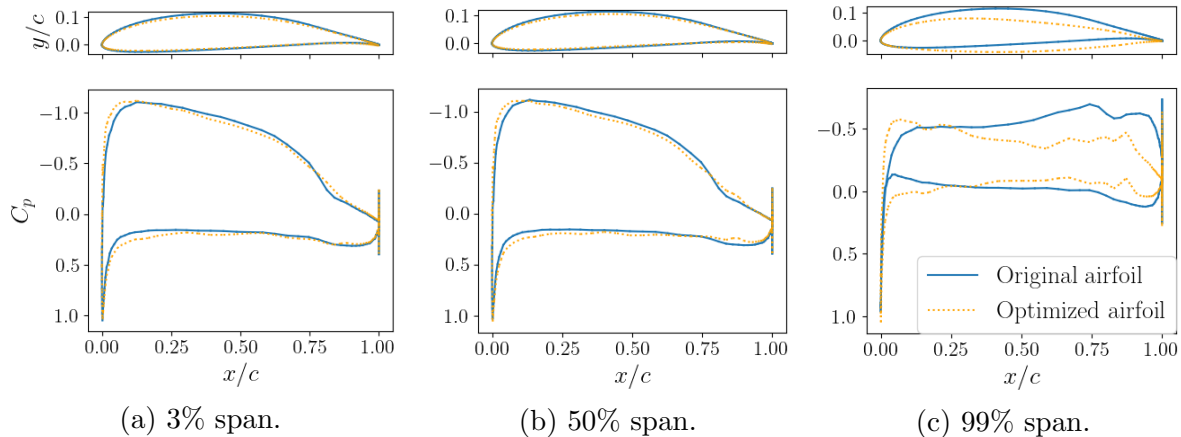


Figure 19: Comparison of the pressure distribution in the suction side between the original and the optimized wing at three sections.

5 CONCLUSIONS

This work presents a study on the optimization of two different starting geometries, namely the rectangular wing with a symmetrical airfoil and the simplified Tekever AR5 wing, using different parametrizations. The results demonstrate notable drag reductions of up to 8.55% for the rectangular wing and 2.54% for the simplified Tekever AR5 wing.

Additionally, most of the optimized lift distributions closely approximated the elliptical one. The use of two starting geometries in this study highlights the potential for optimizing both naive geometries and geometries obtained from extensive design processes, thereby showing capabilities for both refinement of existing designs as well as the creation of new designs from scratch considering a specified flight condition.

ACKNOWLEDGEMENTS

The authors would like to thank Tekever for supporting the production of this article by offering all the needed data to successfully produce the analyses here shown. This work was supported by FCT, through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] K. P. Valavanis and G. J. Vachtsevanos. *Handbook of Unmanned Aerial Vehicles*, volume 1. Springer. ISBN 978-90-481-9706-4.
- [2] V. Prisacariu. The history and the evolution of uavs from the beginning till the 70s. *Journal of Defense Resources Management (JoDRM)*, 8(1):181–189, 2017.
- [3] J. Vos, A. Rizzi, D. Darracq, and E. Hirschel. Navier–Stokes solvers in European aircraft design. *Progress in Aerospace Sciences*, 38(8):601–697, 2002. doi:10.1016/S0376-0421(02)00050-7.
- [4] P. E. Rubbert. CFD and the changing world of airplane design. In *19th Congress of the International Council of the Aeronautical Sciences*, volume 19, pages LVII–LVII,

- Anaheim, USA, September 1994. American Institute of Aeronautics and Astronautics.
- [5] S. Defoort, M. Méheut, B. Paluch, R. Liaboef, R. Murray, D. C. Mincu, and J.-M. David. Conceptual design of disruptive aircraft configurations based on high-fidelity oad process. In *Aviation Technology, Integration, and Operations Conference*, page 3663, Atlanta, USA, june 2018.
- [6] R. Haimes and M. Drela. On the construction of aircraft conceptual geometry for high-fidelity analysis and design. In *50th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 683, Nashville, USA, january 2012.
- [7] P. D. Bravo-Mosquera, H. D. Cerón-Muñoz, G. Díaz-Vázquez, and F. M. Catalano. Conceptual design and CFD analysis of a new prototype of agricultural aircraft. *Aerospace Science and Technology*, 80:156–176, 2018.
- [8] Drone Industry Insights. Global drone market report 2022-2030, 2022.
- [9] J. R. Martins. Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, 239:105391, 2022. doi:10.1016/j.compfluid.2022.105391.
- [10] C. A. Mader, G. K. Kenway, A. Yildirim, and J. R. Martins. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 17(9):508–527, 2020.
- [11] G. K. Kenway, C. A. Mader, P. He, and J. R. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019. doi:10.1016/j.paerosci.2019.05.002.
- [12] G. Kenway, G. Kennedy, and J. R. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *13th AIAA/ISSMO multidisciplinary analysis optimization conference*, page 9231, Fort Worth, USA, September 2010.
- [13] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. Martins. pyOptSparse: A python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. doi:10.21105/joss.02564.
- [14] R. A. McDonald and J. R. Gloude-mans. Open Vehicle Sketch Pad: An open source parametric geometry and analysis tool for conceptual aircraft design. In *AIAA SciTech 2022 Forum and Exposition*, page 4, San Diego, USA, january 2022.
- [15] D. Poirier, S. Allmaras, D. McCarthy, M. Smith, and F. Enomoto. The CGNS system. In *29th AIAA, Fluid Dynamics Conference*, page 3007, Honolulu, USA, June .
- [16] C. Rumsey, B. Wedan, T. Hauser, and M. Poinot. Recent updates to the CFD general notation system (CGNS). In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1264.

- [17] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331. doi:10.1002/nme.2579.
- [18] N. Secco, G. K. W. Kenway, P. He, C. A. Mader, and J. R. R. A. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4), 2021. doi:10.2514/1.J059491.
- [19] M. Z. M. Shah, B. Basuno, and A. Abdullah. Comparative study on several type of turbulence model available in Ansys-Fluent software for Onera M6 wing aerodynamic analysis. *Journal of Advanced Mechanical Engineering Applications*, 1(1):9–19, 2020.
- [20] P. Eliasson. Influence of turbulence modelling and grid resolution in computations of the dpw7 crm configuration. In *AIAA Aviation Forum and Exposition*, page 4091, San Diego, USA, june 2023.
- [21] L. Kral. Recent experience with different turbulence models applied to the calculation of flow over aircraft components. *Progress in Aerospace Sciences*, 34(7-8):481–541, 1998. doi:10.1016/S0376-0421(98)00009-8.
- [22] J. G. Coder, T. H. Pulliam, D. Hue, G. K. Kenway, and A. J. Scalfani. Contributions to the 6th AIAA CFD drag prediction workshop using structured grid methods. In *55th AIAA Aerospace Sciences Meeting*, page 960, Grapevine, USA, january 2017.
- [23] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, Reno, USA, january 1992.
- [24] L. Eca, F. Pereira, and G. Vaz. Viscous flow simulations at high reynolds numbers without wall functions: Is y^+ approximately 1 enough for the near-wall cells? *Computers & Fluids*, 170:157–175, 2018. doi:10.1016/j.compfluid.2018.04.035.
- [25] MDO Lab. *MACH-Aero Documentation*. MDO Lab, University of Michigan, USA, 2022.
- [26] MDO Lab, University of Michigan, USA. *pyGeo Documentation*. MDO Lab, 2022.
- [27] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 17(1-2): 103–126, 2008.
- [28] Z. Lyu, Z. Xu, and J. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics, Chengdu, Sichuan, China*, volume 11, page 585. Citeseer, 2014.
- [29] N. Bons. *High-fidelity Wing Design Exploration with Gradient-based Optimization*. PhD thesis, University of Michigan, USA, 2020.

- [30] H. Koyuncuoglu and P. He. CFD based multi-component aerodynamic optimization for wing propeller coupling. In *AIAA SciTech 2023 Forum and Exposition*, page 1844, National Harbor, USA, january 2023.
- [31] G. Yang and A. Da Ronch. Aerodynamic shape optimisation of benchmark problems using SU2. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0412, 2018.
- [32] Y. Yu, Z. Lyu, Z. Xu, and J. R. Martins. On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization. *Aerospace Science and Technology*, 75:183–199, 2018. doi:10.1016/j.ast.2018.01.016.
- [33] D. Kraft. A software package for sequential quadratic programming. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1988.
- [34] Tekever. Tekever>Meet Our UAS Models>AR5. URL <https://http://www.tekever.com/models/ar5/>.
- [35] MDO Lab. *ADFlow Documentation*. MDO Lab, University of Michigan, USA, 2022.



NEURAL LEVEL SET TOPOLOGY OPTIMIZATION USING UNFITTED FINITE ELEMENTS

Connor N. Mallon^{1*}, Aaron W. Thornton², Matthew R. Hill^{1,2}, Santiago Badia^{3*}

1: Department of Chemical and Biological Engineering
Monash University
Wellington Rd Clayton, 3800, Victoria, Australia.
connor.mallon@monash.edu

2: CSIRO
Research Way Clayton 3168, Victoria, Australia,

3: School of Mathematics
Monash University Wellington Rd Clayton, 3800, Victoria, Australia.
connor.mallon@monash.edu

Abstract. *To facilitate widespread adoption of automated engineering design techniques, existing methods must become more efficient and generalizable. In the field of topology optimization, this requires the coupling of modern optimization methods with solvers capable of handling arbitrary problems. In this work, a topology optimization method for general multiphysics problems is presented. We leverage a convolutional neural parameterization of a level set for a description of the geometry and use this in an unfitted finite element method that is differentiable with respect to the level set everywhere in the domain. We construct the parameter to objective map in such a way that the gradient can be computed entirely by automatic differentiation at roughly the cost of an objective function evaluation. The method produces optimized topologies that are similar in performance yet exhibit greater regularity than baseline approaches on standard benchmarks whilst having the ability to solve a more general class of problems, e.g., interface-coupled multiphysics.*

Keywords: neural, unfitted, level set, topology optimization

1 INTRODUCTION

After the birth of topology optimizations (TOs) in the field of structural design [1], efforts have been made to increase the effectiveness of such automated design approaches and allow for their deployment on a more general class of problems [2, 3].

A plethora of TO strategies exist through the literature, the most common of which being density-based methods using the so-called SIMP (Solid isotropic microstructure with penalization for intermediate densities) method [4]. These involve varying a material distribution continuously between 0 and 1 to introduce an artificial representation of the boundary. Although simple for basic structural problems, a way to represent intermediate design variables arising at the boundary must be included, which becomes increasingly complex in multi-physics applications and makes imposing arbitrary boundary conditions non-trivial [5].

An alternative technique that can overcome some of the problems presented by density methods and tackle a more general class of problems (e.g., interface-coupling multiphysics and problems that involve surface PDEs on boundaries) is the level set (LS) TO method [6, 7]. Using this approach, the boundary is described by the zero iso-surface of a LS function. It is instead this LS function that is varied to obtain optimized designs. A precise location of the boundary is then available.

A variety of alternative implementations of the LS TO method have been made [8]. They can be distinguished, among other things, by how they update the topology at each iteration and their means of geometry mapping. The methods to update geometries involve either updating the solution of Hamilton-Jacobi equations by a velocity field based on the sensitivity information [9, 10] or using a parameterization of the topology that is an explicit function of the design variables of a steepest descent optimization scheme. The latter approach allows one to leverage well-established nonlinear programming techniques and is the method selected for this work. Types of geometry mappings include using the LS function to define a conformal mesh to the boundary, see e.g. [11, 12], which requires re-meshing at each iteration, density-based mappings (see e.g. [13–15]), which recover some of the issues related to density methods, or unfitted/immersed boundary techniques (see, e.g., [16–18]). unfitted methods rely on a fixed background mesh and capture the precise location of the boundary in the model using integration on sub-triangulations aligned with the zero iso-surface of the LS function. By doing so, re-meshing and the introduction of intermediate densities are avoided.

A known issue with unfitted techniques is the ill-conditioning problem associated with small cut elements. The common XFEM [19, 20] approach uses a finite element (FE) space restricted to the interior domain and cut cells for the solution and requires stabilization in the vicinity of the boundary by, for example, ghost penalty terms [21] or cell aggregation [22, 23]. These methods are consistent and can provide high-order approximation [24] however the support of the stabilization terms change depending on the location of the cut cells, leading to potential non-differentiability in the optimization problem which can harm the convergence of gradient-based optimization algorithms. The specific unfitted TO technique used in this work is instead a version of the finite cell method (FCM) [16], in which a non-consistent penalty term is added everywhere in the fictitious domain (outside the physical domain) to provide robustness. This stabilization is suitable for TO because it is differentiable with respect to the level set parameterization (see Section 5). An implementation of the FCM for TO is made in [16], which uses a refined grid for the

material boundary compared to the solution to capture fine-scale geometry. We instead use subgrid triangulations using the LS function as in [19] to capture fine-scale structure in the integration and thus avoid the need to increase the number of design variables parameterizing the geometry. The loss of consistency of the FCM is not an issue in TO, where high-order approximations are not very relevant.

It is natural to use a FE function for a discrete representation of the LS. Doing so, a parameterization is obtained with a user-controlled resolution. A common approach to the optimization problem is to take the degrees of freedom (DOFs) of this LS as the design variables [19, 25, 26]. This choice, however, means that each parameter is only capable of a local influence on the LS function to the surrounding cells. As an alternative, we introduce a neural parameterization of the geometry. We set our design variables in this case to be the parameters of a particular artificial neural network (NN) that outputs the LS function DOFs. Performing this step, we obtain control over the optimization problem by controlling the connectivity of parameters, allowing them to influence cells spread across the domain. Although the expressivity is unchanged, since the LSs are in the same space, the parameters controlling the evolution act to optimize the geometry at multiple scales. The ultimate objective is then for the optimization process to unveil regularized geometries with good performance.

The combination of machine learning and TO was explored as early as the 1990s [27] but has gained massive momentum in recent years [28, 29]. NNs and other ML techniques can be incorporated into the TO process in many ways. Common data-driven approaches attempt to train networks to map problem descriptions directly to a geometry [30–33]. These however require pre-training on already optimized samples and suffer from a lack of generalisability [29]. Others replace some or all of the optimization loop for accelerated convergence by training an auxiliary network [34, 35]. These approaches are based on the premise, which in general is not necessarily true, that early iterations of the optimization contain the information to produce performant optimal geometries. An alternative method is the inclusion of a NN as an alternative parameterization of the geometry [36–38]. These approaches typically optimize the parameters of a NN representing a continuous function that maps positions in space to a density. These approaches tend to focus on reducing the dimensionality of the design space assuming that NNs can efficiently achieve expressiveness with a small number of parameters [39]. A reduction in parameters does not, however, necessarily lead to faster convergence for NNs [37] compared to the standard SIMP approach.

Instead of focusing on a neural parameterization that reduces the dimensionality of the problem, we select a network description of the geometry which is specifically designed to learn effectively on problems involving the segmentation of a domain. The network used in this case is a modification of the U-Net convolutional network. The U-Net architecture was originally developed for biomedical image segmentation tasks [40] but has proven successful for a variety of applications in which multi-scale features and spatial correlation is important [41]. These networks are typically composed of encoding and decoding halves. The encoding section maps the context of input images into a low dimensional latent space which is localized in the upsampling section to provide segmentation at the desired resolution. The work in [38] exploited the properties of this network showing improved performance with a U-Net density parameterization for a SIMP structural optimization problem. Similar to [38], we use a trainable input vector for the network and feed this into the up-sampling half of the U-Net. In contrast to most applications of this network, we

have no input image and therefore do not need the encoding of half of the network. It is the up-sampling (or decoding) part of the network the one that provides the parameterization of the multi-scale features which are important in this context.

The Julia [42] programming language is used to implement all aspects of this project with the FE toolbox Gridap [43, 44] being the main package utilized. We also use the Julia machine learning library Flux [45] for the implementation of the U-Net. Using these foundations, we implement a routine combining NNs with an unfitted FE based TO. The main contributions of this work are the presentation of:

- An unfitted LS TO method with a NN parameterization that assists to achieve simple optimized geometries with similar or better performance compared to baseline methods and
- a fully automatically differentiable unfitted LS TO method for multiphysics problems with complex boundary conditions.

We present the overall framework as follows. First, in Section 2, we present the entire optimization loop at a high level. We then go into more detail about various stages in the loop. Details of the architecture of the neural network are found in Section 3, details of the geometry processing are presented in Section 4, the numerical discretization of the problem is presented in Section 5 and the gradient implementation in Section 6. We then benchmark the method against baseline methods and show the generality of the method with an application to a multiphysics problem with complex boundary conditions in Section 7.

2 OPTIMIZATION PROBLEM

In this section, we provide a succinct overview of the overall TO algorithm proposed in this work. We aim to solve the problem:

$$\begin{aligned} \min_{\mathbf{p}} J(\mathbf{u}(\mathbf{p}), \mathbf{p}) \\ \text{s.t } \mathcal{R}(\mathbf{u}(\mathbf{p}), \mathbf{p}) &= 0, \\ \mathcal{V}(\mathbf{p}) &= 0, \end{aligned} \tag{1}$$

where \mathbf{p} are the parameters that describe our geometry, \mathcal{V} is an equality constraint (e.g. for the volume), \mathcal{R} is the PDE residual, \mathbf{u} is the solution of the PDE and J is the objective. If desired, further equality and inequality constraints can then be imposed by adding penalty terms to the objective function.

2.1 Optimization Loop

To optimize the parameters \mathbf{p} , we make use of a gradient-based optimization strategy. To do so, we establish a map between \mathbf{p} and J and a means to compute the gradient $\frac{dJ}{d\mathbf{p}}$ for parameter updates. For the neural LS TO method, \mathbf{p} represents the parameters of a particular NN that outputs a vector φ . This vector is processed using the operator \mathcal{H} to obtain the LS ϕ used in the PDE and objective function. It is also only through ϕ that the PDE and objective depend on the parameter vector \mathbf{p} . With these definitions, we present the optimization loop for solving (1) in Figure 1.

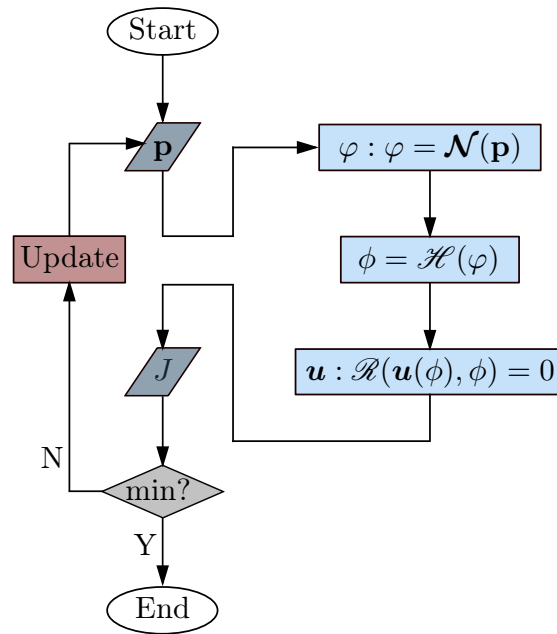


Figure 1: Computational graph of the optimization loop. We start with an input to the system \mathbf{p} , and perform the forward pass by descending through the blue boxes on the right side to obtain a performance measure J as explained in Section 2.1.1. The convergence criteria are used to decide whether this J represents an acceptable minimum. If not, the backward pass is performed to compute an update for the parameters as explained in Section 2.1.2 and the loop is continued.

2.1.1 Forward Pass

To solve the forward problem and get a performance measure J for a set of parameters \mathbf{p} we descend through the light blue boxes on the right hand side of Figure 1 by performing the following steps:

1. In the first light blue box, we evaluate the network $\mathcal{N} : \mathbf{p} \in \mathbb{R}^{N_p} \mapsto \varphi \in \mathbb{R}^N$, where N_p is the number of parameters and \mathcal{N} is as defined in Section 3.
2. In the second light blue box, we process the output of the network φ using the operator $\mathcal{H} : \varphi \in \mathbb{R}^N \mapsto \phi \in V_h^1$ to obtain a suitable LS description of the geometry ϕ , where V_h^1 is the FE space for the LS defined in Section 4. This involves an interpolation on a FE space, smoothing and the inclusion of an equality constraint on the geometry so that the final LS function satisfies $\mathcal{V}(\phi) = 0$. This step is broken down in Section 4. We enforce the equality constraint here to allow for the use of an unconstrained optimization method suitable for NNs.
3. In the third light blue box, we solve the FE problem associated with the weak form of the residual $\mathcal{R}(\mathbf{u}_h(\phi), \phi) = 0$ on the domain segmentation defined by ϕ for an approximate solution \mathbf{u}_h obtained using a FE discretization. Details of the FCM used to solve the problem are given in Section 5. We can then evaluate the objective $J(\mathbf{u}_h(\phi), \phi) \in \mathbb{R}$.

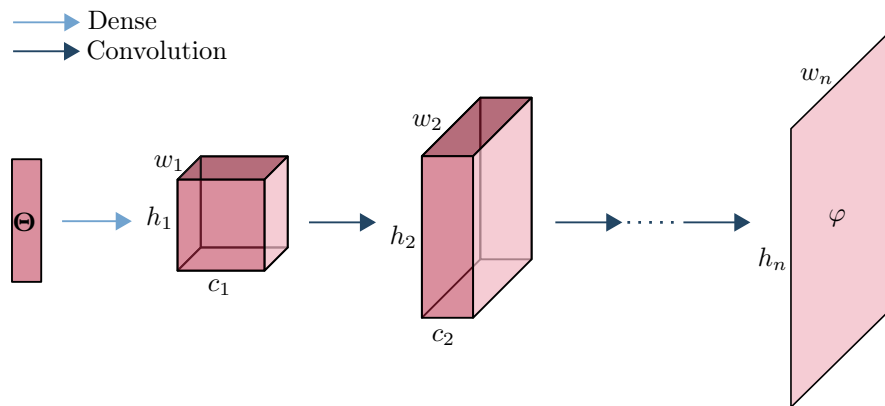


Figure 2: Architecture of the NN. A trainable input vector Θ is fed into the network. The light blue arrow involves a set of operations that include a fully connected layer and the dark blue arrow involves a set of operations that include a convolutional layer. The intermediate data structures are of size (c_l, w_l, h_l) and the final output, after n layers, gives φ .

2.1.2 Backwards Pass

To perform the update for the parameters using a steepest descent optimization strategy, we require an evaluation of the gradient $\frac{\partial J}{\partial \mathbf{p}}$. To do this efficiently at a cost roughly matching that of the forward pass, we use reverse mode differentiation and define rules to propagate sensitivities through each of the steps in the forward pass. We use an adjoint rule for the PDE and use automatic differentiation for all of the partial derivatives, including the derivative of integrals with respect to the LS. The derivative is then passed onto a chosen optimizer to update the parameters \mathbf{p} . The implementation of the gradient computation is discussed in Section 6.

3 NETWORK ARCHITECTURE

In this section, we present the NN architecture used in the geometry parameterization. This describes the mapping between the parameters \mathbf{p} and the LS vector φ . The network used here is mainly built from convolutional layers. In contrast to a fully connected network, convolutional networks connect smaller sets of neurons in each layer assuming that neurons in close proximity have a more important relationship. This is a natural relaxation for the processing of spatial data. Furthermore, convolutional networks are made efficient by the assumption that features that are found in one local block are likely to be found in a different local block, i.e. somewhere else in the domain. This is done by sharing parameters amongst local blocks in the form of a convolutional filter. The specific architecture used in this work is the one presented in [38]. Because we are simply reparameterizing the pixel values, there is no input into the network in the traditional sense. Most approaches combining NNs and TO use fully connected layers and take the network to provide a map between a spatial input \mathbf{x} and a scalar output $f_\theta(\mathbf{x})$ [36, 37]. In our case, we only need a single evaluation of the network to output a vector $f_\theta(\Theta)$ that represents the entire discrete LS function. The input vector in our case is just taken to be a set of trainable parameters Θ .

The architecture of the network is illustrated in Figure 2. The first arrow indicated with the label Dense in Figure 2 contains a fully connected layer and a reshape:

$$\mathbf{x}^{(1)}(\Theta) = \text{reshape}(\mathbf{W}\Theta + \mathbf{b}), \quad \mathbf{x}^1 \in \mathbb{R}^{c_1, w_1, h_1}, \quad (2)$$

where $\mathbf{W} \in \mathbb{R}^{c_1 w_1 h_1, N_\Theta}$ is the dense weight matrix, $\mathbf{b} \in \mathbb{R}^{c_1 w_1 h_1}$ is the bias and reshape : $\mathbb{R}^{c_1 w_1 h_1} \rightarrow \mathbb{R}^{c_1, w_1, h_1}$ is a reshaping map. The lengths N_Θ , c_1 , w_1 and h_1 represent the length of Θ , the initial number of channels, the latent space image width and the latent space image height, respectively. Note here that the first nonlinearity is imposed at the beginning of the next layer.

The upsampling convolutional layers, depicted by the dark blue arrow in Figure 2, are defined as:

$$\mathbf{x}^{(l+1)}(\mathbf{x}^{(l)}) = \mathcal{P}^{(l)}(\varrho(\Phi^{(l)}(\tanh(\mathbf{x}^{(l)}))), \quad \mathbf{x}^{(l)} \in \mathbb{R}^{c_l, w_l, h_l}, \quad \mathbf{x}^{(l+1)} \in \mathbb{R}^{c_{l+1}, w_{l+1}, h_{l+1}}, \quad (3)$$

where $\Phi^{(l)} : \mathbb{R}^{c_l, w_l, h_l} \rightarrow \mathbb{R}^{c_l, w_{l+1}, h_{l+1}}$ is a bilinear resize, ϱ is a normalization to a mean of 0 and variance of 1 across the channel dimension and $\mathcal{P}^{(l)} : \mathbb{R}^{c_l, w_{l+1}, h_{l+1}} \rightarrow \mathbb{R}^{c_{l+1}, w_{l+1}, h_{l+1}}$ is a convolutional operator with kernel size (5, 5).

In this approach, our input to the network Θ is taken to be trainable. So to define our parameter vector \mathbf{p} , we collect the parameters of \mathbf{W} , \mathbf{b} , Θ and \mathbf{P}^i into a vector $\mathbf{p} \in \mathbb{R}^{N_p}$. Then, by composing the layers, we obtain the function $\mathcal{N} : \mathbf{p} \in \mathbb{R}^{N_p} \mapsto \varphi \in \mathbb{R}^N$:

$$\mathcal{N} = \mathbf{x}^{(n)}(\dots(\mathbf{x}^{(2)}(\mathbf{x}^{(1)}(\Theta))), \quad (4)$$

where N is the number of DOFs of the FE function for the LS. The output image of size w_n, h_n is interpreted as a vector φ representing the DOFs of the LS FE function as described in Section 4. Importantly, the locality is preserved when defining this function. As we traverse the network, we follow the design principle of the upsampling section of the U-Net and trade-off channel depth for spatial resolution. This means that, in general, the widths w and heights h will increase as we move through the network and the number of channels c will decrease as we move through the network. The exact trade-offs here can vary, but we must set the resizes and channel refinement to ensure $h_n w_n = N$ and $c_n = 1$ so that our output φ makes sense as a vector representing the DOFs for the LS. We can increase the network size in the dense layer by increasing N_Θ , w_1 and h_1 and in the convolutional layers by increasing the elements in c and the number of layers.

In a typical LS method, the user is required to manually initialize the geometry with holes. One approach to initializing the NN is to pre-train the network to output the manually selected geometry with holes, as in [46]. Using this as an initial guess, however, causes the geometry to converge quickly to poor local minima. A more common approach when working with NNs and a given objective map is to start with small random weights [47]. In this case, we have high asymmetry in the weights and little activation function saturation. It turns out that a random initialization of the NN with the volume constraint gives an initial guess of a domain with a few holes in random locations. This is in contrast to initializing the LS itself with random values which gives a geometry with many small holes and fine features which is not necessarily desirable [48]. As is common in NN approaches, we can then easily take multiple seeds for the geometry using different random initializations to alleviate initialization dependency. The random initializations simply correspond to different size holes in different locations. Using this method we eliminate the need for manual geometry initialization.

4 LEVEL SET FUNCTION PROCESSING

In this section, we detail the LS description of the geometry used in the method. We explain how the output vector from the NN φ is used to define the LS to be used in the

numerical method. The computation of the LS function is performed in four steps. We introduce an interpolation step in Section 4.1 to obtain a first LS function $\phi_{n(1)}$. In Section 4.2, we smooth out this LS to obtain a smooth LS function $\phi_{f(2)}$. Then, we perform a reinitialization of that LS in Section 4.3 to obtain a LS function $\phi_{s(3)}$. Finally, in Section 4.4, we propose a volume correction strategy to end up with the final LS function $\phi_{b(4)}$. In the subsequent sections, we will refer to this final LS function $\phi_{b(4)}$ as ϕ .

4.1 Interpolation

In this method, we work with discrete LS functions $\phi \in V_h^1$. The DOFs of V_h^1 are the values of the function at the vertices of \mathcal{T}_h , which we denote with $\{\mathbf{x}_i\}_{i=1}^N$, where N is the number of mesh nodes. Thus, there is an isomorphism between V_h^1 and \mathbb{R}^N . The output image vector of the NN φ are the DOFs that uniquely determine the LS FE function.

4.2 Smoothing

Next, we convolve the function with a linear filter for smoothing:

$$\phi_{f(2)_i} = \left(\sum_{j=1}^N w_{ij} \right)^{-1} \left(\sum_{j=1}^N w_{ij} \phi_{n(1)_j} \right), \quad (5)$$

where $w_{ij} = \max(0, r_f - |x_i - x_j|)$, $\phi_{n(1)_j}$ is the j^{th} degree of freedom of the function $\phi_{n(1)}$, $\phi_{f(2)_i}$ is the i^{th} degree of freedom of the function $\phi_{f(2)}$ and r_f is the smoothing radius.

4.3 Reinitialization

We then reinitialize the LS as a signed distance function. This is often done to gain control over the spatial gradient of the LS function to improve convergence [8]. In our case, it is of even greater importance as it also guarantees that when we apply the translation to the LS to satisfy the volume constraint, we do not artificially introduce volumes into the domain far from the boundary where the LS is close to zero. This would add discontinuity to the problem harming convergence. To perform this step, we solve the reinitialization equation in [49] to obtain a signed distance function $\phi_{s(3)} \in V_h^1$:

$$\frac{\partial \phi_{s(3)}}{\partial \tau} + \text{sign}(\phi_{s(3)}) (|\nabla \phi_{s(3)}| - 1) = 0. \quad (6)$$

The problem is solved using Picard iterations at steady state using the initial point $\phi_{f(2)}$ so that only one solve of the adjoint equation is required in the backwards pass. Artificial viscosity is added to the problem for stabilization and a surface penalty term is integrated on the embedded boundary to prevent movement of the zero iso-surface in the reinitialization. The weak form of the problem is then to find the solution $\phi_{s(3)} \in V_h^1$ of the equation:

$$\int_{\Omega} (\mathbf{w} \cdot \nabla(\phi_{s(3)})v + c_a h |\mathbf{w}| |\nabla(\phi_{s(3)})| \nabla v - \text{sign}(\phi_{s(3)})v) d\Omega + \int_{\Gamma} (\phi_{s(3)}, v) d\Gamma = 0 \quad (7)$$

for all $v \in V_h^1$, where $\mathbf{w} = \text{sign}(\phi_{s(3)}) (\nabla(\phi_{s(3)}) / |\nabla(\phi_{s(3)})|)$, c_a is the stabilization coefficient, set to 3, and h is the element size.

4.4 Translation

The next task is to impose the volume constraint. This is done by applying a translation to the entire LS function. Here we solve the nonlinear equation \mathcal{V} for the scalar bias b

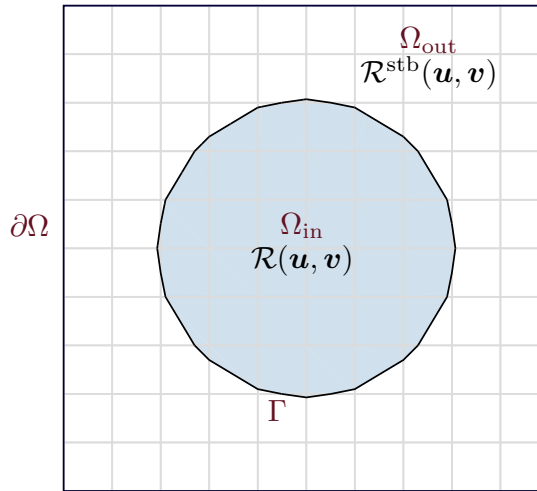


Figure 3: Illustration of the problem domains. The background domain is segmented into the domains Ω_{in} and Ω_{out} . In Ω_{in} , the physical terms $\mathcal{R}(\mathbf{u}, \mathbf{v})$ are integrated. In Ω_{out} , the stabilization terms $\mathcal{R}^{\text{stb}}(\mathbf{u}, \mathbf{v})$ are integrated.

to obtain a translation to the LS function such that the domain Ω_{in} satisfies the volume constraint. A bisection method is used to find the root b of the equation:

$$\mathcal{V} = \int_{\Omega_{\text{in}}} d\Omega - \mathcal{V}_0 = 0, \quad (8)$$

where Ω_{in} is defined by the LS $\phi_{b(4)} = \phi_{s(3)} + b$ and \mathcal{V}_0 is the volume fraction given by the constraint. This is the LS which is then used to define the boundary of the FE problem. Details of how the LS is used to define domains is given in the next Section.

5 NUMERICAL DISCRETIZATION

Here we formulate the numerical discretization of the PDEs being solved in the simulation step of the optimization loop. We consider a background polyhedral bounded domain Ω with boundary $\partial\Omega$. A LS function $\phi(x)$ is used to split Ω into two subdomains $\Omega_{\text{in}}(\phi)$ and $\Omega_{\text{out}}(\phi)$ as follows (see Figure 3):

$$\Omega_{\text{in}}(\phi) = \{x \in \Omega : \phi(x) > 0\}, \quad \Omega_{\text{out}}(\phi) = \{x \in \Omega : \phi(x) < 0\}. \quad (9)$$

We denote the interface between these two subdomains as $\Gamma(\phi) \doteq \partial\Omega_{\text{in}}(\phi) \cap \partial\Omega_{\text{out}}(\phi)$. Let us assume that $\partial\Omega_{\text{in}}(\phi) \cap \partial\Omega \neq \emptyset$. The domain $\Omega_{\text{in}}(\phi)$ is the one in which we consider our PDE problem. The weak form of the continuous problem can be stated as follows: find $\mathbf{u} \in V$ such that

$$\int_{\Omega_{\text{in}}(\phi)} \mathcal{R}(\mathbf{u}, \mathbf{v}) d\Omega = 0, \quad \forall v \in V. \quad (10)$$

where V is a Hilbert space in which the problem is well-posed. We consider zero flux Neumann boundary conditions on $\partial\Omega_{\text{in}}(\phi) \setminus \partial\Omega$ and (for simplicity) homogeneous boundary conditions on $\partial\Omega_{\text{in}}(\phi) \cap \partial\Omega$; the generalization to non-homogeneous Dirichlet boundary conditions is straightforward.

The domain $\Omega_{\text{in}}(\phi)$ will change along the optimization process. As a result, it is not practical to compute body-fitted unstructured meshes for the geometrical discretization

of $\Omega_{\text{in}}(\phi)$. Instead, we consider a background mesh, which can simply be a Cartesian background mesh of Ω and make use of an unfitted FE discretization. Unfitted (or embedded) discretizations relax the geometrical constraints but pose additional challenges to the numerical discretization [21, 50]. The first issue is the integration over cut cells (for details, see [51]). The other issue is the so-called *small cut cell* problem. Cut cells with arbitrary small support lead to ill-conditioned systems [52]. Various techniques can be used to stabilize the problem including the cut finite element method (CutFEM) [21], the aggregated finite element method (AgFEM) [50] and the FCM [53]. We select the FCM method in this case because it is differentiable with respect to the LS everywhere in the domain. The CutFEM and AgFEM are conversely not differentiable.

In order to state the discrete form of the continuous problem, we introduce the unfitted FE space. Let \mathcal{T}_h represent a conforming, quasi-uniform and shape regular partition (mesh) of Ω , h being a characteristic mesh size. Ω can be a trivial geometry, e.g., a square or cube, and \mathcal{T}_h can be a Cartesian mesh. We define a nodal Lagrangian FE space of order $q \geq 1$ on \mathcal{T}_h as:

$$V_h^q = \{\mathbf{v}_h \in \mathcal{C}^0(\Omega) : \mathbf{v}_h|_K \in \mathcal{X}_q(K) \forall K \in \mathcal{T}_h\}, \quad (11)$$

where $\mathcal{X}_q(K)$ is the space $\mathcal{Q}_q(K)$ of polynomials with maximum degree q for each variable when \mathcal{T}_h is a quadrilateral or hexahedral mesh and the space $\mathcal{P}_q(K)$ of polynomials of total degree q when \mathcal{T}_h is a simplicial mesh. In this work, we consider low order spaces, which is the most reasonable choice for TO applications.

The weak formulation of the problem solved by the method is now described. Let us represent with $V_{h,0}^q = V_h^q \cap \mathcal{C}_0^0(\bar{\Omega})$ the nodal FE space that vanishes on the boundary $\partial\Omega$.

Now, we can define a first-order FCM discretization of (10) as follows: find $\mathbf{u}_h \in V_h^1$ such that

$$\int_{\Omega_{\text{in}}(\phi)} \mathcal{R}(\mathbf{u}_h, \mathbf{v}_h) d\Omega + \int_{\Omega \setminus \Omega_{\text{in}}(\phi)} \alpha_{\text{out}} \mathcal{R}^{\text{stb}}(\mathbf{u}_h, \mathbf{v}_h) d\Omega = 0, \quad \forall \mathbf{v} \in V_h^1, \quad (12)$$

where $\alpha_{\text{out}} \ll 1$ is the penalty parameter and \mathcal{R}^{stb} is a *stabilizing* differential operator on the artificial domain.

5.1 Poisson Equation

The poisson equation is used to model the temperature $\theta(\mathbf{x})$ that satisfies

$$-\nabla \cdot (\boldsymbol{\kappa} \nabla \theta) = f \quad \text{in } \Omega_{\text{in}}(\phi), \quad (13)$$

where $\boldsymbol{\kappa}$ is the thermal conductivity tensor of the material and f is a thermal source. A zero Dirichlet condition ($\theta = 0$) is prescribed on $\Omega_{\text{in}}(\phi) \cap \partial\Omega$ and a zero flux condition ($\mathbf{n} \cdot (\boldsymbol{\kappa} \nabla \theta) = 0$) is prescribed on $\Gamma(\phi)$.

The FCM approximation of this problem without a source term reads as: find $\theta_h \in V_{h,0}^1$ such that

$$\int_{\Omega_{\text{in}}(\phi)} \boldsymbol{\kappa} \nabla \theta_h \cdot \nabla v_h d\Omega + \int_{\Omega \setminus \Omega_{\text{in}}(\phi)} \alpha_{\text{out}} \boldsymbol{\kappa} \nabla \theta_h \cdot \nabla v_h d\Omega = 0, \quad \forall v_h \in V_{h,0}^1. \quad (14)$$

One can readily check that this method is weakly enforcing the zero flux condition on $\Gamma(\phi)$ as $\alpha_{\text{out}} \rightarrow 0$. We observe that we use the same differential operator in the artificial domain for stabilisation purposes (times the scaling coefficient α_{out}).

We consider the TO problem in which we aim at finding a level-set ϕ that minimizes the integral of the temperature:

$$J(\phi, \theta_h(\phi)) = \int_{\Omega(\phi)} \theta_h(\phi) \, d\Omega, \quad (15)$$

where $\theta_h(\phi)$ is the solution of (14) given ϕ .

5.2 Linear elasticity

We want to obtain the displacement $\mathbf{d}(\mathbf{x})$ that satisfies the linear elasticity equation

$$-\nabla \cdot \boldsymbol{\sigma}(\mathbf{d}) = \mathbf{f} \quad \text{in } \Omega_{\text{in}}(\phi), \quad (16)$$

where $\boldsymbol{\sigma} = \lambda \text{tr}(\boldsymbol{\varepsilon})\mathbf{I} + 2\mu\boldsymbol{\varepsilon}$ is the stress tensor, $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{d} + (\nabla\mathbf{d})^\top)$ is the symmetric gradient, \mathbf{d} is the displacement, λ and μ are the Lamé parameters given by $\lambda = (E\nu)/((1+\nu)(1-2\nu))$ and $\mu = E/(2(1+\nu))$ and \mathbf{f} is the forcing term. A zero Dirichlet condition ($\mathbf{d} = \mathbf{0}$) is prescribed on $\Omega_{\text{in}}(\phi) \cap \partial\Omega$ and a zero stress condition ($\mathbf{n} \cdot \boldsymbol{\sigma}(\mathbf{d}) = 0$) is prescribed on $\Gamma(\phi)$.

The FCM approximation of this problem without a forcing term reads as: find $\mathbf{d}_h \in \mathbf{V}_{h,0}^1 \doteq [V_{h,0}^1]^D$ such that

$$\int_{\Omega_{\text{in}}(\phi)} \boldsymbol{\sigma}(\mathbf{d}_h) : \boldsymbol{\varepsilon}(\mathbf{v}_h) \, d\Omega + \int_{\Omega \setminus \Omega_{\text{in}}(\phi)} \alpha_{\text{out}} \boldsymbol{\sigma}(\mathbf{d}_h) : \boldsymbol{\varepsilon}(\mathbf{v}_h) \, d\Omega = 0, \quad \forall \mathbf{v}_h \in \mathbf{V}_{h,0}^1. \quad (17)$$

It is easy to check that the zero-stress condition on $\Gamma(\phi)$ is recovered as $\alpha_{\text{out}} \rightarrow 0$. For the linear elasticity equation, we again use the same differential operator in the artificial domain for stabilisation purposes.

A typical TO problem in solid mechanics is the minimization of the strain energy. In this case, we aim at finding a level-set ϕ that minimizes the cost function

$$J(\phi, \mathbf{d}_h(\phi)) = \int_{\Omega(\phi)} \boldsymbol{\sigma}(\mathbf{d}_h) : \boldsymbol{\varepsilon}(\mathbf{d}_h) \, d\Omega, \quad (18)$$

where $\mathbf{d}_h(\phi)$ is the solution of (17) given ϕ .

5.3 Linear Elasticity with Fluid Forcing Terms

Once again, we want to obtain the displacement $\mathbf{d}_h(\mathbf{x})$ that satisfies the linear elasticity formulation in (16). In this case, however, we consider the surface traction exerted by the fluid:

$$\int_{\Gamma(\phi)} (\mathbf{n} \cdot \nabla \mathbf{u}_h - p_h \mathbf{n}) \cdot \mathbf{v} \, dx, \quad (19)$$

where the fluid velocity \mathbf{u}_h and pressure field p_h are obtained by solving a fluid problem in the domain Ω_{out} . These fields are obtained by solving the Stokes equations with a Brinkmann penalization as in [54] but without intermediate interpolation of permeabilities at the boundary.

In order to approximate the fluid problem, we use a mixed FE method, namely the equal order pair $\mathbf{V}_{h,0}^1 \times V_h^1$.

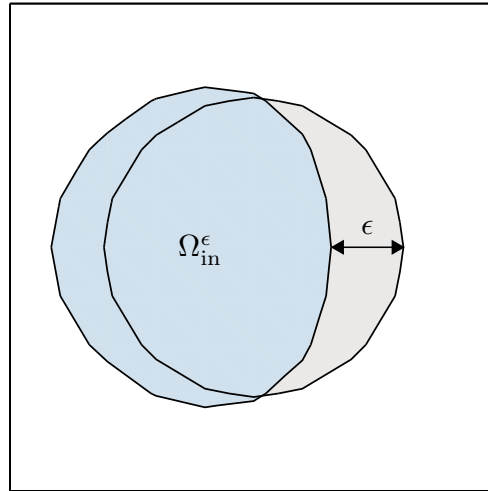


Figure 4: A small perturbation with size ϵ to the LS to form a new domain $\Omega_{\text{in}}^\epsilon$.

We find $(\mathbf{u}_h, p_h) \in \mathbf{V}_{h,0}^1 \times V_{h,0}^1$ such that:

$$\int_{\Omega} [\alpha \mathbf{u}_h \cdot \boldsymbol{\psi}_h + \mu \nabla \mathbf{u}_h \cdot \nabla \boldsymbol{\psi}_h - p_h (\nabla \cdot \boldsymbol{\psi}_h) - (\nabla \cdot \mathbf{u}_h) q_h - h^2 \nabla p_h \cdot \nabla q_h] dx = 0, \quad (20)$$

$$\forall \boldsymbol{\psi}_h, q_h \in \mathbf{V}_{h,0}^1 \times V_h^1,$$

where

$$\begin{cases} \alpha = 0 & \text{in } \Omega_{\text{out}} \\ \alpha = \alpha_u & \text{in } \Omega_{\text{in}} \end{cases} \quad (21)$$

using an artificial porosity α_u to make the fluid problem well posed in the solid domain Ω_{in} and enforce the no-slip boundary condition. In the fluid domain Ω_{out} we recover the Stokes equations.

The TO problem once again involves finding a level-set ϕ that minimizes the elastic strain using (18).

5.4 Differentiability of the unfitted FE solver

An important property for the convergence of a TO strategy is the notion of shape differentiability of the cost function. A functional under a PDE constraint is considered shape differentiable if the mapping $\phi \rightarrow J(\mathbf{u}_h(\phi), \phi)$ is differentiable at the admissible set of domains in Ω defined by ϕ . In this section, we discuss how the choice of FE stabilization can effect this property.

The model problem in (14) with a FCM stabilization is equivalent to that of a typical two phase conductivity problem. With a solution $\mathbf{u}_h \in H^1(\Omega)$, the functional $J(\mathbf{u}_h, \phi)$ for this problem can be proven to be shape differentiable, see [55, Theorem 4.9].

Conversely, unfitted techniques involving stabilization only in the vicinity of the boundary are in general not shape differentiable. Regions of non differentiability arise (typically when the boundary crosses over mesh nodes) harming the convergence of the geometry to optimized solutions [56]. To see why this is the case, we investigate shape perturbations under the CutFEM and AgFEM formulations.

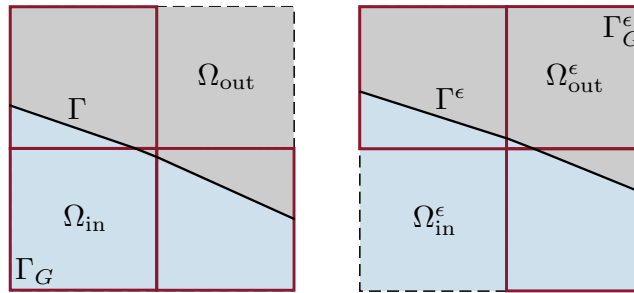


Figure 5: The change in a portion of the ghost skeleton triangulation Γ_G , depicted by the red faces, before and after a perturbation to the boundary.

If we were to use a restricted space for the solution and add ghost penalty terms in the vicinity of the interface following the CutFEM method [57], the problem is to find $u_h \in W_{h,0}^1$, such that:

$$\int_{\Omega_{in}(\phi)} \mathcal{R}(\mathbf{u}_h, \mathbf{v}_h) d\Omega_{\Omega_{in}} + j(\mathbf{u}_h, \mathbf{v}_h)_{\Gamma_G} = 0, \quad \forall v_h \in W_{h,0}^1, \quad (22)$$

for a ghost penalty term j on a ghost skeleton triangulation Γ_G using the space $W_{h,0}^1$ as in [57]. Consider the change to the domain Ω_{in} from Figure 3 caused by a perturbation $\epsilon\delta$, where $\epsilon \in \mathbb{R}$ and $\delta \in V_h^1$, to the LS function ϕ . The resulting domain Ω_{in}^ϵ may be as in Figure 4. The ghost penalty term in this formulation does not depend on ϵ and instead changes depending on which cells are cut. Specifically, if Γ^ϵ crosses over a mesh node as in Figure 5, the ghost skeleton triangulation includes the faces of a new element. Non-zero terms are integrated on this triangulation introducing discontinuity to the problem with respect to the shape, since:

$$\lim_{\epsilon \rightarrow 0} [j(\mathbf{u}_h, \mathbf{v}_h)_{\Gamma_G^\epsilon} - j(\mathbf{u}_h, \mathbf{v}_h)_{\Gamma_G}] \neq 0. \quad (23)$$

With different terms added to the linear system which do not go to zero with ϵ , we can see that the derivative of the solutions with respect to the shape can be ill-defined. A cost function operating on the solution could not, in general, be shape differentiable at these points.

In the AgFEM, the problem is to find $u_h \in V_{h,0}^{agg}$ such that:

$$\mathcal{R}(\mathcal{E}(\mathbf{u}_h), \mathcal{E}(\mathbf{v}_h))_{\Omega_{in}} \quad \forall v_h \in V_{h,0}^{agg}, \quad (24)$$

for the extension operator \mathcal{E} and space $V_{h,0}^{agg}$ as defined in [50]. Similar to the CutFEM, regions of non-differentiability in the problem exist when the zero isosurface of the LS crosses over mesh nodes:

$$\lim_{\epsilon \rightarrow 0} [\mathcal{R}(\mathcal{E}(\mathbf{u}_h), \mathcal{E}(\mathbf{v}_h))_{\Omega_{in}} - \mathcal{R}(\mathcal{E}^\epsilon(\mathbf{u}_h), \mathcal{E}(\mathbf{v}_h))_{\Omega_{in}^\epsilon}] \neq 0, \quad (25)$$

since $\mathcal{E}^\epsilon(\mathbf{u}) \neq \mathcal{E}(\mathbf{u}_h)$ in general because the support for cut cells potentially changes depending on which cells are cut. Following the same reasoning as above, the method is therefore not shape differentiable. The same is true for other methods which use stabilization approaches that act only in the vicinity of cut cells, e.g. [58], using similar branching strategies when crossing a node or reaching a certain threshold. ■

6 GRADIENT IMPLEMENTATION

To the best of our knowledge, there is no existing implementation of an unfitted LS TO method that accepts arbitrary residuals defining the PDE and computes the entire gradient $\frac{dJ}{d\mathbf{p}}$ by automatic differentiation. Making use of a backwards pass, we do this efficiently by defining differentiation rules for each of the steps in the method.

6.1 Integral Differentiation Operator

The backward pass is mainly composed of gradients of integrals with respect to the DOFs of FE functions. To make the derivative computation efficient, we exploit the fact that the DOFs only have an effect on surrounding cells and utilize the optimizations exploiting sparsity in the FE library Gridap [59]. Integrals in the domain can be divided into cell-wise components:

$$\mathcal{J}(\mathbf{u}, \mathbf{v}, \phi) = \sum_{K \in \mathcal{T}_h} \mathcal{J}^K(\mathbf{u}^K, \mathbf{v}^K, \phi^K), \quad (26)$$

where $\mathbf{u}^K \in \mathbb{R}^{\Sigma_u}$, $\mathbf{v}^K \in \mathbb{R}^{\Sigma_v}$ and $\phi^K \in \mathbb{R}^{\Sigma_\phi}$ are the DOFs parameterising the restrictions of u, v and ϕ to the cell K and Σ_u, Σ_v and Σ_ϕ are the number of DOFs in K for the respective functions. Gradients can then be computed at roughly the cost of an integral evaluation for each cell K :

$$\frac{\partial \mathcal{J}^K}{\partial \phi} = \nabla_\phi^F \mathcal{J}^K(\mathbf{u}^K, \mathbf{v}^K, \phi^K), \quad (27)$$

where the operator ∇_ϕ^F represents taking the gradient with respect to ϕ^K using a vectorized forward propagation of dual numbers [60]. To make taking derivatives in this way possible for the LS, we implement the integrals so that each ϕ^K is accepted as the argument to compute the contribution \mathcal{J}^K :

$$\mathcal{J}^K : \phi^K \in \mathbb{R}^{\Sigma_\phi} \mapsto \mathcal{J}_K(\mathbf{u}^K, \mathbf{v}^K, \phi^K) \in \mathbb{R}. \quad (28)$$

where

$$\mathcal{J}^K(\mathbf{u}^K, \mathbf{v}^K, \phi^K) = \int_{K(\phi^K)} \mathcal{I}(\mathbf{u}^K, \mathbf{v}^K) dK \quad (29)$$

A key point is that the integral function subroutines, including all the unfitted FE tools, are implemented in such a way as to allow the propagation of dual numbers through the code. We also make use of a reverse mode operator ∇^R for the backwards propagation of derivatives used where appropriate, e.g., for the NN.

6.2 Backwards Pass Routine

We now present the backwards pass in detail. To compute the sensitivity of the objective with respect to the parameters, we start with the seed $\frac{dJ}{dJ} = 1$ and propagate derivatives in reverse mode using the chain rule:

$$\frac{dJ}{d\mathbf{p}} = \frac{dJ}{dJ} \left(\frac{\partial J}{\partial \phi} + \frac{\partial J}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\phi} \right) \frac{d\phi}{d\varphi} \frac{d\varphi}{d\mathbf{p}} \quad (30)$$

where an adjoint method on the problem residual \mathcal{R} is used to differentiate through the PDE:

$$\frac{\partial J}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\phi} = -\lambda^T \frac{d\mathcal{R}}{d\phi} \quad (\text{here we solved } \frac{d\mathcal{R}}{d\mathbf{u}} \lambda = \frac{dJ}{d\mathbf{u}}). \quad (31)$$

We then use the chain rule to differentiate through the LS function processing steps:

$$\frac{d\phi}{d\varphi} = \frac{d\phi}{d\phi_{s(3)}} \frac{d\phi_{s(3)}}{d\phi_{f(2)}} \frac{d\phi_{f(2)}}{d\varphi}. \quad (32)$$

The volume constraint here involved a root finding method. To differentiate through this step, we utilize the implicit function theorem:

$$\frac{d\phi}{d\phi_{s(3)}} = \frac{\partial\phi}{\partial\phi_{s(3)}} - \frac{\partial\phi}{\partial b} \frac{\partial\mathcal{V}^{-1}}{\partial b} \frac{\partial\mathcal{V}}{\partial\phi_{s(3)}}, \quad (33)$$

and to differentiate through the signed distance map, we use the adjoint method once again for the residual \mathcal{R}_s equal to the integral in (7). Finally, we use standard backpropagation to compute the derivative with respect to the parameters of the NN. The steps of the backward pass are presented explicitly in Algorithm 1.

7 NUMERICAL EXPERIMENTS

7.1 Benchmark Results

We first compare the optimized results obtained for benchmark problems against baseline methods using the model problems presented in Section 5.

The method presented in this work, the NN-LS method, is compared against its non-neural counterpart the Pixel LS (Pixel-LS) method, that is, the same method without a neural prior where instead the DOFs of the LS function are taken as the optimization parameters. We also compare the method against the SIMP method of TO again using both a neural prior (NN-SIMP) and the standard approach (Pixel-SIMP). Following a standard SIMP implementation of the heat conduction problem, we use a conductivity based on the power law $k = \alpha_T + (1 - \alpha_T)\rho^\gamma$ where ρ is a design variable given by a FE function constructed on the space V_h^1 where the DOFs are the optimization parameters or output vector of the NN in the Pixel-SIMP and NN-SIMP cases, respectively, and γ is the penalization parameter, taken to be equal to 3. Similarly, for the SIMP implementation of the structural problem, we use a Youngs Modulus $E = \alpha_d + (1 - \alpha_d)\rho^\gamma$. To maintain a fair comparison between methods, we use the optimizers most widely regarded as suitable for the particular parameterization. Namely, we take the most commonly used MMA optimization strategy [61] for the pixel parameterization and the ADAM strategy [62] for the NN.

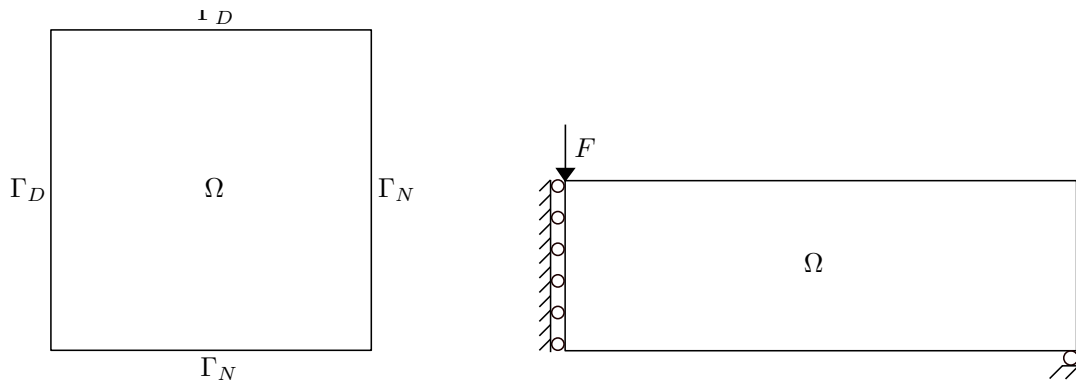
7.1.1 Benchmark Problems

The first problem studied is the poisson equation to model heat conduction with the setup in Figure 6a selected from [63]. For this problem, we set $k_0 = 1m^2s^{-1}$, $\alpha_T = 0.01$, $f = 0.01Ks^{-1}$, use homogenous Dirichlet and Neumann conditions and use a mesh of 95×95 with a 0.4 volume fraction. For the NN, we set the number of convolutional layers to 5, $N_\Theta = 64$, $w = (12, 12, 24, 46, 96, 96)$, $l = (12, 12, 24, 46, 96, 96)$ and $c = (16, 128, 64, 32, 16, 1)$.

The second problem studied is the typical MBB problem described in [4] with the setup as in Figure 6b. For this problem, we set $\nu = 0.3$, $E = 1Pa$, $\alpha_d = 0.001$ and $F = 1N$ and use a mesh of 287×95 with a 0.4 volume fraction. We use the same network as in the poisson problem but set $w = (36, 36, 64, 128, 256, 256)$

Algorithm 1 Backwards Pass

Initialize $\frac{dJ}{dJ} \leftarrow 1$
Extract $\phi^K \in \mathbb{R}^{\Sigma\phi}$, $\mathbf{u}^K \in \mathbb{R}^{\Sigma u}$ from $\phi, \mathbf{u} \forall K \in \mathcal{T}_h$.
for $K \in \mathcal{T}_h$ **do**
 $\frac{\partial J^K}{\partial \mathbf{u}} \leftarrow \nabla_{\mathbf{u}}^F J(\mathbf{u}^K, \phi^K)$
 $\frac{\partial J^K}{\partial \phi} \leftarrow \nabla_{\phi}^F J(\mathbf{u}^K, \phi^K)$
end for
Assemble the gradients $\frac{\partial J}{\partial \mathbf{u}} \in \mathbb{R}^{N_u}$ and $\frac{\partial J}{\partial \phi} \in \mathbb{R}^N$
Assemble the sparse jacobian associated with the residual $\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \in \mathbb{R}^{N_u, N_u}$
Solve the adjoint equation $\frac{\partial \mathcal{R}}{\partial \mathbf{u}} \lambda = \frac{\partial J}{\partial \mathbf{u}}$ for $\lambda \in \mathbb{R}^{N_u}$
Extract $\lambda^K \in \mathbb{R}^{\Sigma u}$ from $\lambda \forall K \in \mathcal{T}_h$
for $K \in \mathcal{T}_h$ **do**
 $\frac{\partial J}{\partial \mathbf{u}} \frac{\partial \mathbf{u}^K}{\partial \phi} \leftarrow \nabla_{\mathbf{u}}^F \mathcal{R}(\mathbf{u}^K, \lambda^K, \phi^K)$
end for
Assemble the gradient $\frac{\partial J}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \phi} \in \mathbb{R}^N$
 $\frac{dJ}{d\phi} \leftarrow \frac{\partial J}{\partial \phi} + \frac{\partial J}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \phi}$
Compute the vector-jacobian-products:
 $\frac{\partial J}{\partial b} \leftarrow \frac{dJ}{d\phi} \nabla_b^R \phi(\phi_{s(3)}, b)$
 $\frac{\partial J}{\partial \phi} \leftarrow \frac{dJ}{d\phi} \nabla_{\phi}^R \phi(\phi_{s(3)}, b)$
Compute the gradients:
 $\frac{\partial V}{\partial \phi} \leftarrow \nabla_{\mathbf{u}}^R V(\phi_{s(3)}, b)$
 $\frac{\partial V}{\partial b} \leftarrow \nabla_{\mathbf{u}}^F V(\phi_{s(3)}, b)$
 $\frac{dJ}{d\phi_{s(3)}} \leftarrow \frac{\partial J}{\partial \phi_{s(3)}} - \frac{\partial J}{\partial b} \frac{\partial \psi^{-1}}{\partial b} \frac{\partial \psi}{\partial \phi_{s(3)}}$
Assemble the sparse jacobian associated with the residual $\frac{\partial \mathcal{R}_s}{\partial \phi_{s(3)}} \in \mathbb{R}^{N, N}$
Solve the adjoint equation $\frac{\partial \mathcal{R}_s}{\partial \phi_{s(3)}} \lambda_s = \frac{\partial J}{\partial \phi_{s(3)}}$ for $\lambda_s \in \mathbb{R}^N$
for $K \in \mathcal{T}_h$ **do**
 $\frac{d\mathcal{R}_s}{d\phi_{f(2)}}^K \leftarrow \nabla_{\phi_{f(2)}}^F \mathcal{R}_s(\phi_{s(3)}^K, \lambda_s^K, \phi_{f(2)}^K)$
end for
Assemble the gradient $\frac{dJ}{d\phi_{f(2)}} \in \mathbb{R}^N$
Compute the vector-jacobian-products:
 $\frac{dJ}{d\varphi} \leftarrow \frac{dJ}{d\phi_{f(2)}} \nabla_{\varphi}^R (\phi_{f(2)}(\varphi))$
 $\frac{dJ}{d\mathbf{p}} \leftarrow \frac{dJ}{d\varphi} \nabla_{\mathbf{p}}^R (N(\mathbf{p}))$



(a) Heat conduction problem setup. The top and left sides Γ_D are given a Dirichlet condition and the bottom and right sides Γ_N are given a Neumann condition.

(b) The right half of the MBB problem exploiting symmetry. The roller supports provide vertical restraint on the left-hand side and horizontal restraint in the bottom right corner. A downward force F is prescribed on the top left corner.

Figure 6: Benchmark problems

7.1.2 Optimized Structures

To compare the SIMP and LS results, the optimized density is converted to a LS by taking the 0.5 isosurface of the density and recomputing the objective function value.

The optimized geometries using the various methods are seen in Figure 7. In all cases, the neural parameterization results in more regular geometries. The pixel-based methods could be regularized by augmenting the objective function with a penalization term although this would require manual tuning of a penalization parameter and may have an impact on the convergence. Simplistic structures could also be obtained for the pixel-based cases by controlling the filter radius and mesh resolution although this would prevent fine-scale structure. The NN-LS method instead allows for fine-scale features resolved in the final layer of the network but still produces performant regular geometries as the multi-scale influence of the parameters in the neural parameterization encourages globally performant structures to emerge. The use of the NN is also seen to suppress numerical artifacts observed in the pixel-LS solutions. For the MBB problem, the compliance measurements for all methods fall within a few percent of each other. For the heat conduction problem, however, the NN-LS method has the best performance by a fair amount followed by the NN-SIMP baseline. The regular geometries produced by the NNs in these cases outperform their pixel counterparts by a significant amount in both the LS and SIMP cases. The U-Net here seems able to find better minima because of its ability to focus on larger scale structure while the pixel-based parameterization makes improvements locally by adding finer scale branches to the geometry.

7.1.3 Convergence Plots

The convergence of the methods is plotted for the heat conduction and MBB problems in Figure 8. Since the use of the word iteration in the context of optimization is somewhat ambiguous, we plot the compliance against the number of objective function value calls for the process. For the ADAM and MMA optimizers, the ratio of function and gradient

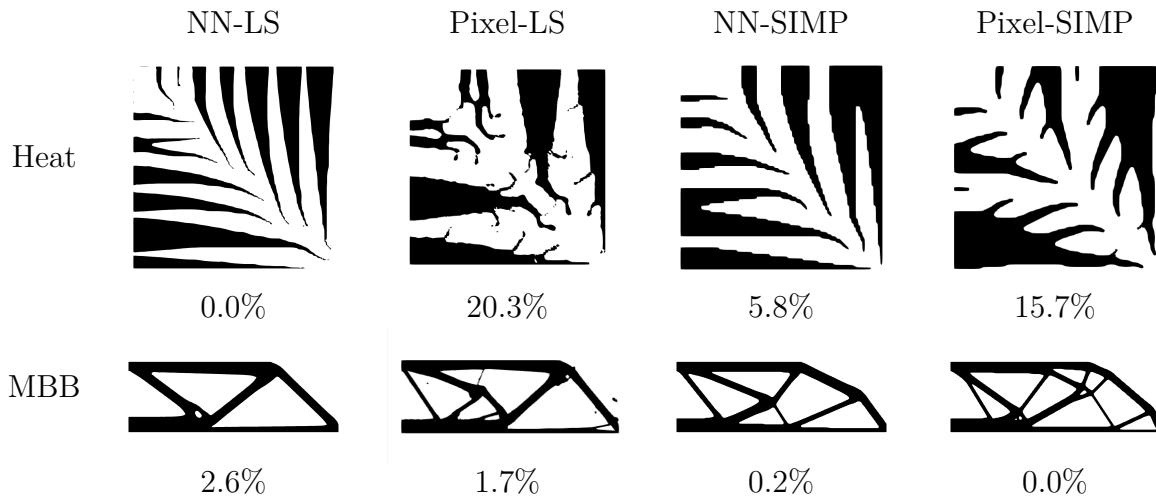


Figure 7: Optimized geometries for the various methods. The percentage under each geometry represents its performance relative to the best performing geometry in the row as measured by the objective function.

calls is 1:1. To compare the SIMP and LS methods against each other, we bias the SIMP method using the optimized structures converted LS once again. The bias is computed as the difference between the SIMP method final objective value using the interpolated material and the SIMP method final objective value using the converted LS. This bias is applied to all of the series data in Figures 8 for the SIMP methods.

For the heat conduction problem in Figure 8a, the pixel-SIMP method converges the fastest, albeit to an inferior solution. The rest of the methods converge at similar rates. For the MBB problem in Figure 8b, the SIMP methods converge much faster than the LS methods although the NN-SIMP method jumps out of the minimum at later iterations and stabilizes later on.

7.2 Beam Support Design

The design of a beam support in a fluid channel is optimized in this section with the setup in [64]. The goal is to demonstrate the generality of the method which is shown here by its capacity to solve a multiphysics problem with complex boundary conditions. Density methods do not extend naturally to handle such problems and are faced with difficulty in obtaining accurate coupling between the fluid and structure since the representation of the interface is spread across cells in the vicinity of the boundary.

The problem setup is seen in Figure 9. For this problem, we set the fluid parameters as $\mu = 1m^2s^{-1}$ and $\alpha_u = 2.5\mu/0.01^2$ and use a parabolic velocity profile on the inlet with an average velocity of $0.01ms^{-1}$. For the structural parameters, we set $E = 1Pa$ and $\nu = 0.3$ and use a 0.45 volume fraction. We use the same network as in the poisson problem but set $w = (12, 12, 24, 46, 96, 96)$ and $l = (24, 24, 48, 96, 192, 192)$. The optimized geometry for problem is seen in Figure 10.

8 CONCLUSION

Here we propose a neural LS method as a means of coupling convolutional NNs and the unfitted LS method. The neural parameterization provides more regular geometries and similar, in some cases better, performance compared to the well known SIMP method of TO and the pixel-based counterpart LS method. The neural parameterization learns

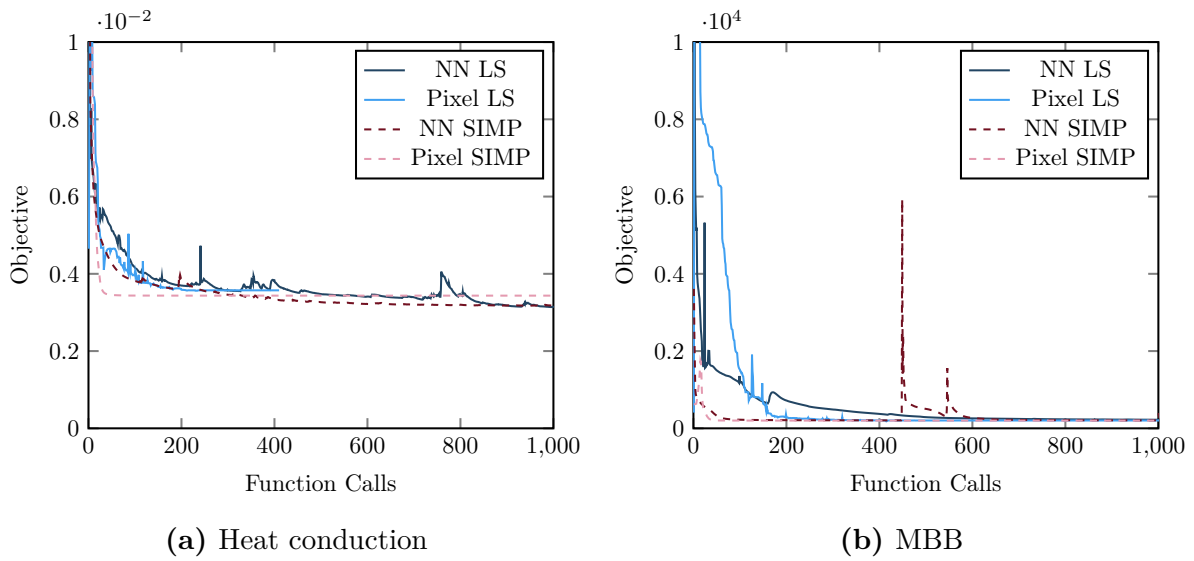


Figure 8: Convergence plot comparison for the benchmark problems. The objective value at each objective function call is plotted.

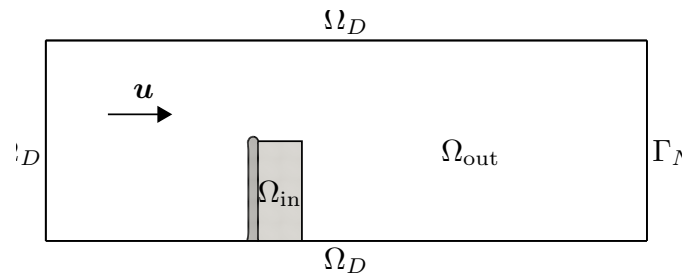


Figure 9: The beam support problem setup. An inlet velocity is prescribed on the left side, a homogenous Dirichlet condition on the top and bottom walls and a homogenous Neumann condition on the right wall Ω_N . The design region supporting the beam in light grey can be either fluid or solid. The solid domain Ω_{in} is composed of the dark grey beam and the solid part of the design region. The fluid region Ω_{out} is composed of the remainder of the channel, including the non-solid part of the design region.

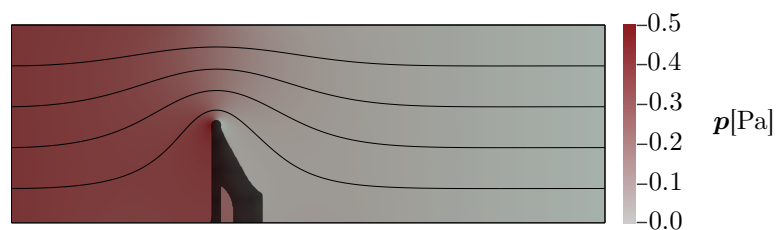


Figure 10: Optimized geometry for the beam support problem. Streamlines are plotted to represent the velocity field and pressure is indicated in the legend.

features at multiple scales during the optimization allowing for the emergence of regular structures whilst maintaining performance. The method takes longer to converge than the SIMP method and may not be suitable for simple problems such as those in linear elasticity. In this case, the network can be used in conjunction with the SIMP method. In contrast to SIMP, however, the method extends naturally to interface multiphysics problems and is more suitable in the general case. It should be noted that in our approach, the expressivity of geometries for the neural parameterization is the same as pixel counterpart methods since we use the same space V_h^1 in all cases to describe the geometry. A direction for development in the future would be to relax this constraint and also allow for the NN to control the space in which it expresses itself, potentially allowing for more efficient expressions of better topologies. Further investigation into architecture design and optimization strategy may also lead to a greater and more general improvement in performance.

9 REPLICATION OF RESULTS

All the codes being used in this paper are distributed as open source software. For reproducibility purposes, the implementation of the proposed TO methodology as well as the drivers being used to compare the different numerical methods are publicly available in the following repository: <https://github.com/ConnorMallon/NLSTO>

ACKNOWLEDGEMENTS

This research was partially funded by the Australian Government through the Australian Research Council (project number DP220103160).

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

- [1] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, Dec. 1989. doi:10.1007/bf01650949. URL <https://doi.org/10.1007/bf01650949>.
- [2] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, Aug. 2013. doi:10.1007/s00158-013-0978-6. URL <https://doi.org/10.1007/s00158-013-0978-6>.
- [3] X. Guo and G.-D. Cheng. Recent development in structural design and optimization. *Acta Mechanica Sinica*, 26(6):807–823, Dec. 2010. doi:10.1007/s10409-010-0395-7. URL <https://doi.org/10.1007/s10409-010-0395-7>.
- [4] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, Apr. 2001. doi:10.1007/s001580050176. URL <https://doi.org/10.1007/s001580050176>.
- [5] G. H. Yoon. Stress-based topology optimization method for steady-state fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 278:499–523, Aug. 2014. doi:10.1016/j.cma.2014.05.021. URL <https://doi.org/10.1016/j.cma.2014.05.021>.

- [6] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, Nov. 1988. doi:10.1016/0021-9991(88)90002-2. URL [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2).
- [7] J. A. Sethian. Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, 1999.
- [8] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, Mar. 2013. doi:10.1007/s00158-013-0912-y. URL <https://doi.org/10.1007/s00158-013-0912-y>.
- [9] S. J. Osher and F. Santosa. Level set methods for optimization problems involving geometry and constraints. *Journal of Computational Physics*, 171(1):272–288, July 2001. doi:10.1006/jcph.2001.6789. URL <https://doi.org/10.1006/jcph.2001.6789>.
- [10] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson. Shape optimization using the cut finite element method. *Computer Methods in Applied Mechanics and Engineering*, 328:242–261, Jan. 2018. doi:10.1016/j.cma.2017.09.005. URL <https://doi.org/10.1016/j.cma.2017.09.005>.
- [11] S.-H. Ha and S. Cho. Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Computers & Structures*, 86(13-14):1447–1455, July 2008. doi:10.1016/j.compstruc.2007.05.025. URL <https://doi.org/10.1016/j.compstruc.2007.05.025>.
- [12] S. Yamasaki, T. Nomura, A. Kawamoto, K. Sato, and S. Nishiwaki. A level set-based topology optimization method targeting metallic waveguide design problems. *International Journal for Numerical Methods in Engineering*, 87(9):844–868, Feb. 2011. doi:10.1002/nme.3135. URL <https://doi.org/10.1002/nme.3135>.
- [13] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, Feb. 2004. doi:10.1016/j.jcp.2003.09.032. URL <https://doi.org/10.1016/j.jcp.2003.09.032>.
- [14] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2):227–246, Jan. 2003. doi:10.1016/s0045-7825(02)00559-5. URL [https://doi.org/10.1016/s0045-7825\(02\)00559-5](https://doi.org/10.1016/s0045-7825(02)00559-5).
- [15] F. Dugast, Y. Favennec, and C. Josset. Reactive fluid flow topology optimization with the multi-relaxation time lattice boltzmann method and a level-set function. *Journal of Computational Physics*, 409:109252, May 2020. doi:10.1016/j.jcp.2020.109252. URL <https://doi.org/10.1016/j.jcp.2020.109252>.
- [16] J. Parvizian, A. Düster, and E. Rank. Topology optimization using the finite cell method. *Optimization and Engineering*, 13(1):57–78, July 2011. doi:10.1007/s11081-011-9159-x. URL <https://doi.org/10.1007/s11081-011-9159-x>.

- [17] E. Burman, P. Hansbo, M. G. Larson, and S. Zahedi. Cut finite element methods for coupled bulk–surface problems. *Numerische Mathematik*, 133(2):203–231, July 2015. doi:10.1007/s00211-015-0744-3. URL <https://doi.org/10.1007/s00211-015-0744-3>.
- [18] S. Badia, A. F. Martin, and F. Verdugo. Mixed Aggregated Finite Element Methods for the Unfitted Discretization of the Stokes Problem. *SIAM Journal on Scientific Computing*, 40(6):B1541–B1576, jan 2018. ISSN 1064-8275. doi:10.1137/18M1185624. URL <https://epubs.siam.org/doi/10.1137/18M1185624>.
- [19] S. Kreissl and K. Maute. Levelset based fluid topology optimization using the extended finite element method. *Structural and Multidisciplinary Optimization*, 46(3):311–326, Mar. 2012. doi:10.1007/s00158-012-0782-8. URL <https://doi.org/10.1007/s00158-012-0782-8>.
- [20] C. H. Villanueva and K. Maute. CutFEM topology optimization of 3d laminar incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 320:444–473, June 2017. doi:10.1016/j.cma.2017.03.007. URL <https://doi.org/10.1016/j.cma.2017.03.007>.
- [21] E. Burman. Ghost penalty. *Comptes Rendus Mathématique*, 348(21-22):1217–1220, Nov. 2010. doi:10.1016/j.crma.2010.10.006. URL <https://doi.org/10.1016/j.crma.2010.10.006>.
- [22] S. Badia, F. Verdugo, and A. F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533–553, jul 2018. ISSN 00457825. doi:10.1016/j.cma.2018.03.022. URL <https://linkinghub.elsevier.com/retrieve/pii/S0045782518301476>.
- [23] S. Badia, E. Neiva, and F. Verdugo. Linking ghost penalty and aggregated unfitted methods. *Computer Methods in Applied Mechanics and Engineering*, 388:114232, Jan. 2022. doi:10.1016/j.cma.2021.114232. URL <https://doi.org/10.1016/j.cma.2021.114232>.
- [24] S. Badia, E. Neiva, and F. Verdugo. Robust high-order unfitted finite elements by interpolation-based discrete extension. *Computers & Mathematics with Applications*, 127:105–126, Dec. 2022. doi:10.1016/j.camwa.2022.09.027. URL <https://doi.org/10.1016/j.camwa.2022.09.027>.
- [25] S. Kreissl, G. Pingen, and K. Maute. An explicit level set approach for generalized shape optimization of fluids with the lattice boltzmann method. *International Journal for Numerical Methods in Fluids*, 65(5):496–519, Jan. 2011. doi:10.1002/flid.2193. URL <https://doi.org/10.1002/flid.2193>.
- [26] N. Dijk, M. Langelaar, and F. Keulen. Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 91(1):67–97, May 2012. doi:10.1002/nme.4258. URL <https://doi.org/10.1002/nme.4258>.
- [27] H. Adeli and H. S. Park. A neural dynamics model for structural optimization—theory. *Computers & Structures*, 57(3):383–390, Nov. 1995. doi:10.1016/0045-7949(95)00048-1. URL [https://doi.org/10.1016/0045-7949\(95\)00048-1](https://doi.org/10.1016/0045-7949(95)00048-1).

- [28] Z. Zhang, Y. Li, W. Zhou, X. Chen, W. Yao, and Y. Zhao. TONR: An exploration for a novel way combining neural network with topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 386:114083, Dec. 2021. doi:10.1016/j.cma.2021.114083. URL <https://doi.org/10.1016/j.cma.2021.114083>.
- [29] R. V. Woldseth, N. Aage, J. A. Bærentzen, and O. Sigmund. On the use of artificial neural networks in topology optimisation, 2022. URL <https://arxiv.org/abs/2208.02563>.
- [30] V.-N. Hoang, N.-L. Nguyen, D. Q. Tran, Q.-V. Vu, and H. Nguyen-Xuan. Data-driven geometry-based topology optimization. *Structural and Multidisciplinary Optimization*, 65(2), Jan. 2022. doi:10.1007/s00158-022-03170-8. URL <https://doi.org/10.1007/s00158-022-03170-8>.
- [31] Y. Yu, T. Hur, J. Jung, and I. G. Jang. Deep learning for determining a near-optimal topological design without any iteration. *Structural and Multidisciplinary Optimization*, 59(3):787–799, Oct. 2018. doi:10.1007/s00158-018-2101-5. URL <https://doi.org/10.1007/s00158-018-2101-5>.
- [32] B. Li, C. Huang, X. Li, S. Zheng, and J. Hong. Non-iterative structural topology optimization using deep learning. *Computer-Aided Design*, 115:172–180, Oct. 2019. doi:10.1016/j.cad.2019.05.038. URL <https://doi.org/10.1016/j.cad.2019.05.038>.
- [33] S. Zheng, Z. He, and H. Liu. Generating three-dimensional structural topologies via a u-net convolutional neural network. *Thin-Walled Structures*, 159:107263, Feb. 2021. doi:10.1016/j.tws.2020.107263. URL <https://doi.org/10.1016/j.tws.2020.107263>.
- [34] N. A. Kallioras, G. Kazakis, and N. D. Lagaros. Accelerated topology optimization by means of deep learning. *Structural and Multidisciplinary Optimization*, 62(3):1185–1212, Mar. 2020. doi:10.1007/s00158-020-02545-z. URL <https://doi.org/10.1007/s00158-020-02545-z>.
- [35] Y. Joo, Y. Yu, and I. G. Jang. Unit module-based convergence acceleration for topology optimization using the spatiotemporal deep neural network. *IEEE Access*, 9:149766–149779, 2021. doi:10.1109/access.2021.3125014. URL <https://doi.org/10.1109/access.2021.3125014>.
- [36] H. Deng and A. C. To. Topology optimization based on deep representation learning (DRL) for compliance and stress-constrained design. *Computational Mechanics*, 66(2):449–469, May 2020. doi:10.1007/s00466-020-01859-5. URL <https://doi.org/10.1007/s00466-020-01859-5>.
- [37] A. Chandrasekhar and K. Suresh. TOuNN: Topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, 63(3):1135–1149, Nov. 2020. doi:10.1007/s00158-020-02748-4. URL <https://doi.org/10.1007/s00158-020-02748-4>.
- [38] S. Hoyer, J. Sohl-Dickstein, and S. Greydanus. Neural reparameterization improves structural optimization, 2019.

- [39] A. R. Barron. Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14(1):115–133, Jan. 1994. doi:10.1007/bf00993164. URL <https://doi.org/10.1007/bf00993164>.
- [40] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.
- [41] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. 128(7):1867–1888, Mar. 2020. doi:10.1007/s11263-020-01303-4. URL <https://doi.org/10.1007/s11263-020-01303-4>.
- [42] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi:10.1137/141000671.
- [43] S. Badia and F. Verdugo. Gridap: An extensible finite element toolbox in julia. *Journal of Open Source Software*, 5(52):2520, 2020. doi:10.21105/joss.02520. URL <https://doi.org/10.21105/joss.02520>.
- [44] F. Verdugo and S. Badia. The software design of gridap: A finite element package based on the julia JIT compiler. *Computer Physics Communications*, 276:108341, July 2022. doi:10.1016/j.cpc.2022.108341. URL <https://doi.org/10.1016/j.cpc.2022.108341>.
- [45] M. Innes, E. Saba, K. Fischer, D. Gandhi, M. C. Rudilosso, N. M. Joy, T. Karmali, A. Pal, and V. Shah. Fashionable modelling with flux. *CoRR*, abs/1811.01457, 2018. URL <https://arxiv.org/abs/1811.01457>.
- [46] H. Deng and A. C. To. A parametric level set method for topology optimization based on deep neural network. *Journal of Mechanical Design*, 143(9), Mar 2021. ISSN 1528-9001. doi:10.1115/1.4050105. URL <http://dx.doi.org/10.1115/1.4050105>.
- [47] D. RUMELHART, G. HINTON, and R. WILLIAMS. Learning internal representations by error propagation. In *Readings in Cognitive Science*, pages 399–421. Elsevier, 1988. doi:10.1016/b978-1-4832-1446-7.50035-2. URL <https://doi.org/10.1016/b978-1-4832-1446-7.50035-2>.
- [48] J. L. Barrera, M. J. Geiss, and K. Maute. Hole seeding in level set topology optimization via density fields. *Structural and Multidisciplinary Optimization*, 61(4):1319–1343, Feb. 2020. doi:10.1007/s00158-019-02480-8. URL <https://doi.org/10.1007/s00158-019-02480-8>.
- [49] X. Xing, P. Wei, and M. Y. Wang. A finite element-based level set method for structural optimization. *International Journal for Numerical Methods in Engineering*, 82(7):805–842, Nov. 2009. doi:10.1002/nme.2785. URL <https://doi.org/10.1002/nme.2785>.
- [50] S. Badia, F. Verdugo, and A. F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533–553, July 2018. doi:10.1016/j.cma.2018.03.022. URL <https://doi.org/10.1016/j.cma.2018.03.022>.

- [51] S. Badia, P. A. Martorell, and F. Verdugo. Geometrical discretisations for unfitted finite elements on explicit boundary representations. *Journal of Computational Physics*, 460:111162, July 2022. doi:10.1016/j.jcp.2022.111162. URL <https://doi.org/10.1016/j.jcp.2022.111162>.
- [52] F. de Prenter, C. Verhoosel, G. van Zwieten, and E. van Brummelen. Condition number analysis and preconditioning of the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 316:297–327, Apr. 2017. doi:10.1016/j.cma.2016.07.006. URL <https://doi.org/10.1016/j.cma.2016.07.006>.
- [53] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, Apr. 2007. doi:10.1007/s00466-007-0173-y. URL <https://doi.org/10.1007/s00466-007-0173-y>.
- [54] T. Borrvall and J. Petersson. Topology optimization of fluids in stokes flow. *International Journal for Numerical Methods in Fluids*, 41(1):77–107, 2002. doi:10.1002/fld.426. URL <https://doi.org/10.1002/fld.426>.
- [55] G. Allaire, C. Dapogny, and F. Jouve. Shape and topology optimization. In *Geometric Partial Differential Equations - Part II*, pages 1–132. Elsevier, 2021. doi:10.1016/bs.hna.2020.10.004. URL <https://doi.org/10.1016/bs.hna.2020.10.004>.
- [56] A. Sharma, H. Villanueva, and K. Maute. On shape sensitivities with heaviside-enriched XFEM. *Structural and Multidisciplinary Optimization*, 55(2):385–408, Dec. 2016. doi:10.1007/s00158-016-1640-x. URL <https://doi.org/10.1007/s00158-016-1640-x>.
- [57] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. Cutfem: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015. doi:10.1002/nme.4823. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823>.
- [58] C. Lang, D. Makhija, A. Doostan, and K. Maute. A simple and efficient preconditioning scheme for heaviside enriched XFEM. *Computational Mechanics*, 54(5):1357–1374, Aug. 2014. doi:10.1007/s00466-014-1063-8. URL <https://doi.org/10.1007/s00466-014-1063-8>.
- [59] F. Verdugo and S. Badia. The software design of gridap: a finite element package based on the julia jit compiler. 2021. doi:10.48550/ARXIV.2109.12818. URL <https://arxiv.org/abs/2109.12818>.
- [60] J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in julia, 2016. URL <https://arxiv.org/abs/1607.07892>.
- [61] K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, Feb. 1987. doi:10.1002/nme.1620240207. URL <https://doi.org/10.1002/nme.1620240207>.
- [62] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014. URL <https://arxiv.org/abs/1412.6980>.

- [63] A. Gersborg-Hansen, M. P. Bendsøe, and O. Sigmund. Topology optimization of heat conduction problems using the finite volume method. *Structural and Multidisciplinary Optimization*, 31(4):251–259, Mar. 2006. doi:10.1007/s00158-005-0584-3. URL <https://doi.org/10.1007/s00158-005-0584-3>.
- [64] N. Jenkins and K. Maute. An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. *Structural and Multidisciplinary Optimization*, 54(5):1191–1208, May 2016. doi:10.1007/s00158-016-1467-5. URL <https://doi.org/10.1007/s00158-016-1467-5>.



DESIGN OPTIMIZATION OF TRUSS STRUCTURES USING A NON- UNIFORM CELLULAR AUTOMATA PARADIGM

Mohamed El Bouzouiki, Ramin Sedaghati and Ion Stiharu

Department of Mechanical, Industrial and Aerospace Engineering
Gina Cody School of Engineering and Computer Science
Concordia University
1455 de Maisonneuve Blvd. West, Montreal, Quebec, H3G 1M8.
m_elbouz@encs.concordia.ca, ramin.sedaghati@concordia.ca, ion.stiharu@concordia.ca
<https://www.concordia.ca>.

Abstract *The conventional uniform Cellular Automata (CA) approach requires identical uniform cells (all cells must have the same number of neighbouring nodes) for design optimization of truss structures. As in CA every node of a truss structure represents a cell (one-to-one relationship between nodes and cells), conventional CA has difficulties to address nodes located on the boundaries and also to consider the position of the nodes as design variable for layout optimization. In this study, a modified non-uniform Cellular Automata (CA) algorithm for the optimal topology, sizing, and layout design of truss structures has been proposed. The proposed approach builds upon the success of our previous study that utilized non-uniform CA for a sizing and topology optimization of truss structures under stress and displacement constraints. Specifically, the new algorithm incorporates the Fully Stressed Design (FSD) method and the distribution of strain energy to identify optimal cell coordinates. By considering simultaneous sizing, topology and layout optimization, the proposed non-uniform CA algorithm can effectively minimize the weight of truss structures subjected to displacement and stress constraints. The efficacy and accuracy of this approach are demonstrated through its successful application to various benchmark truss design problems.*

Keywords: Non-uniform Cellular Automata, Sizing, topology and layout Design Optimization, Truss Structures, Stress and Displacement Constraints

1. INTRODUCTION

Optimizing truss structures involves finding the best connection, member sizes, and shape to minimize weight and cost while improving strength and stability. Various methods, including mathematical programming, convex approximation, fully stressed design (FSD), and metaheuristic techniques, have been used to optimize sizing and layout variables. Single-level methods simultaneously consider these variables, while bi-level methods solve for joint coordinates and cross-sectional areas separately. The present study proposes a non-uniform cellular automata (CA) algorithm for truss optimization that minimizes strain energy and redistributes it evenly within the structure. Cross-sectional areas and node coordinates are optimized separately using the FSD approach in conjunction with a strain energy criterion. An alternating procedure couples these variables until the optimal solution is found.

2. PROBLEM FORMULATION

This study aims to optimize the topology, sizing, and layout of a structure to minimize the weight of a truss structure under stress and displacement constraints. The optimization problem can be formally formulated as follows:

$$\text{minimize } W(P) = \sum_{j=1}^m \rho_j A_j L_j \quad (1a)$$

$$\text{Subject to } \mathbf{K}u = \mathbf{f} \quad (1b)$$

$$|\sigma_j| < \bar{\sigma}_j, \quad j = 1, 2, \dots, m \quad (1c)$$

$$|u_i| < \bar{u}_i, \quad i = 1, 2, \dots, k \quad (1d)$$

$$\bar{A}^l < A_j < \bar{A}^u, \quad j = 1, 2, \dots, m \quad (1e)$$

$$|x_i| < \bar{x}_i, \quad i = 1, 2, \dots, k \quad (1d)$$

where m is the total number of bar members in the structure. \mathbf{K} is the stiffness matrix, \mathbf{u} is the vector of nodal displacements, \mathbf{f} is the vector of nodal forces, and A_j are the cross-sectional area of the truss members. Constraint (1b) represents the equilibrium condition, while constraints (1c)-(1e) represent stress, displacement, and side constraints. Allowable stresses and displacements are denoted by $\bar{\sigma}_j$ and \bar{u}_i , respectively. For buckling constraints, $\bar{\sigma}_j$ is set to the smallest value between allowable compressive stress and Euler buckling stress. The upper and lower bounds on the cross-sectional areas are \bar{A}^u and \bar{A}^l , respectively, and k states the number of displacement constraints. To ensure kinematic stability, a very small positive value is used as the lower bound of the cross-sectional area. Topology optimization is achieved by removing members with a cross-sectional area equal to their lower bound in the final optimal configuration.

3. NON-UNIFORM CELLULAR AUTOMATA

This study extends a prior approach used for topology and sizing optimization of truss structures by adding layout optimization using the non-uniform Cellular Automata (CA) algorithm [1]. Non-identical cells, consisting of a center node and members linking it to nearby nodes, are used. Fig. 1(a) shows a two-dimensional truss structure represented by the proposed non-uniform CA, while Fig. 1(b) displays a boundary unit cell. The algorithm optimizes node positions and member cross-sectional areas to find the minimum weight structure.

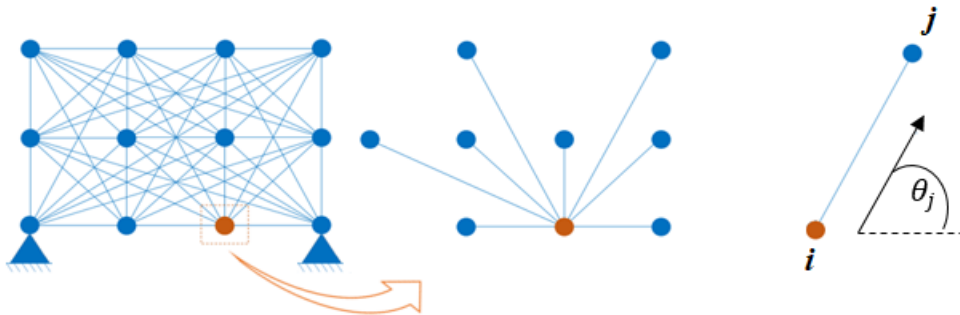


Figure 1. (a) Non-uniform CA representation of a two-dimensional truss structure, (b) Representation of the non-uniform unit cell located on the boundary

Analysis update rules:

After each design solution, equilibrium at the cell level is verified by applying analysis update rules. The field variables $\{u_i, v_i\}$ and $\{F_{xi}, F_{yi}\}$ are determined by minimizing the potential

energy of the cell [1]. The equilibrium equations for cell i are expressed as:

$$\sum_{j=1}^N \frac{E_j A_j}{L_j} \cos^2 \theta_j u_i + \sum_{j=1}^N \frac{E_j A_j}{L_j} \sin \theta_j \cos \theta_j v_i = F_{xi} + \sum_{j=1}^N \frac{E_j A_j}{L_j} (\cos^2 \theta_j u_j + \sin \theta_j \cos \theta_j v_j) \quad (2)$$

$$\sum_{j=1}^N \frac{E_j A_j}{L_j} \sin \theta_j \cos \theta_j u_i + \sum_{j=1}^N \frac{E_j A_j}{L_j} \sin^2 \theta_j v_i = F_{yi} + \sum_{j=1}^N \frac{E_j A_j}{L_j} (\sin \theta_j \cos \theta_j u_j + \sin^2 \theta_j v_j)$$

Design update rules:

To optimize the sizing variables, this study employs the Fully Stressed Design approach (FSD) [2]. Additionally, FSD is utilized along with a strain energy criterion to optimize the layout variables. It is assumed that in an optimally designed structure, the strain energy is uniformly distributed, and all members should attain their stress limit. Consequently, the update rule for the member's cross-sectional areas is expressed as:

$$A_j^{t+1} = A_j^t \frac{|\sigma_j^t|}{\bar{\sigma}_j} \quad (3)$$

The total strain energy of the structure and the individual cell's contribution to the total strain energy may also be written as:

$$U_T = \sum_{j=1}^m \frac{\sigma_j^2 L_j A_j}{2E_j} \quad (4)$$

$$U_c^i = \frac{1}{2} \sum_{j=1}^N \frac{\sigma_j^2 L_j A_j}{2E_j} \quad (5)$$

Here m represents the total number of members in the structure, and N is the total number of members within a specific cell. The $\frac{1}{2}$ factor is introduced for consistency so that the sum of cells' strain energy is equal to U_T . Assuming all members are at their maximum stress, minimizing the total strain energy by shifting the nodes is equivalent to minimizing the total weight. Hence, the change in total weight and total strain energy due to a shift in the coordinate x_i of a given node can be expressed as:

$$\frac{dW_T}{dx_i} = \rho \frac{d}{dx_i} \left(\sum_{j=1}^m L_j A_j \right) \quad (6)$$

$$\frac{dU_T}{dx_i} = \frac{\bar{\sigma}^2}{2E} \frac{d}{dx_i} \left(\sum_{j=1}^m L_j A_j \right) = \frac{\bar{\sigma}^2}{2\rho E} \frac{dW_T}{dx_i} \quad (7)$$

Thus, to minimize the total weight of the structure, the nodes are shifted in the direction that minimizes the total strain energy. FSD is applied after each change in joint coordinates to ensure the optimization moves towards minimizing the total weight (descent direction). The algorithm calculates the shift Δx_i of a given node (cell) i using the following equation:

$$\Delta x_i = -\text{sign} \left(\frac{dU_T}{dx_i} \right) \text{sign} \left(\frac{(\Delta U_c^i)^t}{(\Delta U_c^i)^0} \right) \min \left(\alpha \left| \frac{(\Delta U_c^i)^t}{\left(\frac{dU_c^i}{dx_i} \right)^t} \right|, \beta L_{max} \right) \quad (8)$$

where

$$(\Delta U_c^i)^t = (U_c^i)^t - \bar{U}_c^t \quad (9)$$

Constant parameters α and β which determine the extent of the displacement in the design space, play a crucial role in the algorithm's convergence.

4. TEST PROBLEMS AND OPTIMIZATION RESULTS

An-object oriented has been effectively used to implement the proposed bi-level non-uniform CA approach on various benchmark truss structures to verify its efficiency and accuracy. Both cross-sectional areas and coordinates of nodes are considered as design variables for sizing and layout optimization. It is noted that the topology optimization has been indirectly implemented by removing truss members who cross-sectional areas have converged to lower bounds which are set to very small values.

4.1. Problem 2: 15-Member, 8-Node Truss Structure

The 15-bar, 8-node planar truss test problem depicted in Fig. 2(a) was previously solved by Rahami et al. [3] using a single-level Genetic Algorithm (GA) and by Gholizadeh [4] using a hybrid bi-level optimization algorithm that combines Cellular Automata (CA) with Particle Swarm Optimization (PSO). Material properties and problem parameters are available in [3,4].

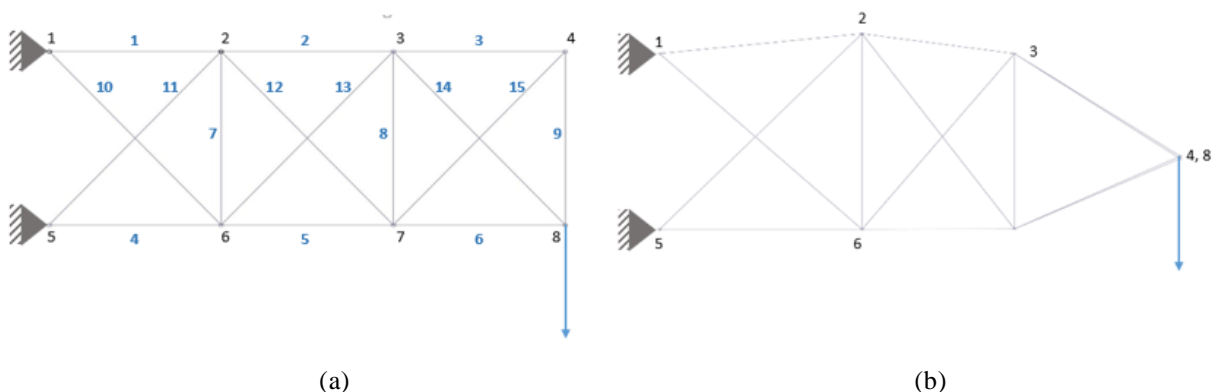


Figure 2. (a) 15-bar, 8-node truss structure and (b) the Optimized geometry.

The proposed non-uniform CA approach generated an optimized layout as shown in Fig. 2(b) which is comparable to those reported by other methods. Rahami et al. [3] used a single-level GA and found an optimal weight of 76.6854 lb after 8000 structural analyses, while Gholizadeh [4] used a hybrid bi-level optimization algorithm combining CA with PSO, and obtained an optimal weight of 72.5143 lb with 4500 structural analyses. The proposed methodology achieved a total optimal weight of 69.176 lb, after only 240 structural analyses. It is important to note that none of the constraints were violated during the optimization process.

4.2. Test Problem 5: 77-bar, 40-node truss bridge

The problem involves a 77-member, 40-node bridge structure depicted in Fig. 3. The bridge is 6000 in long and has 21 evenly spaced panel joints at intervals of 300 in. Flager et al. [5] solved this problem using a single geometry variable - the y-coordinate of the top joints of the truss - and discrete sizing variables selected from a list of W-shape profiles. Forces acting on the bridge include self-weight and loads specified in [5].

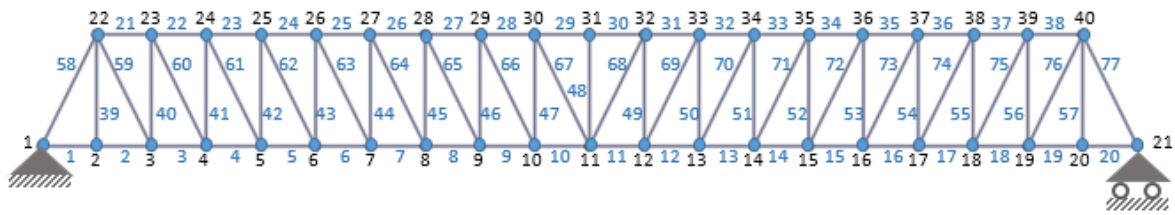


Figure 2 77-bar, 40-node truss bridge

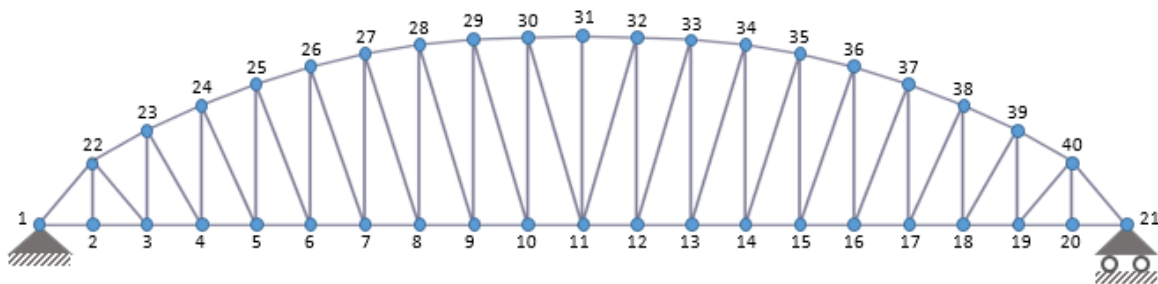


Figure 4. Optimized truss for the 77-bar space truss

The optimal layout of the minimum weight truss structure is depicted in Fig. 4. Flager et al. [5] performed 7553 structural analyses to obtain an overall weight of 511,037 lb, while the proposed method achieved a much lower weight of 320038 lb after 5707 structural analyses without violating any stress or displacement constraints.

5. CONCLUSION

This study proposes a bi-level algorithm that utilizes a non-uniform cellular automata method to simultaneously optimize the sizing, topology, and layout of truss structures under stress and displacement constraints. By coupling sizing and layout in an alternating procedure, the proposed design optimization methodology was successfully implemented in various bench mark problem. The algorithm's ability to handle complex problems and obtain optimal solutions with fewer structural analyses contributes to its superiority.

ACKNOWLEDGMENTS

Support from Gina Cody School of Engineering and Computer Science at Concordia University is gratefully acknowledged.

REFERENCES

- [1] El Bouzouiki M, Sedaghati R, Stiharu I. A non-uniform cellular automata framework for topology and sizing optimization of truss structures subjected to stress and displacement constraints. *Computers & Structures* 2021;242:106394.
- [2] Gallagher, R. H., & In Zienkiewicz, O. C. (1973). *Optimum structural design: Theory and applications*. London: Wiley. D. P. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA Education Series, 5th edition, 2012. ISBN 9781600869112.

- [3] Rahami H, Kaveh A, Gholipour Y. Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures* 2008;30:2360–9.
- [4] Gholizadeh S. Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization. *Computers & Structures* 2013;125:86–99.
- [5] F. Flager, A. Adya, J. Haymaker, and M. Fischer, "A bi-level hierarchical method for shape and member sizing optimization of steel truss structures," *Computers & Structures*, vol. 131, pp. 1-11, 2014.



STRESS-BASED SPATIAL GRADIENT RECONSTRUCTION FOR SHAPE SENSITIVITY ANALYSIS

Robert A. Canfield

Kevin T. Crofton Department of Aerospace and Ocean Engineering
College of Engineering
Virginia Tech
1600 Innovation Dr., Blacksburg, VA 24061
bob.canfield@vt.edu, <https://www.aoe.vt.edu/people/faculty/canfield.html>

Abstract. *Continuum Sensitivity Analysis (CSA) is an approach for calculating analytic design derivatives. Spatial derivatives of the response, at least on the boundary, are needed to formulate CSA for shape design variables that alter the shape of a structure. Inaccurate spatial derivatives on the boundaries has historically hampered the use of the so-called boundary velocity form of shape CSA. Spatial Gradient Reconstruction (SGR) has proven to be a remedy for inaccurate spatial derivatives of structural response, at least for displacements and their derivatives; however, SGR for stresses on the boundary where surface tractions may be applied, continues to be problematic. Here stress-based SGR, used in the literature for stress error estimation, is developed for shape CSA. Stress-based SGR with and without enforcement of traction boundary conditions are compared with stresses and stress gradients recovered from displacement-based SGR. Traction enforcement, together with equilibrium constraints, is found here to improve stress-based SGR, making it more accurate than without traction enforcement and more accurate than stress gradients recovered from displacement-based SGR.*

Keywords: continuum sensitivity, shape derivatives, sensitivity analysis, two-and three-dimensional structural analysis, nonintrusive

1 INTRODUCTION

Continuum Sensitivity Analysis (CSA) has been developed to compute gradients for shape optimization of structural response [1–4] and fluid response [5–8]. The motivation for using CSA for shape sensitivity is twofold: (a) gradients are computed from analytic expressions and are therefore more accurate and efficient than finite difference, and (b) mesh sensitivity is avoided, which is a drawback of the discrete analytic shape sensitivity approach.

As will be shown in the review of CSA, the boundary-velocity formulation involving local shape derivatives depends heavily upon spatial derivatives of the response from the structural analysis solution. Often those very derivatives of the numerical solution are of poor quality; however, a Spatial Gradient Reconstruction (SGR) method has been shown in the literature [9–13] to overcome this challenge. Nevertheless, this SGR method was based on recovering spatial derivatives from displacements of the solution. SGR based on reconstruction from stresses of the solution offers an avenue for further improvement for shape CSA of structures where tractions are applied as loads on boundary surfaces that change shape. Zienkiewicz and Zhu’s superconvergent patch recovery (SPR) method [14], heretofore has been used for SGR; however, the enhancement of SPR developed by Wiberg *et al.*, whereby traction boundary conditions (and static equilibrium) are added the least-squares solution for SPR [15], is a promising approach for SGR used for CSA. Babuska *et al.* [16] concluded that those “enhancements can lead to a significant deterioration of the asymptotic robustness of the estimator.” However, Babuska’s measure of robustness was an energy norm of the solution for the displacement or stress field, whereas for CSA spatial derivatives of those fields need to be reconstructed by SGR. Therefore, we study here the accuracy of SGR based on Wiberg’s enhanced SPR for those derivative terms it provides for CSA sensitivity loads.

After reviewing the governing structural equations, their finite element solution, and conventional discrete sensitivity analysis in Section 2, the boundary-velocity (local) continuum sensitivity equations for CSA are summarized in Section 3. The SGR used with CSA, which is the subject of the numerical studies in this paper, is reviewed in Section 4. A two-dimensional cantilever beam example is presented in Section 5. Convergence studies for SGR of stress components of that Timoshenko beam and their spatial derivatives, presented in Section 6 and the Appendices, demonstrate that stress-based SGR with traction enforcement improves the accuracy of recovered stresses and their spatial derivatives.

2 GOVERNING STRUCTURAL ANALYSIS EQUATIONS

2.1 Continuous Domain Equations and Boundary Conditions

The partial differential equation governing structural response can be written compactly as

$$\mathbf{R}(\mathbf{u}, t; b) = \mathcal{A}(\mathbf{u}, L(\mathbf{u})) - \mathbf{f}(\mathbf{x}, t; b) = 0 \quad \text{on} \quad \Omega, \quad (1)$$

with the corresponding boundary conditions (BCs)

$$\mathcal{B}(\mathbf{u}, L(\mathbf{u})) = \mathbf{g}(\mathbf{x}, t; b) \quad \text{on} \quad \Gamma(b), \quad (2)$$

where the vector of dependent (state) variables $\mathbf{u}(\mathbf{x}, t; b)$ are functions of the spatial and temporal independent coordinates, \mathbf{x} and t , respectively and depend implicitly on design

variable b . Since the design variable b changes the shape of the domain, the boundary can also be represented as $\Gamma(b)$. The linear differential operator L has terms such as $\left\{ \frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial y^2}, \dots \right\}$ that appear in the governing equations or boundary conditions. \mathcal{A} and \mathcal{B} are algebraic or integral operators acting on \mathbf{u} and $L(\mathbf{u})$, possibly in a nonlinear fashion. The distributed body force applied on the system is given by \mathbf{f} in (Eq.(1)), and the general BCs in (Eq.(2)) can be either Dirichlet (essential or geometric) such as a prescribed value

$$\mathcal{B}_e(\mathbf{u}) \equiv \mathbf{u}|_{\Gamma_e} = \mathbf{g}_e \equiv \bar{\mathbf{u}} \quad (3)$$

on the boundary Γ_e , or they may involve a differential operator for Neumann (nonessential or natural) BCs such that

$$\mathcal{B}_n(L(\mathbf{u}))|_{\Gamma_n} = \text{mat}g_n \quad (4)$$

on the boundary Γ_n .

2.2 Finite Element Discretization of the Structural Analysis Equations

The continuous partial differential Eqs.(1)) for a structure commonly are integrated by energy methods or weighted residual methods to create a discrete finite element model (FEM), $[K]\{\mathbf{u}\} = \{\mathbf{F}\}$, that can be solved to obtain the structural response $\{\mathbf{u}\}$. This is the response analysis. The discrete FEM equations are partitioned as

$$\begin{bmatrix} \mathbf{K}_{ee} & \mathbf{K}_{ef} \\ \mathbf{K}_{fe} & \mathbf{K}_{ff} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_e \\ \mathbf{u}_f \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_e \\ \mathbf{F}_f \end{Bmatrix}. \quad (5)$$

Here the notation $\{\mathbf{u}\}$ stands for a discretized version of the continuous solution \mathbf{u} on the domain Ω . The vector $\{\mathbf{u}_e\}$ corresponds to the degrees of freedom that are constrained by essential (or geometric) boundary conditions on Γ_e . The vector $\{\mathbf{u}_f\}$ consists of the degrees of freedom $\{\mathbf{u}_n\}$ constrained by the natural boundary conditions on the boundary Γ_n and the unconstrained degrees of freedom $\{\mathbf{u}_\Omega\}$ in the interior of the domain Ω . The partitioned Eq.(5) is re-arranged to move the prescribed terms involving essential and non-essential boundary conditions to the right side to represent the system of equations in abbreviated fashion to be solved for the primary unknowns, the free displacements, $\{\mathbf{u}\} = \{\mathbf{u}_f\}$

$$[K]\{\mathbf{u}\} = \{\mathbf{F}\} \quad (6)$$

where $\{\mathbf{F}\} = \{\mathbf{F}_f - \mathbf{K}_{fe}\mathbf{u}_e\}$.

2.3 Discrete Sensitivity Analysis (DSA)

The direct method of Discrete Sensitivity Analysis (DSA) differentiates Eq.(6) directly to solve for the displacement sensitivity $\dot{\mathbf{u}}$

$$\{\dot{\mathbf{u}}\} = [K]^{-1}(\dot{\mathbf{F}} - [\dot{K}]\{\mathbf{u}\}) = [K]^{-1}\{\mathbf{P}\} \quad (7)$$

where $\{\mathbf{P}\} \equiv \{\dot{\mathbf{F}}\} - [\dot{K}]\{\mathbf{u}\} = \{\dot{\mathbf{F}}_f\} - [\dot{K}_{fe}]\{\mathbf{u}_e\} - [\mathbf{K}_{fe}]\{\dot{\mathbf{u}}_e\} - [\dot{K}_{ff}]\{\mathbf{u}_f\}$ is known as the discrete sensitivity pseudo-load and the notation $(\dot{})$ denotes the total (material) derivative with respect to the shape design variable, b . Since the stiffness matrix $[K]$ depends upon nodal coordinates that are a function of the shape design variable, the stiffness sensitivity $[\dot{K}]$ necessarily involves mesh sensitivity, *i.e.*, how spatial coordinates

change with a design variable throughout the domain Ω . Mesh sensitivity is often found by solving an auxiliary problem, say by the pseudo-solid method [17]. Boundary-Velocity CSA avoids mesh sensitivity analysis.

3 CONTINUUM SENSITIVITY ANALYSIS (CSA)

3.1 Differentiation of the Continuous Equations

CSA is a method to obtain the derivative of the response $\mathbf{u}(\mathbf{x}; b)$ with respect to design parameter b , possibly at all points in the domain. The response depends on the spatial variable \mathbf{x} and it may have an explicit or implicit dependence on the shape variable b , as indicated by the semicolon. CSA is based on the philosophy of “differentiate and then discretize.” First, Eqs.(1)) and ((2)) are differentiated with respect to b , leading to the associated Continuum Sensitivity Equations (CSEs), which are then discretized. The resulting discrete CSEs are then solved in a similar fashion as the original response equations, *e.g.* by Finite Element Analysis (FEA).

Based on the type of differentiation, CSA is categorized as either local form CSA, also known as *boundary-velocity* form, or total (material derivative) form CSA, also known as *domain velocity* form [18]. Regardless of the form, just like DSA, the resulting CSEs are always a linear in terms of sensitivity variable \mathbf{u}' , even if the governing Eqs.(1)) are nonlinear. Also, regardless of the form, since the material boundary changes due to a change in the shape design parameter b , the CSA boundary conditions are obtained by total or material differentiation of the original boundary conditions ((2)).

Differentiating Eqs.(1)) and their boundary conditions ((2)) gives the CSEs

$$\frac{\partial \mathbf{R}}{\partial b} = \mathcal{A}_b(\mathbf{u}, L(\mathbf{u}')) - \mathbf{f}'(\mathbf{x}, t; b) = 0 \quad \text{on } \Omega, \quad (8)$$

and their corresponding sensitivity boundary conditions.

$$\mathcal{B}_b(\mathbf{u}, L(\mathbf{u}')) = \mathbf{g}_b(\mathbf{x}, t; b) \quad \Rightarrow \quad \left\{ \begin{array}{l} \mathcal{B}_{e,b}(\mathbf{u}, L(\mathbf{u}')) \\ \mathcal{B}_{n,b}(\mathbf{u}, L(\mathbf{u}')) \end{array} \right\} = \left\{ \begin{array}{l} \mathbf{g}_{e,b}(\mathbf{x}, t; b) \\ \mathbf{g}_{n,b}(\mathbf{x}, t; b) \end{array} \right\} \quad \text{on } \Gamma = \left\{ \begin{array}{l} \Gamma_e \\ \Gamma_n \end{array} \right\}, \quad (9)$$

The similarity of Eqs.(8) and (9) to Eqs.(1) and (2) motivates the use of the same numerical solution method and the same mesh for the discretized form. The notation $\mathbf{g}_{e,b}$ and $\mathbf{g}_{n,b}$ is used to distinguish prescribed essential or natural boundary conditions, respectively, for the sensitivity boundary conditions and loads associated with a shape design variable b . If the governing Eqs.(1) are linear, then $\mathcal{A}_b = \mathcal{A}$ and $\mathcal{B}_b = \mathcal{B}$. If the governing Eqs. (1) are nonlinear, the solution \mathbf{u} can be obtained from the analysis solution of Eq.(1) for use in Eqs.(8)) and (9). The load or forcing terms for CSA appear at two places: Body loads \mathbf{f}' and sensitivity boundary loads \mathbf{g}_b . Body loads are often zero for a shape variable, unless the body loads \mathbf{f} explicitly or implicitly depend on the shape variable b . Sensitivity boundary loads arise due to the non-homogeneous CSA boundary conditions (9) and are typically the main forcing terms that drive the local derivatives \mathbf{u}' . As shown in Eq.(9), the term \mathbf{g}_b includes spatial gradients $\nabla_{\mathbf{x}}\mathbf{u}$ of the response. Thus, the accuracy of the spatial gradients $\nabla_{\mathbf{x}}\mathbf{u}$ directly affects the accuracy of the local derivatives \mathbf{u}' . Here $\mathbf{u}' = \partial\mathbf{u}/\partial b$ is the local derivative of the response.

Solution of the local CSEs yields the local derivative, $\mathbf{u}' = \partial\mathbf{u}/\partial b$. The total or material derivative $\dot{\mathbf{u}} = D\mathbf{u}/Db$ is then obtained by adding the convective term to the

local derivative.

$$\frac{D\mathbf{u}}{Db} = \frac{\partial\mathbf{u}}{\partial b} + \sum_{i=1}^3 \frac{\partial\mathbf{u}}{\partial x_i} \frac{\partial x_i}{\partial b} \iff \dot{\mathbf{u}} = \mathbf{u}' + \nabla_{\mathbf{x}}\mathbf{u} \cdot \mathcal{V} \quad (10)$$

The convective term consists of the spatial gradients of the response $\nabla_{\mathbf{x}}\mathbf{u} = \partial\mathbf{u}/\partial\mathbf{x}$, and the geometric sensitivity or design velocity $\mathcal{V}(\mathbf{x}) = \partial\mathbf{x}/\partial b$ which depends on the geometric parametrization of the domain. For value design parameters that are independent of the domain shape, the convective term goes to zero, because the design velocity is zero, and so the material derivative is same as the local derivative. However, for shape design variables, the design velocity and hence the convective term are not zero. The next step involves discretization of the domain and CSA numerical solution on the discretized domain.

3.2 Direct Formulation of CSA

A direct formulation of CSA is useful when derivatives of many performance measures is required with respect to a modest number of design variables, which is often the case for shape design variables. Discretization of (8) in a similar fashion as the FEA in Section 2.2, leading to a similar set of discrete CSEs, which for Boundary-Velocity CSA may be written as

$$[\mathbf{K}] \{\mathbf{u}'\}_i = \{\mathbf{F}'\}_i. \quad (11)$$

The stiffness matrix $[\mathbf{K}]$ for the discretized CSEs (11) is same as the stiffness matrix of the primary analysis for linear cases or same as the converged tangent stiffness matrix for nonlinear cases [5, 10, 19]. As stated earlier, although the original system may be nonlinear with respect to the state variables \mathbf{u} , the CSEs are always linear with respect to the sensitivity variables \mathbf{u}' . The vector $\{\mathbf{u}'\}_i = \partial\{\mathbf{u}\}/\partial b_i$ denotes the solution for local derivatives, and b_i is one of the n_b design variables from the set $\{\mathbf{b}\} = [b_1, b_2, \dots, b_{n_b}]^T$.

Following the same scheme as in Section 2.2, Eq.(11) can be partitioned as $\{\mathbf{u}'\} = \{\{\mathbf{u}'\}_e^T, \{\mathbf{u}'\}_f^T\}^T$, where $\{\mathbf{u}'\}_e$ represents the known local derivatives of the constrained degrees of freedom and $\{\mathbf{u}'\}_f$ represents the unknown (free) local derivatives. Subscript i represents derivatives with respect to design variable b_i for $i = 1, 2, \dots, n_b$. The prescribed $\{\mathbf{u}'\}_e$ are obtained from the CSA essential boundary conditions Eq.(9) discretized at given BC locations on Γ_e

$$\mathbf{u}'|_{\Gamma_e} = \dot{\mathbf{g}}_e - \nabla_{\mathbf{x}}\mathbf{u}_e \cdot \mathcal{V}(\mathbf{x})|_{\Gamma_e}. \quad (12)$$

The effect of perturbing a boundary is incorporated through design velocity $\mathcal{V}(\mathbf{x})$ in the last term multiplying the displacement spatial derivative whose accurate computation is the subject of this paper.

The sensitivity load on the right side of Eq.(11) includes a partition for the sensitivity of the natural boundary condition loads, $\{\mathbf{F}'_n\}_i$. The corresponding CSA BC can be stated as

$$\mathcal{B}_b(L(\mathbf{u}'))|_{\Gamma_n} = \dot{\mathbf{g}}_n(\mathbf{x}; b) - \nabla_{\mathbf{x}}(\mathcal{B}_n(L(\mathbf{u}))) \cdot \mathcal{V}(\mathbf{x})|_{\Gamma_n}. \quad (13)$$

Here $\mathcal{B}_n(L(\mathbf{u}))$ represents the secondary variables such as forces and stresses that can be output from the structural solver Cross and Canfield [10] first showed that the loads $\{\mathbf{F}'_n\}_i$ can be calculated based on the first-order spatial gradients of these secondary variables, rather than from the higher-order derivatives of the primary variables such as displacements. The best approach to accurately compute these spatial gradients from the original FEA results is considered in the next section.

4 SPATIAL GRADIENT RECONSTRUCTION (SGR)

The CSA forcing terms consist of the body loads \mathbf{f}' , as given in Eq.(9), and the loads arising from the CSA boundary conditions \mathbf{g}_b . Typically the loads \mathbf{f}' can be derived analytically, since the analysis body loads \mathbf{f} are known. However, \mathbf{g}_b requires calculation of the spatial gradient $\nabla_{\mathbf{x}}\mathbf{u}$ or $\nabla_{\mathbf{x}}(\mathcal{B}_n(L(\mathbf{u})))$ as shown in Eqs.(12) and (13)). After discretization, this leads to the load terms $\{\mathbf{F}'_n\}_i$ and $\{\mathbf{u}'_e\}_i$. Essentially, spatial gradients of analysis primary responses such as displacements, or of secondary responses such as forces or stresses, at the boundary Γ are required to be approximated from the analysis solution to obtain loads for the sensitivity analysis. Finally, after the solution to local derivatives is obtained, spatial gradients of the responses are required to calculate the convective term $\nabla_{\mathbf{x}}\mathbf{u} \cdot \mathcal{V}$ which is added to get the total derivatives according to Eq. (10). In summary, accuracy of the local and total derivatives obtained using CSA is significantly affected by the accuracy of the boundary spatial gradients [7]. Elsewhere we have shown that reliance upon low-order element shape functions limits the accuracy of these spatial gradients thus limiting the accuracy of the local and effectively total derivatives [9]. As a remedy, Cross and Canfield [10, 11, 12] proposed the spatial gradient reconstruction (SGR), which is based on the least-squares patch-recovery approach used by [14] and Duvigneau and Pelletier [7]. The difference between the approaches is that SGR is used to reconstruct first-order gradients of the natural boundary conditions directly, instead of reconstructing the higher-order gradients that appear in them. As an example, for a second-order SGR, a least-squares solution for the derivatives of the following second-order Taylor series expansion is used for two-dimensional Cartesian coordinates.

$$\begin{aligned} \phi(x + \Delta x, y + \Delta y) &= \phi(x, y, z) + \phi_{,x}\Delta x + \phi_{,y}\Delta y \\ &+ \frac{1}{2}\phi_{,xx}(\Delta x)^2 + \frac{1}{2}\phi_{,yy}(\Delta y)^2 \\ &+ \phi_{,xy}\Delta x\Delta y \\ &= a_0 + a_1\Delta x + a_2\Delta y + a_3\Delta x\Delta y + a_4(\Delta x)^2 + a_5(\Delta y)^2 \end{aligned} \quad (14)$$

Estimated field variable ϕ could be a scalar such as a component of displacement or stress and $\phi_{,x}$, $\phi_{,y}$, $\phi_{,z}$ are the spatial gradients that are computed using linear regression to match the finite element nodal displacements or element stresses in a least-squares sense. For stress approximation, $\phi(x, y) \equiv \sigma_P(x, y)$ where σ_P is one of the stress components, σ_x , σ_y , or τ_{xy} recovered over a patch, P , Eq.(14) may be written in matrix-vector form as

$$\sigma_P \cong [\mathbf{P}]\{\mathbf{a}\} \quad (15)$$

Motivated by the oft-cited observation that “the quality of the recovered derivatives is the inferior at boundary points compared to interior points,” [15] Wiberg *et al.* include in the weighted residual formulation for the regression of the coefficients for (14) the residual error for stress components and equilibrium equations evaluated at element Gauss points, as well as the residual error for applied tractions evaluated at nodes on the element exterior boundary. A complication of this enhanced residual is that it couples the solution for SGR coefficients of the stress components. Whereas Wiberg *et al.* present accuracy of patch recovery for the stress field, in CSA we are interested in the recovery of the spatial gradient of the stress field, which appear in the CSA boundary conditions that drive the solution of the sensitivity equations. Consequently, we present in Section 6 results that

demonstrate accuracy of not only stresses, but more importantly the recovery of stress spatial derivatives.

The SGR coefficients \mathbf{a} in Eq.(15) may be determined by weighted least-squares minimization of residual error between the patch stress σ_P and the element stress σ^* evaluated at n_G Gauss points

$$R_\sigma = \sum_{G=1}^{n_G} w_G [\sigma_P(x_G, y_G) - \sigma^*(x_G, y_G)] \quad (16)$$

The residual weighting factors w_G in Eq.(16) are taken as inversely proportional to the distance from the Gauss point to the Taylor series expansion point. Minimizing the residual (16) alone allows uncoupled recovery of each stress component independently.

Minimizing (16) together with residual error between the patch stress vector components and n_T tractions—normal Φ_n and tangential (shear) Φ_t stress components—couples the stress components recovery

$$R_n = \sum_{m=1}^{n_T} w_m [(\sigma_n(x_m, y_m) - \Phi_n(x_m, y_m)) + (\tau_t(x_m, y_m) - \Phi_t(x_m, y_m))] \quad (17)$$

due to the surface normal vector \vec{n} on the boundary Γ_n for natural boundary conditions \mathbf{g}_n in Eq.(4)

$$\{\Phi\} = \begin{Bmatrix} \Phi_n \\ \Phi_t \end{Bmatrix} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{xy} & \sigma_{yy} \end{bmatrix} \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} = [\sigma]_P \vec{n} = \begin{Bmatrix} \sigma_n \\ \tau_t \end{Bmatrix} \quad (18)$$

for normal stress σ_n and shear stress τ_t , where the traction residual weighting factors w_m in Eq.(17) are taken as an order of magnitude large than the largest w_G weighting factor.

Finally, satisfaction of the equilibrium equations, $[\partial]\{\sigma\} + f = 0$ for body forces assumed zero, $f = 0$ may be considered.

$$\begin{bmatrix} \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad (19)$$

Whereas Wilberg *et al.* [15] incorporate Eq.(19) as another contribution to residual error in their least-squares solution for their SPRE (superconvergent patch recovery with equilibrium) method for coefficients \mathbf{a} , Kvamsdal [20] and Okstad *et al.* [21] avoided the residual error of the equilibrium Eqs.(19) by using them as constraint equations to eliminate some of the regression coefficients from Eq.(14), so as to create statically admissible stress fields that automatically satisfied internal equilibrium (at Gauss points). That approach was also considered here, which tended to average the residual of the spatial derivative coefficient that was eliminated with its counterpart that was retained.

For the following coupled second-order stress SGR approximation

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{sy} \end{Bmatrix} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \end{bmatrix} \begin{Bmatrix} \Delta x \\ \Delta y \\ \Delta x \Delta y \\ (\Delta x)^2 \\ (\Delta y)^2 \end{Bmatrix} \quad (20)$$

enforcement of Eq.(20) for all x and y in the patch implies that the following coefficients may be eliminated: $a_4 = -c_3/2$, $c_5 = -a_3/2$, $c_2 = -a_1$, $c_4 = -b_3/2$, $b_5 = -c_3/2$, and $c_1 = -b_2$.

The advantages of SGR include:

- Spatial gradients $\nabla_{\mathbf{x}}\mathbf{u}$ can be approximated without knowledge of the finite element formulation or its element shape functions.
- Accuracy of the spatial gradients can be increased by choosing the correct order of Taylor series for SGR.
- SGR is a post processing step, following the structural analysis, using any black-box tool, which makes CSA amenable to a nonintrusive implementation.

SGR presented here to approximate the spatial gradients will be used for required CSA sensitivity loads and material derivative recovery.

SGR results reported in [22] were based on SGR of nodal displacements, whereas new results presented here were developed with stresses output at element centers. Improvements in accuracy were achieved by incorporating the traction boundary conditions as additional equations for the least-squares fit to determine the coefficients in Eq.(14), following [15].

5 TIMOSHENKO BEAM ANALYTIC 2D EXAMPLE

Timoshenko and Goodier [23] derived the analytic elasticity solution for bending of a two-dimensional rectangular cantilever beam loaded with a parabolically distributed transverse shear force at its free end. That elasticity solution may be differentiated with respect to its length L and height h as shape parameters to serve as a reference sensitivity solution to evaluate the accuracy of computational sensitivity analysis. Augarde and Deeks [24] elucidated the proper boundary conditions to impose when the analytic elasticity solution is used as a benchmark to evaluate convergence of adaptive finite element models. Here we adopt their reference frame in which the beam is cantilever at $x = 0$ on the left with the parabolic load of magnitude P applied downward at $x = L$ on the right, as shown in Fig. 1. The displacement solution in [24], however, does not correspond to the

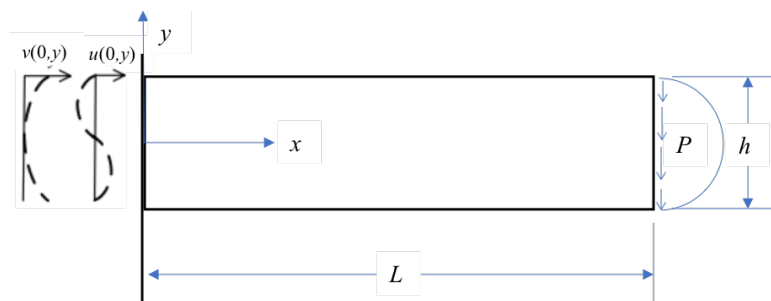


Figure 1: Timoshenko two-dimensional cantilever beam coordinate system, dimensions, and shear load

one in [23] that allows shear deformation. The Airy stress function derivation, found in [25] in terms of a coordinate system origin at the cantilever root on the left, incorporating

the shear factor in [24], is used here as follows.

$$\begin{aligned} u(x, y) &= \frac{P}{EI} \left[\left(L - \frac{x}{2} \right) xy + \left(\kappa - (1 + \nu) \frac{h^2}{4} \right) y + \frac{(2 + \nu)}{6} y^3 \right] \\ v(x, y) &= \frac{-P}{EI} \left[\kappa x + \frac{Lx^2}{2} - \frac{x^3}{6} + \frac{\nu}{2} (L - x)y^2 \right] \end{aligned} \quad (21)$$

where the beam has second area moment of inertia $I = \frac{bh^3}{12}$ for a rectangular cross-section of unit depth $b = 1$ for an isotropic material with elastic modulus E , shear modulus, $G = \frac{E}{2(1+\nu)}$, Poisson ratio ν , and shear force effect factor $\kappa = \frac{EI}{kAG} = (1 + \nu)h^2/4$ for shear area reduction factor k , subject to parabolic transverse shear with total downward force P . Timoshenko's original solution corresponds to a shear angle of $P/(kAG)$ at the root with respect to the horizontal, where he used the shear area reduction factor $k = 2/3$, thereby canceling the coefficient of the linear term in y of the middle term on the right side $u(x, y)$ in Eq.(21). For the numerical results to follow, the parameter values specified in [24] were used: $h=2$, $L=8$, $P=2$, $E=1,000$, $\kappa=1.25$, and $\nu=0.25$. For these values tip deflection $v(L, 0) = \frac{-PL^3}{3EI}(1 + 3\kappa/L^2) = -0.542$, axial normal stress and strain at the top corner of the root are $\sigma_{xx}(0, 1) = 24$ and $\epsilon_{xx}(0, 1) = 0.024$, transverse normal stress and strain are $\sigma_{yy}(0, 1) = 0$ and $\epsilon_{yy}(0, 1) = -0.006$, and shear stress and strain at the root neutral axis are $\tau_{xy}(0, 0) = -1.5$ and $\gamma_{xy} = -0.00375$, respectively. In general, analytical stresses are

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{P(L-x)y}{I} \\ 0 \\ \frac{P(y^2-c^2)}{2I} \end{Bmatrix} \quad (22)$$

Essential boundary conditions (3) at the root nodes were imposed as enforced displacements in the finite element model for the analysis in accordance with Eqs.(21) at $x = 0$.

$$\mathbf{g}_e = \begin{Bmatrix} u(0, y) \\ v(0, y) \end{Bmatrix} = \frac{P}{EI} \begin{Bmatrix} \left[\kappa - (1 + \nu) \frac{h^2}{4} \right] y + \frac{(2+\nu)}{6} y^3 \\ -\frac{\nu}{2} Ly^2 \end{Bmatrix} = \begin{Bmatrix} 0.001125y^3 \\ -0.003y^2 \end{Bmatrix} \quad (23)$$

The natural boundary condition is a prescribed traction on the free right end of the beam

$$\mathbf{g}_n = [\boldsymbol{\sigma}] \cdot \mathbf{n} = \Phi(L, y) \quad (24)$$

where $[\boldsymbol{\sigma}]$ is the matrix of stress components and \mathbf{n} is the surface normal vector. Parabolic shear stress traction Φ_y at $x = L$ was applied at the tip

$$\Phi_y(L, y) = \frac{3}{2} \left(\frac{-P}{bh} \right) \left[1 - \left(\frac{y}{c} \right)^2 \right] = 1.5(y^2 - 1) \quad (25)$$

where $c = \frac{h}{2}$. Consistent nodal loads were applied at each tip element e node by integrating Eq.(25) along the free edge on the right end of the beam

$$\{\mathbf{F}_n^e\} = \int_0^1 \Phi_y(L, y(\eta)) \begin{Bmatrix} 1 - \eta \\ \eta \end{Bmatrix} b l_e d\eta \quad (26)$$

where $\ell_e = y_{e2} - y_{e1}$ is the length of the free edge of element e from y -coordinate of the lower node $e1$ to the upper node $e2$. The natural coordinate $\eta \in \{0, 1\}$ maps the vertical coordinate $y = y_{e1} + \eta\ell_e$ of the free edge of the tip elements.

To illustrate the concept of patches used in SGR, two SGR patches are shown on the left in Fig. 2 as colored elements for the coarsest mesh. The first layer of elements are colored green, the second layer colored blue, and the third layer colored red. In the mesh above a 2.5-layer patch at the root neutral axis is shown. The mesh below depicts a 2.5-layer patch at the root bottom corner.

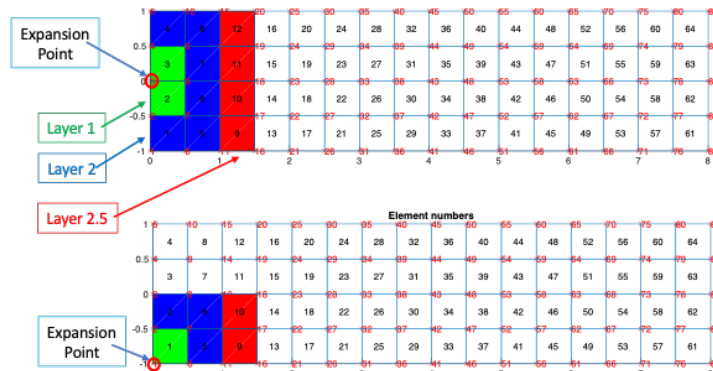


Figure 2: FEM QUAD4 Mesh Convergence for Tip Deflection Error

Differentiation of Eq.(22) with respect to coordinates x and y provides the exact solution for spatial derivatives that will be used to evaluate the accuracy of SGR, where numerical values at the tip are given here.

$$\nabla_x \{\sigma\} = \begin{Bmatrix} \frac{\partial \sigma_{xx}(x,y)}{\partial x} \\ \frac{\partial \sigma_{yy}(x,y)}{\partial x} \\ \frac{\partial \tau_{xy}(x,y)}{\partial x} \end{Bmatrix} = \begin{Bmatrix} Py/I \\ 0 \\ 0 \end{Bmatrix} \quad (27)$$

$$\nabla_y \{\sigma\} = \begin{Bmatrix} \frac{\partial \sigma_{xx}(x,y)}{\partial y} \\ \frac{\partial \sigma_{yy}(x,y)}{\partial y} \\ \frac{\partial \tau_{xy}(x,y)}{\partial y} \end{Bmatrix} = \begin{Bmatrix} P(L-x)/I \\ 0 \\ Py/I \end{Bmatrix} \quad (28)$$

Most of these derivatives are zero at the root and tip corners and neutral axis where SGR results were generated for convergence studies for which absolute true errors were plotted. Relative true errors were plotted for non-zero values: $\frac{\partial \sigma_{xx}}{\partial x}(x, c) = -3$, $\frac{\partial \sigma_{xx}}{\partial y}(0, y) = 24$, $\frac{\partial \tau_{xy}}{\partial y}(L, c) = 3$, and $\frac{\partial \tau_{xy}}{\partial x}(L, -c) = -3$.

Differentiation of $v(x, y)$ in Eq.(21) evaluated at $x = L$ with respect to beam length L and height h give the analytical material derivatives of the tip deflection $\dot{v}(L, 0)$

$$\frac{Dv(L, 0)}{DL} = \frac{-P}{EI} [\kappa + L^2] = -0.19575 \quad (29)$$

$$\frac{Dv(L, 0)}{Dh} = \frac{P}{EI} \frac{L}{h} [\kappa + L^2] = 0.783 \quad (30)$$

Likewise, analytic material shape derivatives of stress at the root and tip are

$$\begin{aligned} \frac{D\sigma_{xx}(0, c)}{DL} &= \frac{Pc}{I} = 3 & \frac{D\tau_{xy}(0, 0)}{DL} &= 0 \\ \frac{D\sigma_{xx}(0, c)}{Dh} &= \frac{-PL}{I} = -24 & \frac{D\tau_{xy}(0, 0)}{Dh} &= \frac{3P}{2bh^2} = 0.75 \end{aligned} \quad (31)$$

$$\begin{aligned} \frac{D\sigma_{xx}(L, c)}{DL} &= \frac{Pc}{I} = 3 & \frac{D\tau_{xy}(L, 0)}{DL} &= 0 \\ \frac{D\sigma_{xx}(L, c)}{Dh} &= \frac{-PL}{I} = -24 & \frac{D\tau_{xy}(L, 0)}{Dh} &= \frac{3P}{2bh^2} = 0.75 \end{aligned} \quad (32)$$

The sensitivity $\dot{\mathbf{g}}_e$ of the geometric boundary condition (23) is non-zero with respect to length L for the vertical root deflection $v(0, y)$

$$\dot{\mathbf{g}}_{e,L} = \frac{D\mathbf{g}_e}{DL} = \left\{ \begin{array}{c} \frac{Du(0,y)}{DL} \\ \frac{Dv(0,y)}{DL} \end{array} \right\} = \left\{ \begin{array}{c} \dot{u}(0, y) \\ \dot{v}(0, y) \end{array} \right\} = \frac{P}{EI} \left\{ \begin{array}{c} 0 \\ \frac{-\nu}{2}y^2 \end{array} \right\} \quad (33)$$

By virtue of the chain rule of differentiation for the moment of inertia I and the choice $\kappa = (1+\nu)c^2$, the total derivative of root deflections with respect to height h is proportional to \mathbf{g}_e , the enforced boundary condition (23) of the analysis

$$\dot{\mathbf{g}}_{e,h} = \frac{D\mathbf{g}_e}{Dh} = \frac{P}{EI} \left\{ \begin{array}{c} 0 \\ \frac{\nu L}{2h}y^2 \end{array} \right\} \quad (34)$$

The material derivative boundary conditions (33) and (34) are defined by user input and used to impose different enforced displacements for each shape design variable and its associated design velocity, according to Eq.(12), in the CSA for the local shape derivatives. Design velocity is not unique. For lengthening the beam, the left end could be moved left, giving design velocity $\mathcal{V}_1(x, y) = \frac{x}{L} - 1$. In general, the design velocity may depend on both x and y , but in this case, it depends only on x . Then the local sensitivity boundary condition (9) at the fixed end on the left for beam length and its design velocity becomes

$$\mathbf{g}_{e,L1} = \left\{ \begin{array}{c} u'(0, y) \\ v'(0, y) \end{array} \right\} = \left\{ \begin{array}{c} \dot{u}(0, y) - \nabla u(0, y) \cdot \mathcal{V}_1(0, y) \\ \dot{v}(0, y) - \nabla v(0, y) \cdot \mathcal{V}_1(0, y) \end{array} \right\} = \frac{P}{EI} \left\{ \begin{array}{c} Ly \\ -(1+\nu)c^2 \end{array} \right\} = \left\{ \begin{array}{c} 0.024y \\ -0.00375 \end{array} \right\} \quad (35)$$

The upper component in the vector of Eq.(35) is the convective term that in practice is approximated by SGR. The lower component in the vector is user input from the boundary condition defined by Eq.(33). The sensitivity load is zero at the tip, because the prescribed traction Eq. (25) does not depend on the beam length and the design velocity is zero at the tip.

Alternatively, the right end could be moved to the right, giving design velocity $\mathcal{V}_2(x, y) = \frac{x}{L}$. Then, instead of (35), the local sensitivity essential boundary condition on the fixed end on the left becomes simply the material derivative $\dot{\mathbf{g}}_e$ given by user input of Eq.(33). A local traction sensitivity load would be defined at the free end on the right as

$$\mathcal{B}_{n,L2}(\mathbf{u}') = \nabla[\boldsymbol{\sigma}'] \cdot \mathbf{n} = \mathbf{g}_{n,L2} = \boldsymbol{\Phi}'_{,L2}(L, y) = \dot{\boldsymbol{\Phi}}_{,L}(L, y) - [\boldsymbol{\sigma}] \dot{\mathbf{n}} - (\nabla[\boldsymbol{\sigma}] \cdot \mathbf{n}) \cdot \mathcal{V}_2(L, y), \quad (36)$$

which is non-zero due to the convective term, despite that $\dot{\boldsymbol{\Phi}}_{,L}(L, y) = 0$. Here $\dot{\mathbf{n}} = 0$ because the normal direction of the rectangular boundary surface does not change for

length or height shape design variable. The sole convective term in Eq.(36), $(\nabla[\sigma] \cdot \mathbf{n}) \cdot \mathcal{V}_2(L)$, is determined by SGR. From analytic differentiation of Eq.(22), the true value for traction local derivative with respect the beam lengthening toward the right for this benchmark is

$$\mathbf{g}_{n,L2} = \mathbf{\Phi}'_{,L2} = \begin{Bmatrix} \Phi'_{x,L2} \\ \Phi'_{y,L2} \end{Bmatrix} = -(\nabla[\sigma] \cdot \mathbf{n}) \cdot \mathcal{V}_2(L, y) = \begin{Bmatrix} \frac{Py}{I} \\ 0 \end{Bmatrix} = \begin{Bmatrix} 3y \\ 0 \end{Bmatrix} \quad (37)$$

Whereas the traction for the analysis problem has only shear force given by Eq.(25), the sensitivity traction given by Eq.(37) involves only normal axial force.

The design velocity for increasing the height could be expressed as $\mathcal{V}_3(x, y) = \frac{y}{c}$. Multiplying it by the gradient of Eq.(21), and substituting this convective product together with Eq.(34) into Eq.(12) leads to the exact expression for essential local sensitivity boundary condition (9) with respect to height change, imposed as an enforced displacement for the CSA

$$\mathbf{g}_{e,h} = \begin{Bmatrix} u'(0, y) \\ v'(0, y) \end{Bmatrix} = \frac{P}{EI} \begin{Bmatrix} \frac{3(2+\nu)}{2h} y^3 \\ -\frac{7L\nu}{2h} y^2 \end{Bmatrix} = \begin{Bmatrix} 0.0050625y^3 \\ -0.0105y^2 \end{Bmatrix} \quad (38)$$

Both components of Eq.(38) have convective contributions that in practice are computed from SGR, whereas the contribution from Eq.(34) is user-defined input to the computational CSA.

The exact expression for the natural local sensitivity boundary condition (9) at the free end on the right for the height shape variable and its design velocity is

$$\mathbf{g}_{n,h} = \dot{\mathbf{g}}_{n,h} - [\sigma] \dot{\mathbf{n}} - \nabla[\sigma] \cdot \mathcal{V}_3(y) \quad (39)$$

Again, Eq.(39) has convective contributions that, in practice, are approximated by SGR, while user input defines the design velocity.

6 RESULTS

Each of the following figures plot the error of reconstructed stress components or their spatial derivative for four increasingly refined meshes of 4×16 , 8×32 , 16×64 , and 32×128 quadrilateral elements, respectively, labeled as Mesh #1, 2, 3, and 4, respectively, on the x -axis of the plots. The full set of plots appear in the appendices, including additional plots for SGR for which the exact solution for displacements from Eq.(21) were used in place of the FEM displacements post-processed from ASTROS or NASTRAN. Each figure plots convergence rates for three SGR methods: (a) displacement-based, (b) stress-based without traction enforcement, and (c) stress-based with traction enforcement. Curves in the plots correspond to choices of order for the SGR Taylor series expansion and the number of element layers in the patch, as indicated in the plot's legend. The rate of convergence for each SGR patch was determined from the slope of its curve and indicated by the value m in the legend, if linear enough. For some non-monotonic curves with good convergence at the finest mesh, m^* indicates the slope of the line segment between the last two meshes. Collectively, the figures demonstrate that stress-based, traction-enforced SGR is the most accurate, although the choice of Taylor series order for the best SGR patch varies with the quantity being reconstructed.

Displacement-based SGR requires at least a second-order patch for stress derivatives, since stresses are second derivatives of displacement. In contrast, stress-based SGR may

use first-order patches to estimate stress first derivatives, but requires more layers at beam surfaces than the SGR order to have enough equations to solve for the SGR Taylor series coefficients. However, when traction equations are added, only as many layers as the SGR order are needed.

6.1 Quadrilateral Meshes

Figure 3 clearly shows that stress-based, traction-enforced SGR is more accurate for x-normal stress, $\sigma_{xx} = 0$, at the top corner of the beam tip for all choices of patches. As expected, increasing order improves the accuracy, except for stress-based without tractions. Displacement-based SGR exceeds second-order rate of convergence for second-order patches. Although more accurate, third-order patches do not exceed first-order rate of convergence. For stress-based SGR this may be due to round-off for the small values of σ_{xx} at the corner, since ASTROS computes stress only in single precision. Whereas stress-based SGR reconstructs tip corner normal stress somewhat more accurately than displacement-based SGR, traction-enforced stress-based SGR greatly improves accuracy.

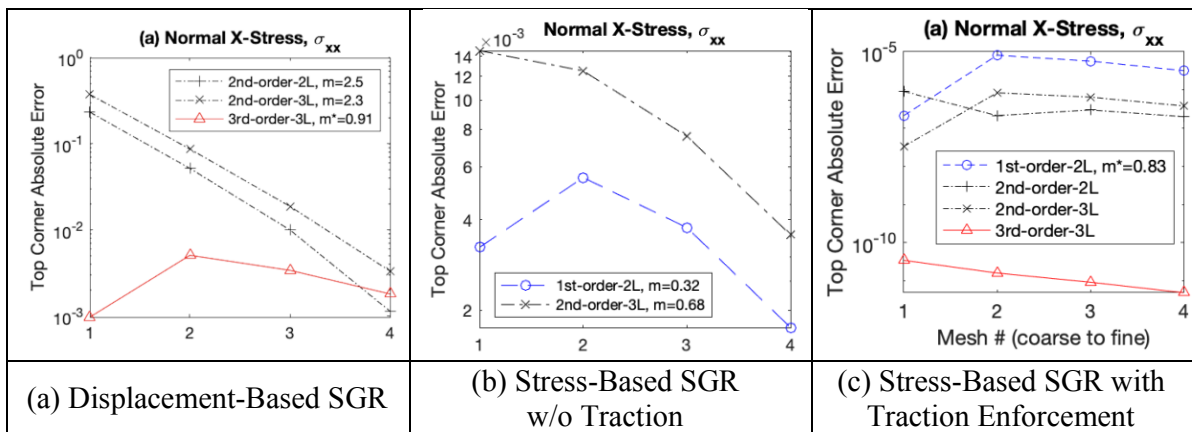


Figure 3: SGR Convergence of Normal X-stress at Top Corner of Beam Tip

Figure 4 again shows that stress-based, traction-enforced SGR is the most accurate for shear stress, $\tau_{xy} = 0$, at the top corner of the beam tip, although its first-order patch is not the most accurate for the coarsest mesh. Surprisingly, first-order stress-based SGR exhibits a second-order rate of convergence. Second and third-order SGR with traction enforcement achieve machine-precision accuracy.

Figure 5 also shows that stress-based, traction-enforced SGR is the most accurate method for shear stress, $\tau_{xy} = -1.5$, except for first-order, at the neutral axis of the beam tip. The patches are super-convergent with second and third-order SGR achieving numerically zero relative error for single precision. Interestingly, the two-layer patches for second-order displacement SGR in Fig 5 (a) behave much like stress-based SGR without traction in Fig 5 (b), and likewise for the three-layer third-order displacement-based compared to second-order stress-based SGR.

Figure 6 shows that stress-based SGR is better than displacement-based SGR for the σ_{xx} derivative with respect to x , although accuracy without traction enforcement curiously deteriorates with mesh refinement (*i.e.*, negative slope for rate of convergence). The likely reason is that, whereas the free surface on the top patch provides additional information for normal stress components $\sigma_{yy} = \Phi_y = 0$ and tangential shear stress com-

Robert A. Canfield

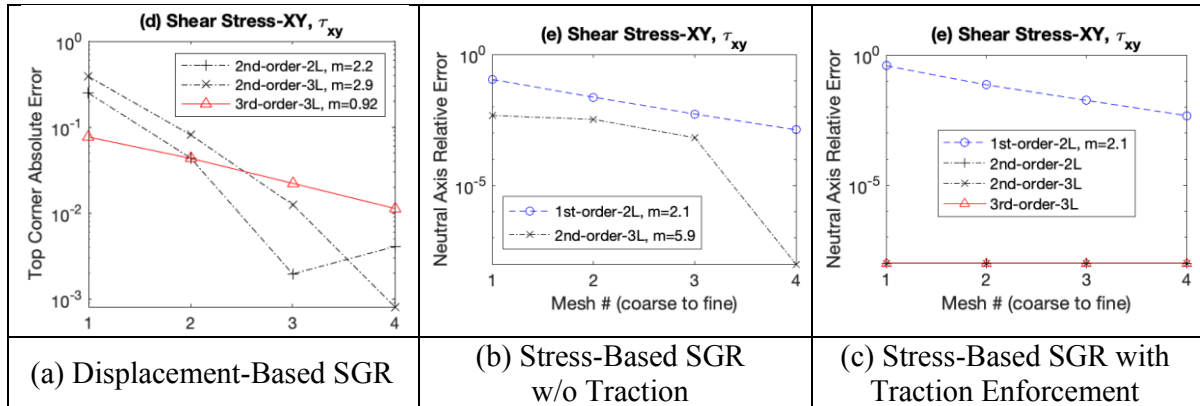


Figure 4: SGR Convergence of Shear XY-stress at Top Corner of Beam Tip

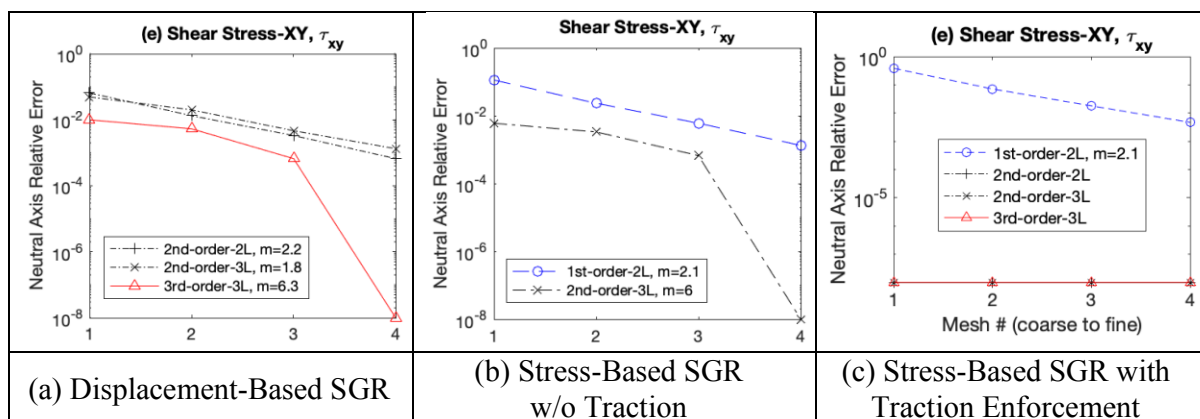


Figure 5: SGR Convergence of Shear XY-stress at Neutral Axis of Beam Tip

Robert A. Canfield

ponent $\tau_{xy} = \Phi_x = 0$, it provides no additional information about σ_{xx} for the $\frac{\partial \sigma_{xx}}{\partial x} = -3$ derivative. Unusually, third-order traction-enforced SGR is worse even than first-order. Traction-enforced second-order SGR does remarkably well, achieving machine precision accuracy! It is explained by the coefficient reduction $c_2 = -a_1$ from Eq.(20) due to the equilibrium constraints in Eq.(19) implies that $\frac{\partial \sigma_{xx}}{\partial x} = -\frac{\partial \tau_{xy}}{\partial y}$ wherein the latter is reconstructed accurately from the exactly known parabolic shear traction, which will be seen in Fig.10.

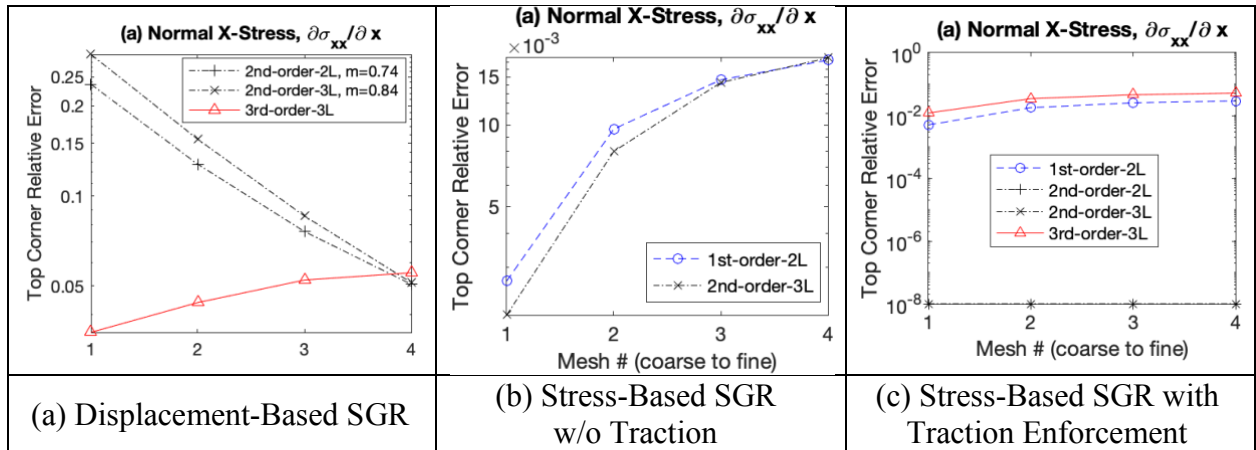


Figure 6: SGR X-Derivative of Normal X-stress at Beam Tip Top Corner

Figure 7 again shows that stress-based, traction-enforced SGR is the most accurate for shear stress derivative, $\frac{\partial \tau_{xy}}{\partial x} = 0$, at the top corner of the beam tip. First-order SGR has a first-order rate of convergence, compensating for the deterioration with mesh refinement without traction enforcement. Higher-order traction-enforced SGR accuracy is constant with mesh refinement that does not improve traction accuracy, since it is known analytically as a natural boundary condition.

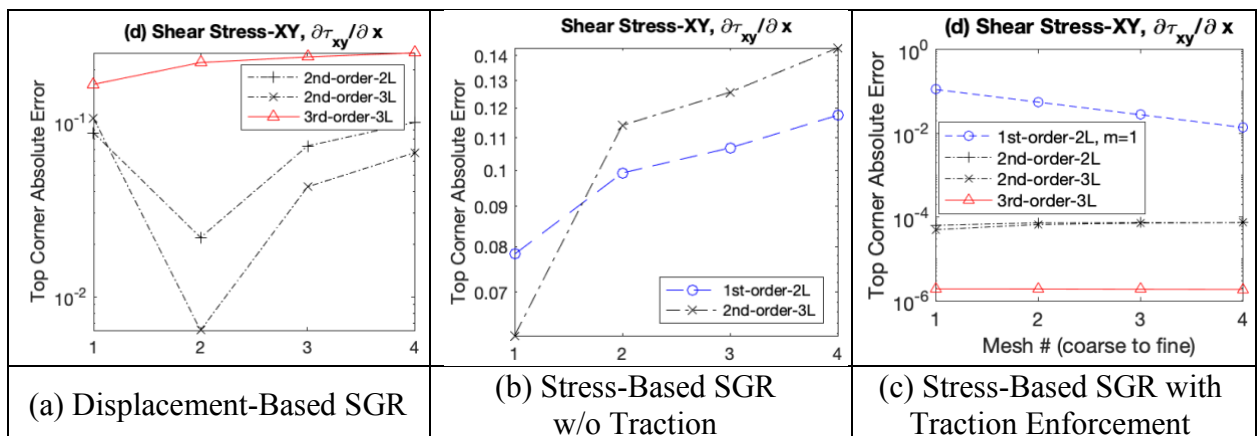


Figure 7: SGR X-Derivative of Shear XY-stress at Beam Tip Top Corner

Figure 8 for shear-stress derivative $\frac{\partial \tau_{xy}}{\partial x} = 0$ at the neutral axis behaves much like Figure 5 for shear stress itself. Once again stress-based SGR without traction-enforcement performs much like displacement-based SGR patches with the same number of layers, but at one order less. The second-order patches for stress-based traction-enforced SGR are the

Robert A. Canfield

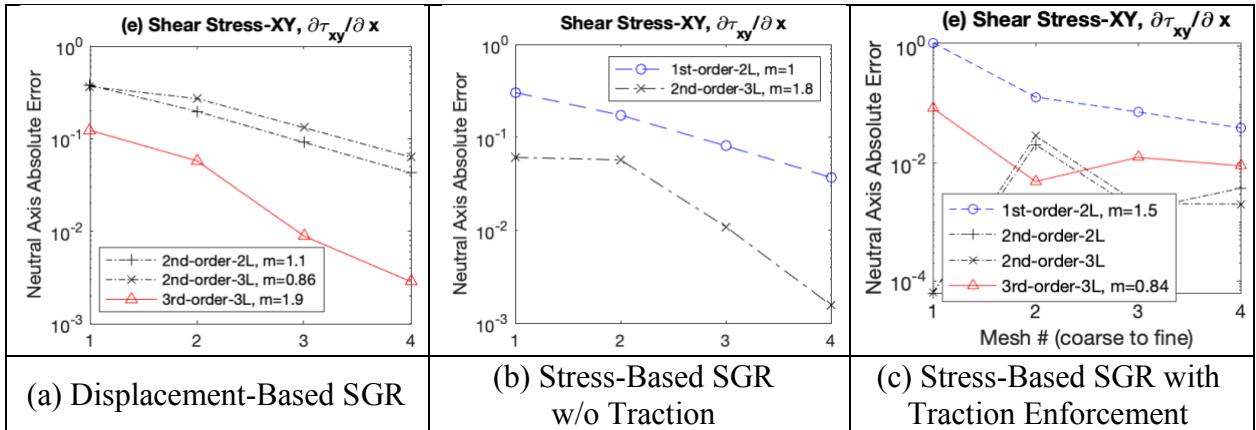


Figure 8: SGR X-Derivative of Shear XY-stress at Beam Tip Neutral Axis

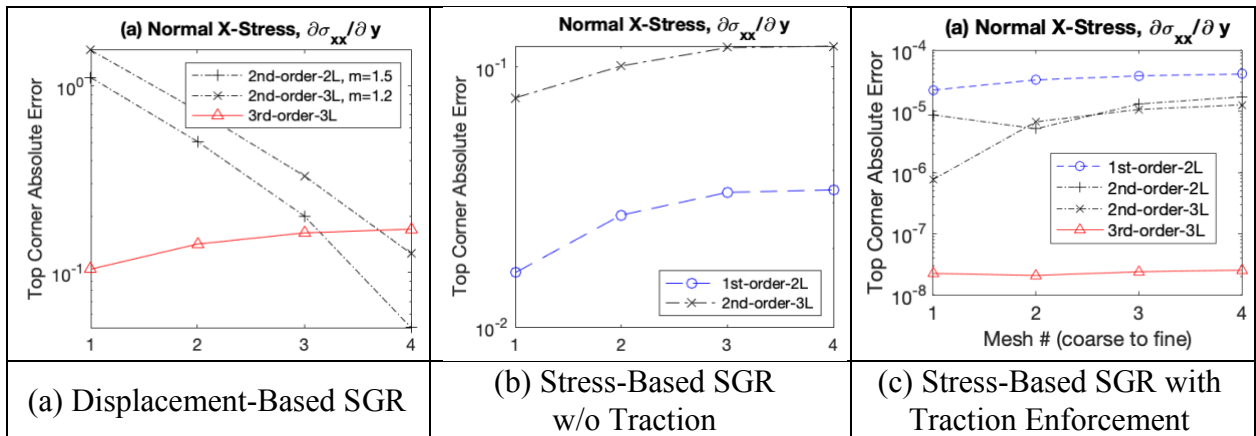


Figure 9: SGR Y-Derivative of Normal X-stress at Beam Tip Top Corner

best choice for the shear stress y -derivative at the neutral axis, as they were at the corner. Third-order traction-enforced SGR is expected to improve once equilibrium constraints are applied to reduce the number or third-order coefficients as they were for the second-order coefficients in Eq.(20).

Likewise, for normal stress σ_{xx} derivative with respect to y , shown in Figure 9, stress-based traction-enforced SGR is still the best choice for any order with third-order SGR the clear choice, achieving machine precision accuracy for this $\frac{\partial \sigma_{xx}}{\partial dy} = 0$ derivative. Rate of convergence for none of the patches for any SGR method exceeds its order.

Finally, relative error plotted Figure 10 for $\frac{\partial \tau_{xy}}{\partial y} = 3$ indicates that stress-based SGR is superior to displacement-based SGR. Traction enforcement is remarkably superior for second-order and third-order stress-based SGR—all results achieving machine precision level accuracy. Otherwise, patches converge at no higher a rate than their order indicates for shear stress derivatives.

The complete set of mesh convergence plots are presented in Appendix A for the ASTROS QUAD4 meshes.

Robert A. Canfield

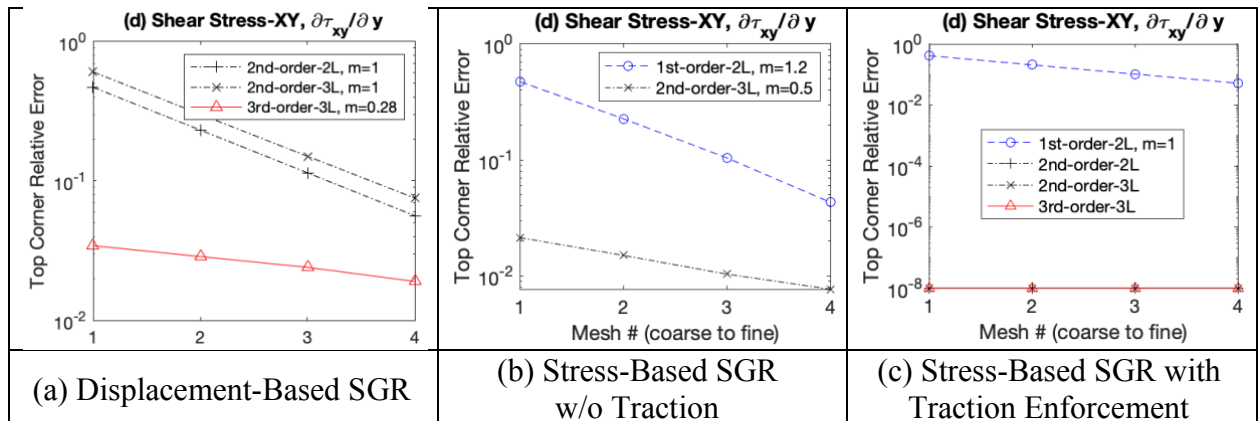


Figure 10: SGR Y-Derivative of Shear XY-stress at Beam Tip Top Corner

6.2 Triangular Mesh

Analogous plots appear in Appendix B for NASTRAN TRIA3 meshes (Fig.11). Although TRIA3 is less accurate an element than QUAD4 for this 2-D Timoshenko beam, the triangular meshes are an important demonstration of the effectiveness of SGR for unstructured meshes. Whereas higher-order compact finite difference schemes could be used to improve accuracy of spatial derivatives on structured meshes, they are not applicable to unstructured triangular element meshes.

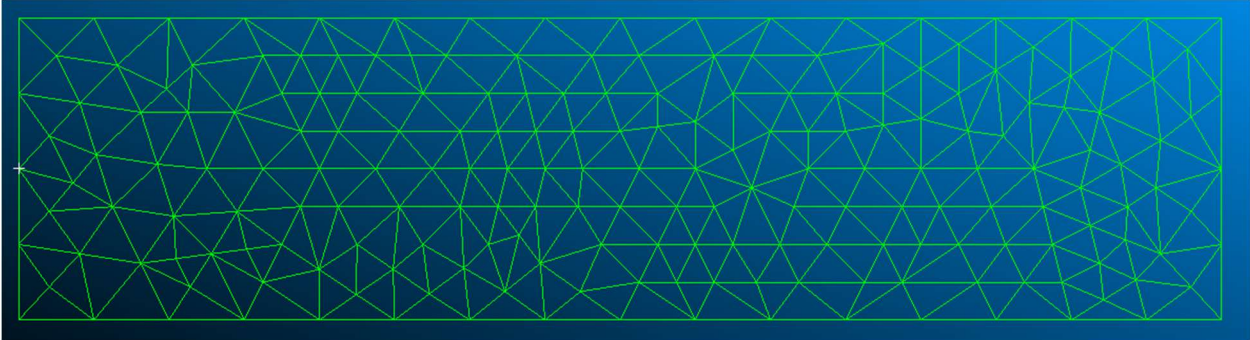
7 SUMMARY AND CONCLUSION

The CSA method for computing analytic shape derivatives of displacement and stress in structures was summarized, highlighting its dependence on spatial derivatives of the FEA solution. Its advantages are that it is element agnostic and can be implemented non-intrusively by post-processing FEA results without requiring access to the source code, much like finite difference methods. Clearly, its accuracy (relative to finite difference) depends on accurate spatial derivatives that are used for form CSA sensitivity loads and recover material shape derivatives from local shape derivatives.

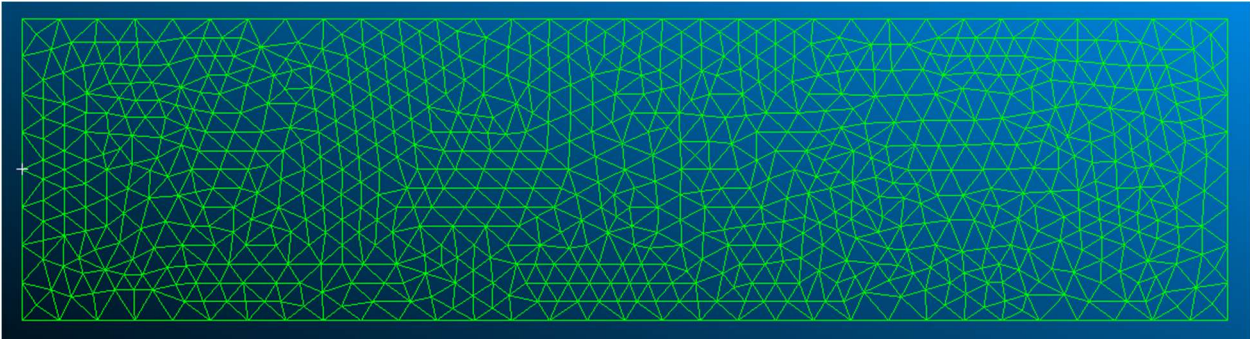
The analytic elasticity solution was presented for a classic benchmark problem, the two-dimensional Timoshenko cantilever beam, and its CSA sensitivity equations and sensitivity boundary conditions were derived by differentiating the Timoshenko beam governing equations and boundary conditions. The Timoshenko beam CSA derivation and analytic solution for its spatial and shape derivatives constitute a worthwhile tutorial for the CSA method in and of themselves. Values for the exact solution for stress and displacement at the root and tip of the beam were used then to evaluate the true error of SGR computed from FEA results.

The SGR convergence studies for in-plane bending of the two-dimensional Timoshenko cantilever showed that stress-based SGR is superior to displacement-based SGR generally for stresses and their spatial derivatives. Traction enforcement greatly improved accuracy of stress-based SGR. Whereas third-order SGR was clearly most accurate for reconstructing stress components themselves, sometimes second-order for shear stress or second-order SGR for normal stress was best for spatial derivatives when tractions were enforced. The results presented are promising for computing more accurate shape derivatives using boundary-velocity CSA. The next results to generate will be shape sensitivity

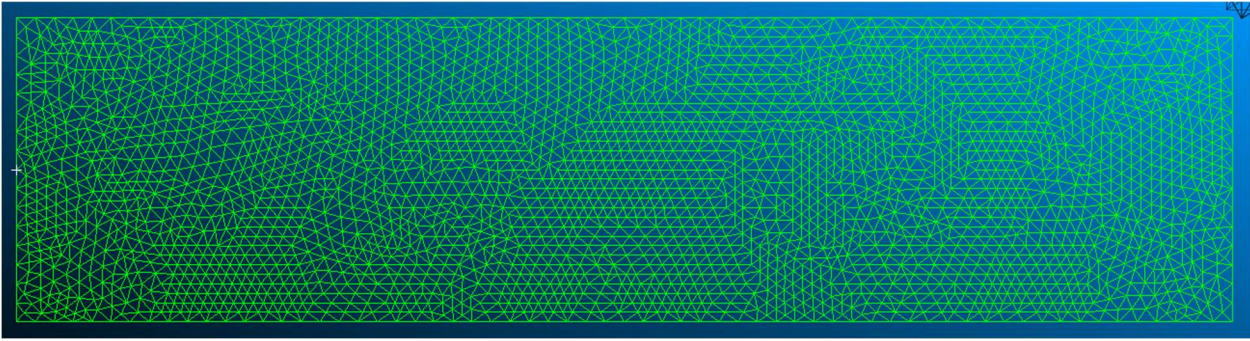
Robert A. Canfield



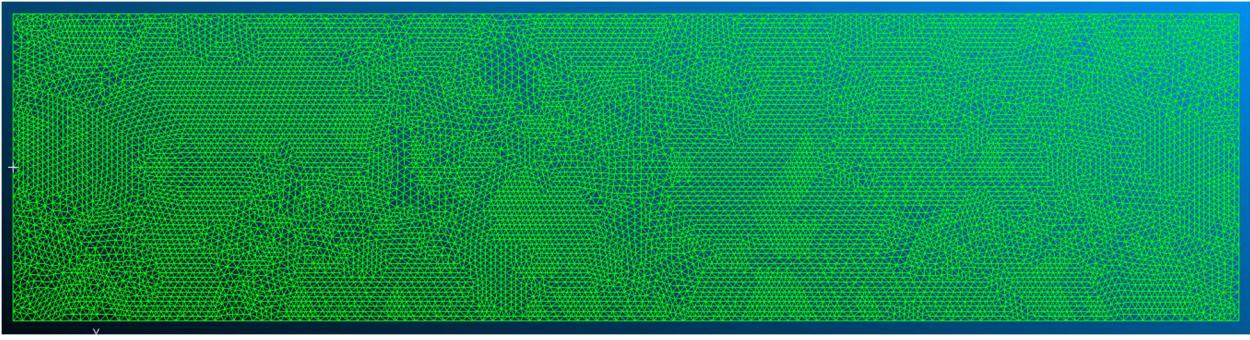
4x16 Mesh



8x32 Mesh



16x64 Mesh



32x128 Mesh

Figure 11: SGR Y-Derivative of Shear XY-stress at Beam Tip Top Corner

convergence using boundary-velocity CSA based on traction-enforced SGR. TRIA3 and QUAD4 meshes will be studied for derivatives of critical displacements and stresses with respect to shape variables for changing the beam length and height using different design velocities.

ACKNOWLEDGEMENTS

Virginia Tech undergraduate students Suood Alnaqbi, Amanda Butynes, Griffin Burd, and Emily Sellards, graduate student Aditya Narkhede, and graduate research assistant Abigail Coffing are gratefully acknowledged for their generation of ASTROS and NAS-TRAN FEM meshes and SGR results plotted here.

Appendices

A ASTROS QUAD4 MESHES

A.1 Convergence Results for Displacement-Based SGR

Root Stress - Enforced Traction BC (stress SGR)

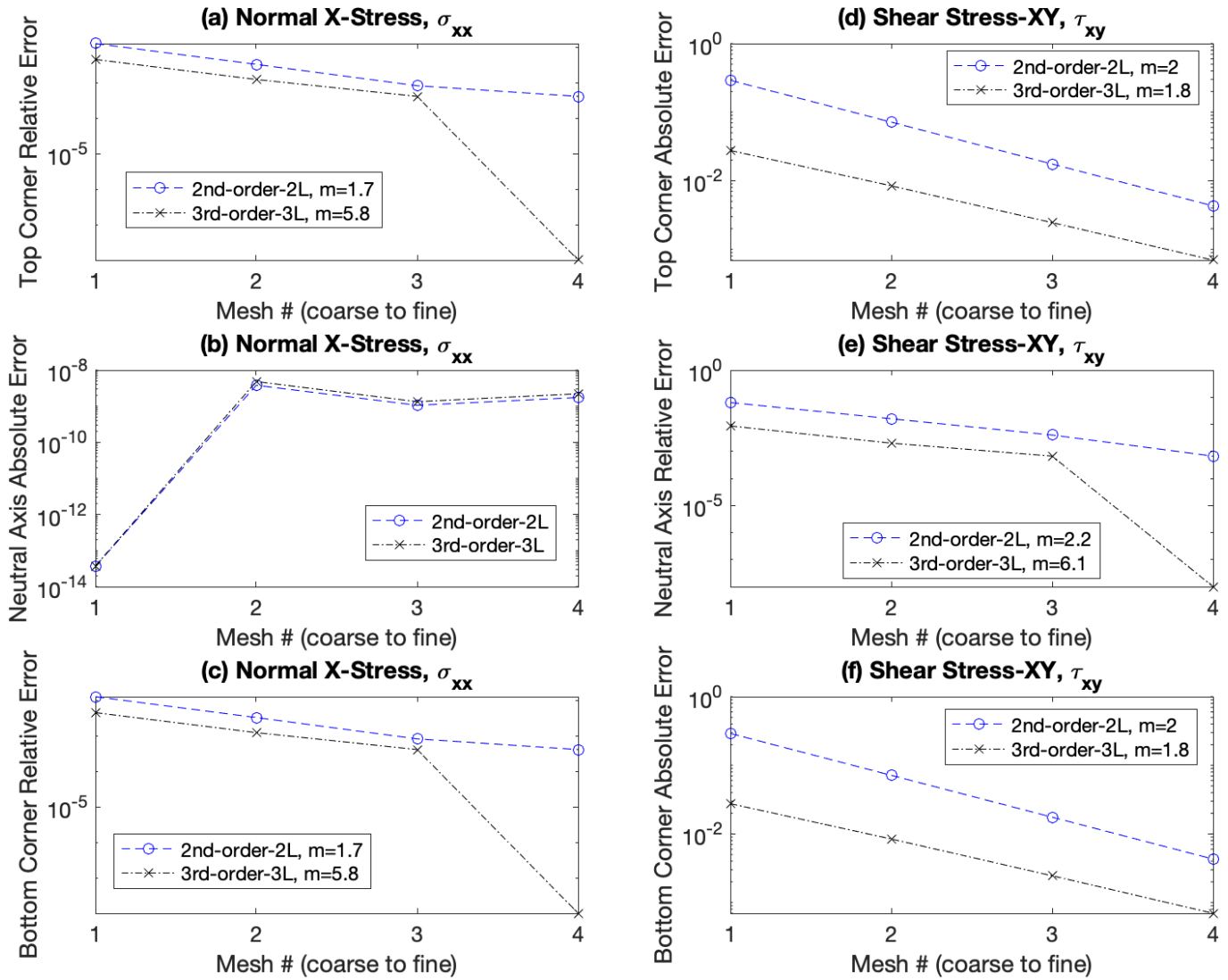


Figure A.1: Displacement-Based SGR Convergence at Root

Root Stress - Enforced Traction BC (stress SGR) -- Exact Displacements

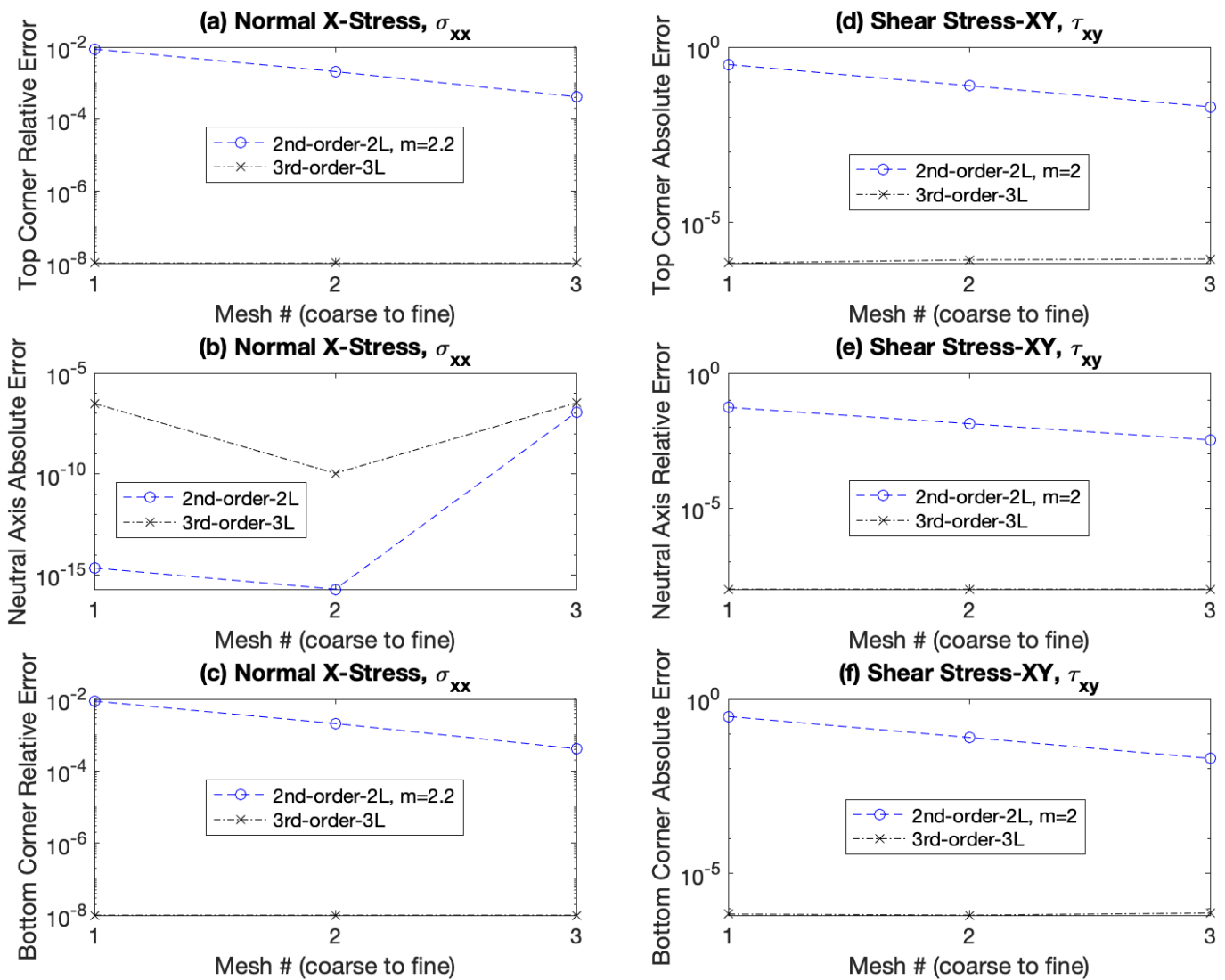


Figure A.2: Exact Displacement-Based SGR Convergence at Root

Tip Stress - Displacement SGR

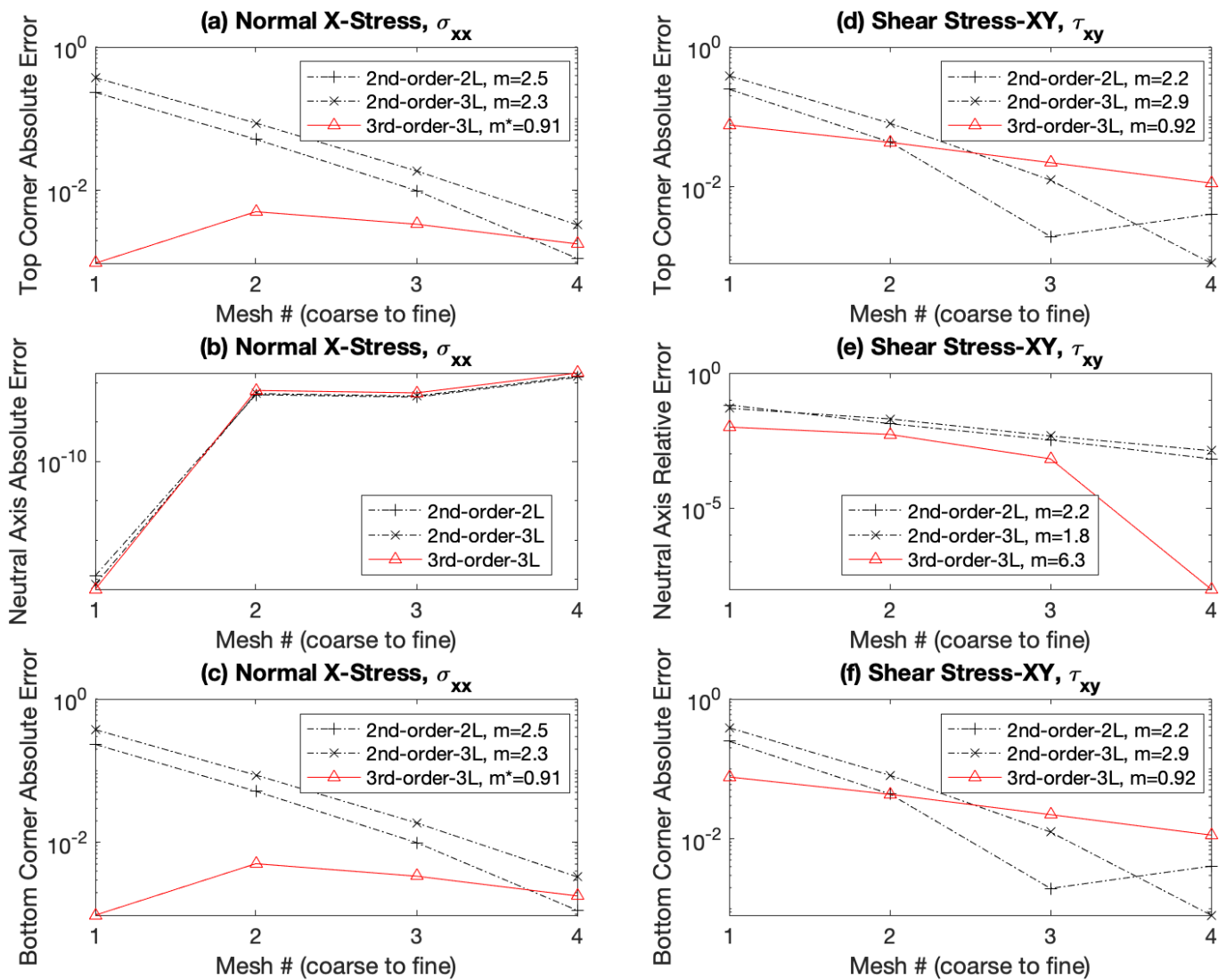


Figure A.3: Displacement-Based SGR Convergence at Tip

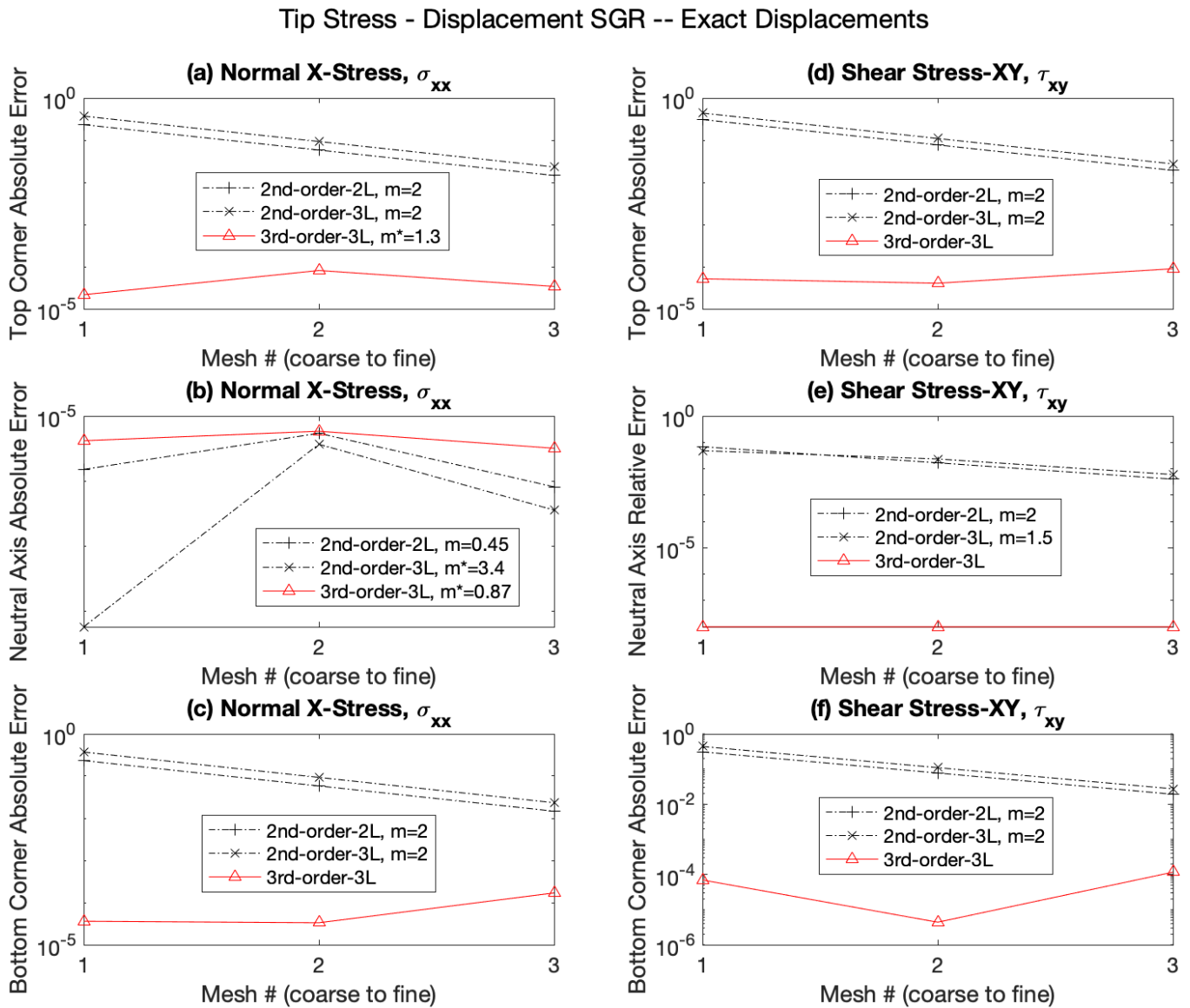


Figure A.4: Exact Displacement-Based SGR Convergence at Tip

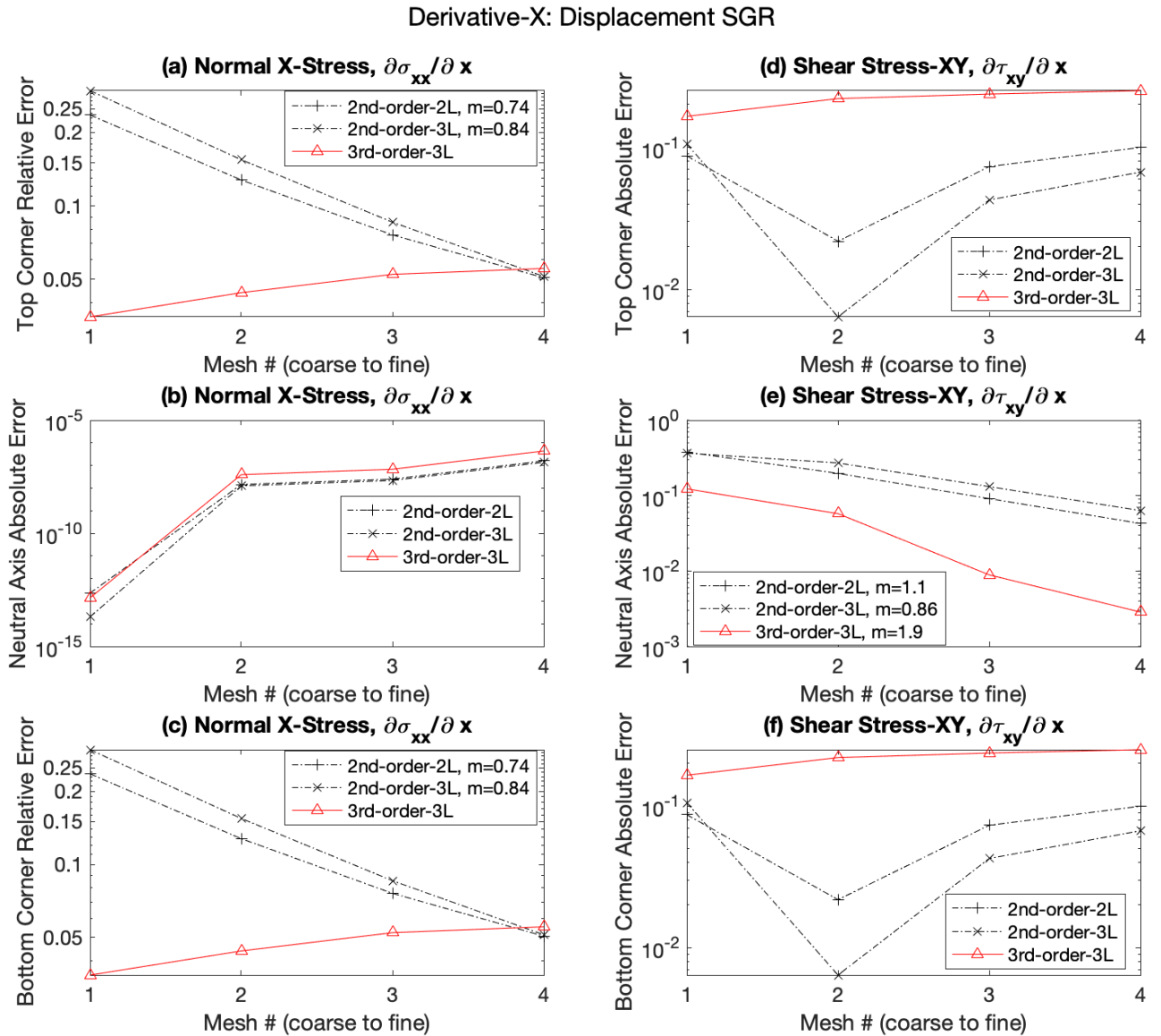


Figure A.5: Displacement-Based SGR Convergence of X-derivatives at Tip

Derivative-Y: Displacement SGR

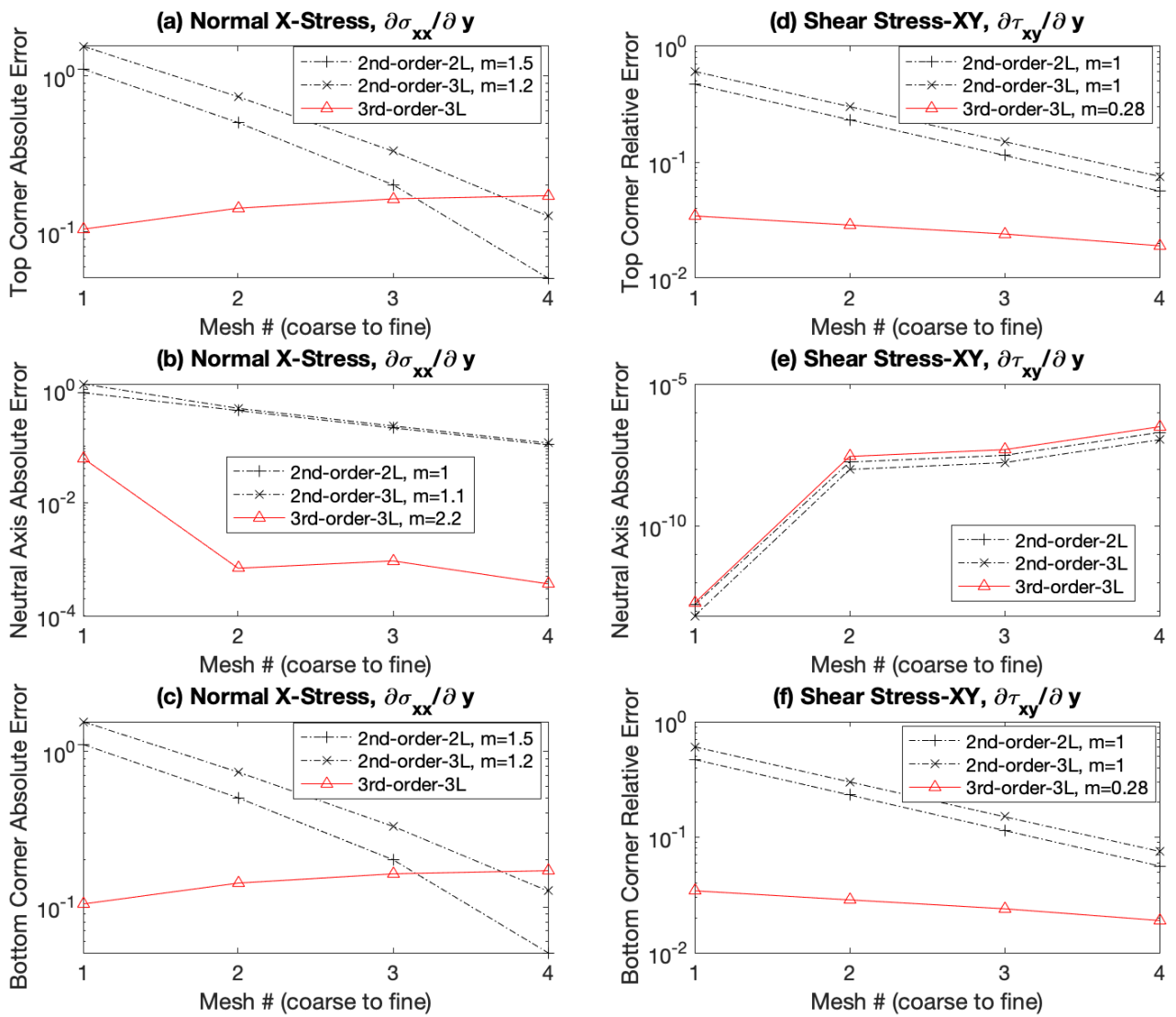


Figure A.6: Displacement-Based SGR Convergence of Y-derivatives at Tip

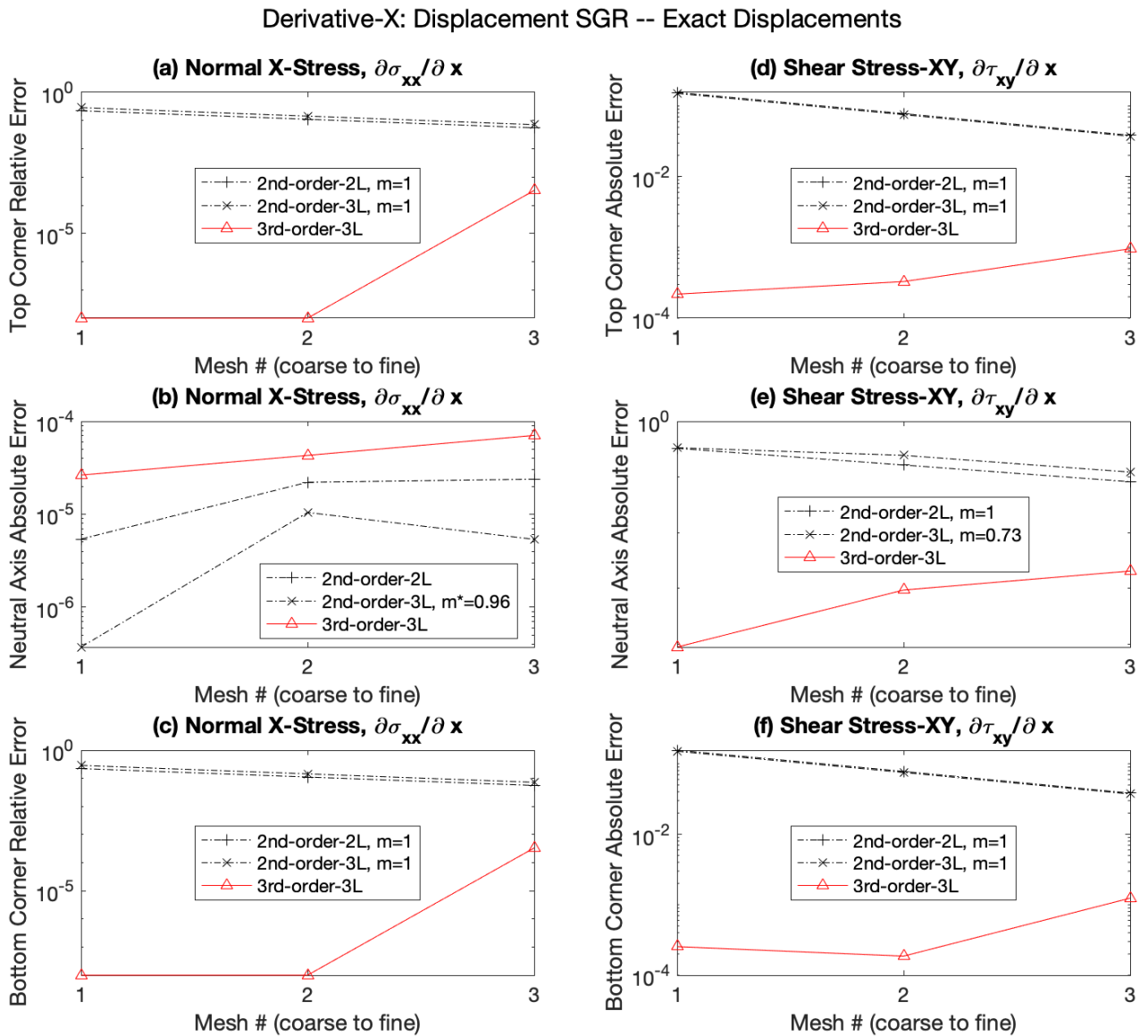


Figure A.7: Exact Displacement-Based SGR of X-derivatives Convergence at Tip

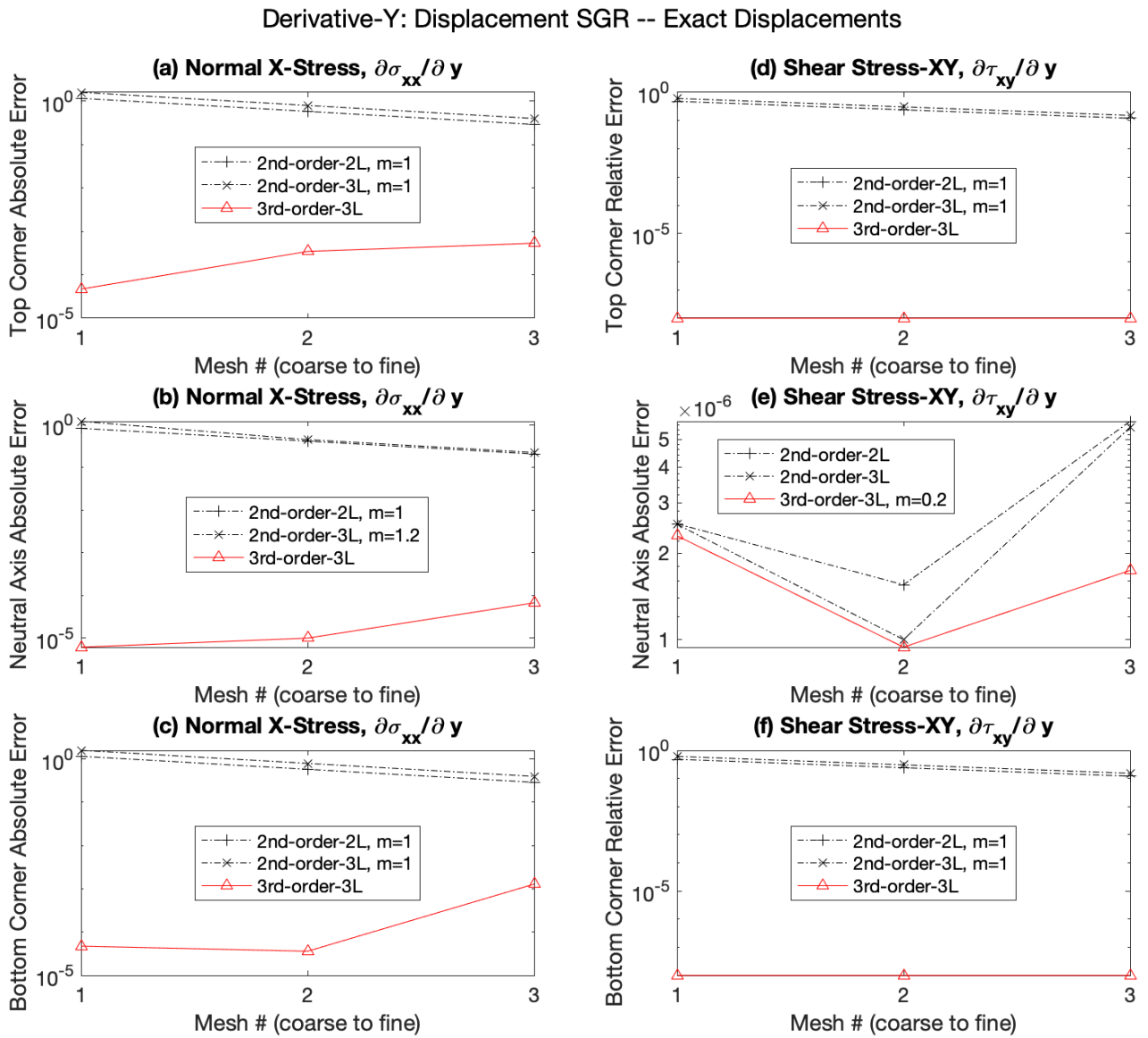


Figure A.8: Exact Displacement-Based SGR of Y-derivatives Convergence at Tip

A.2 Convergence Results for Stress-Based SGR without Traction Enforcement

Tip Stress - No Traction BC (stress SGR)

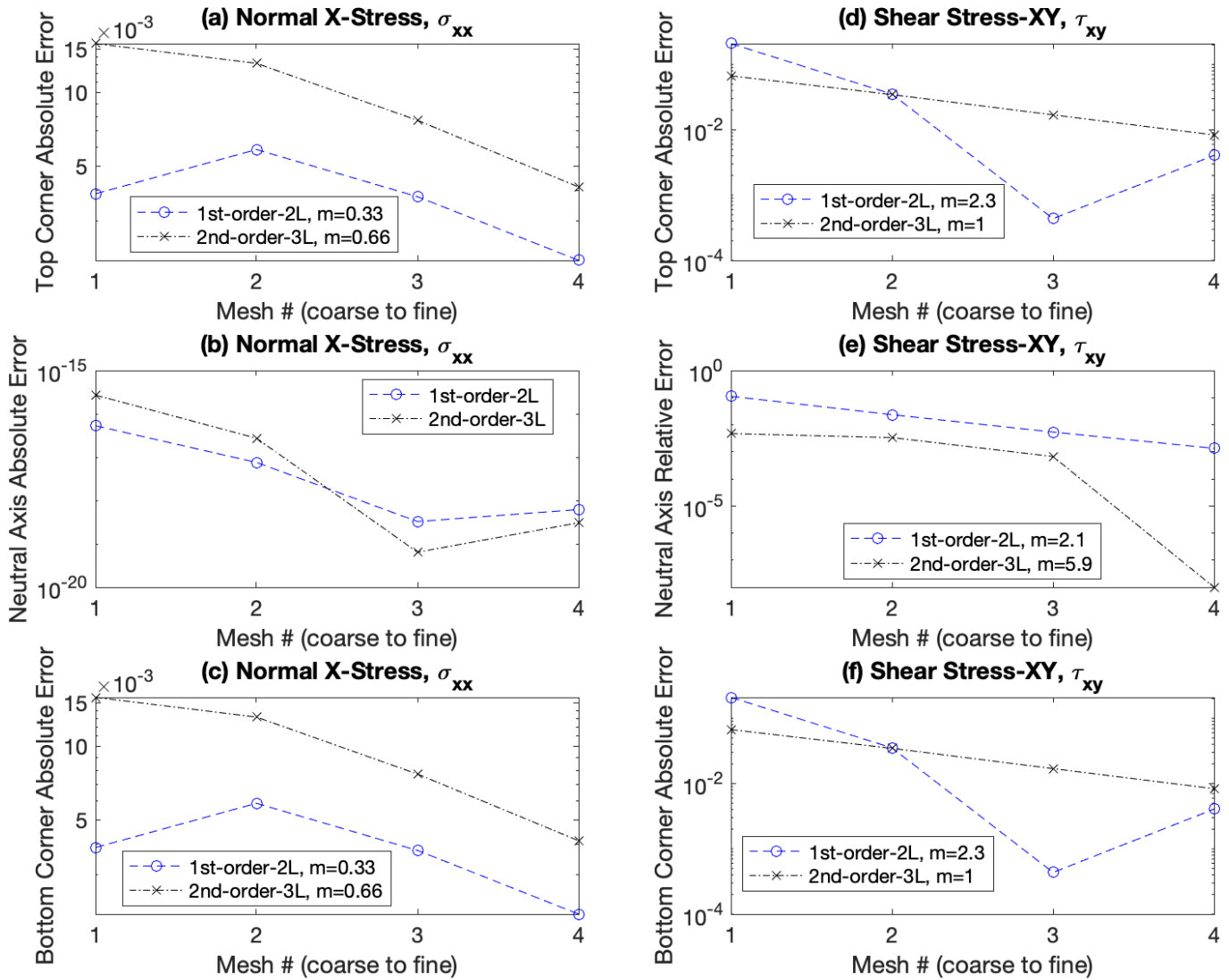


Figure A.9: Stress-Based SGR without Traction Convergence at Tip

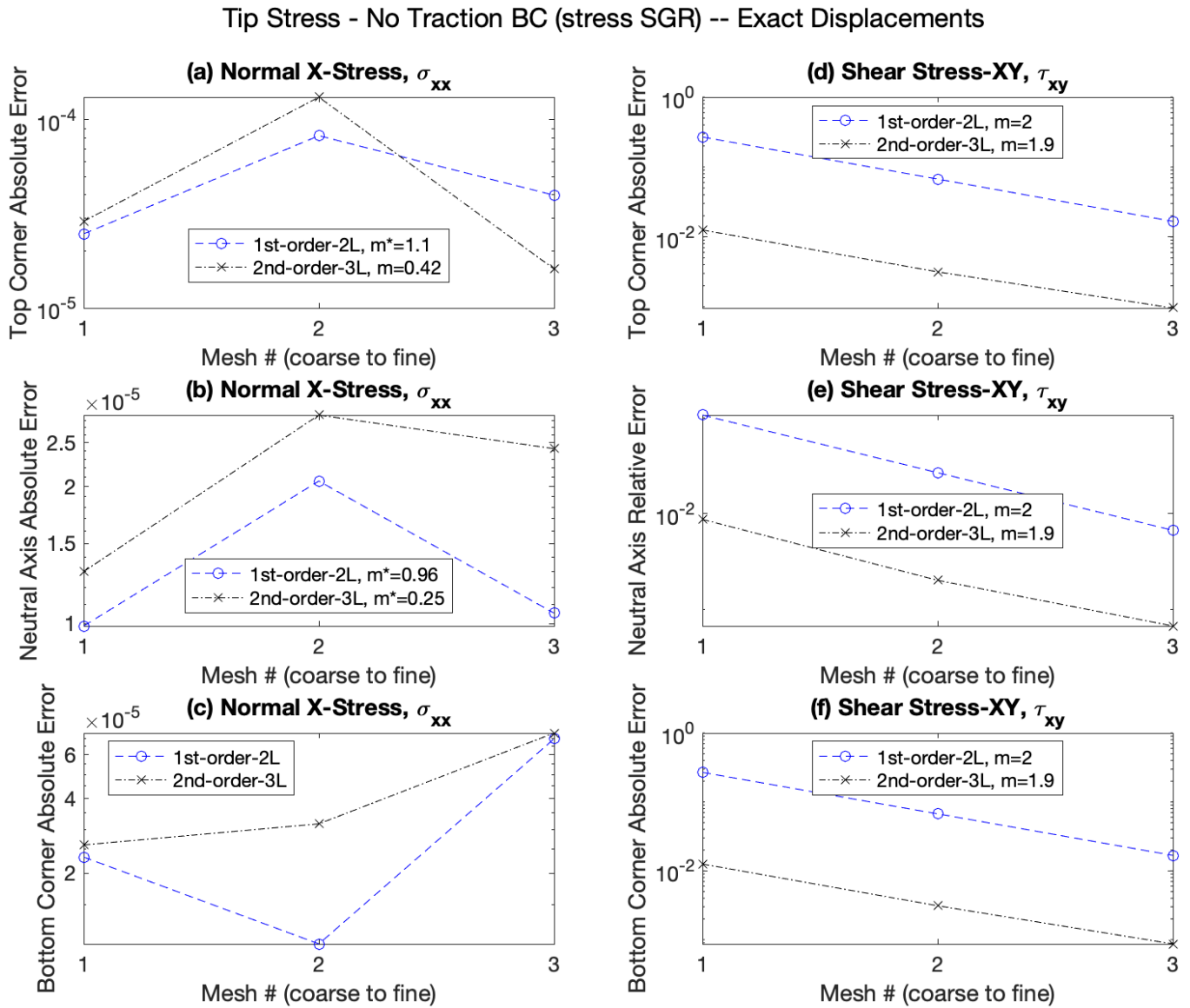


Figure A.10: Exact Displacement, Stress-Based SGR without Traction Convergence at Tip

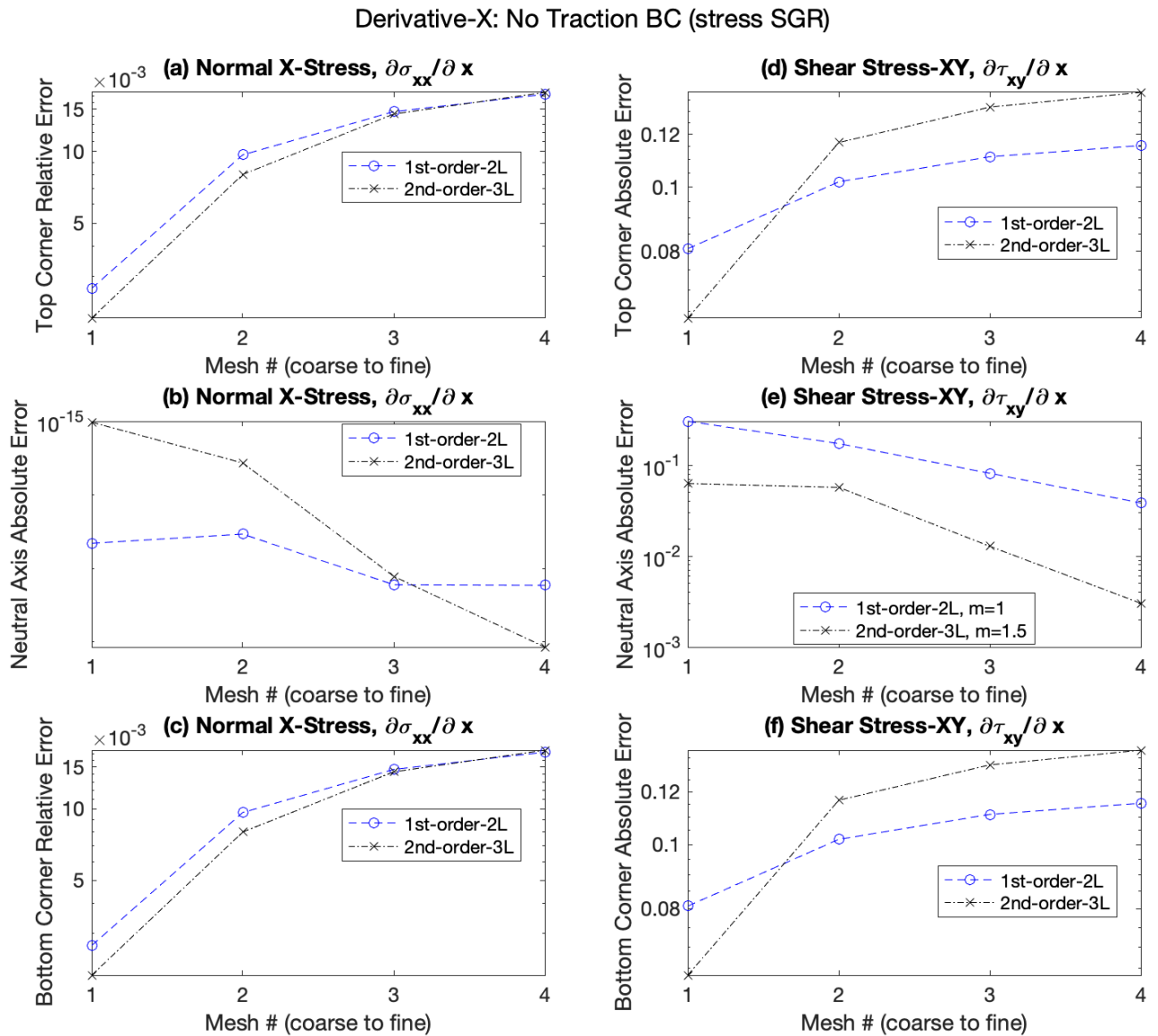


Figure A.11: Stress-Based SGR without Traction Convergence of X-derivatives at Tip

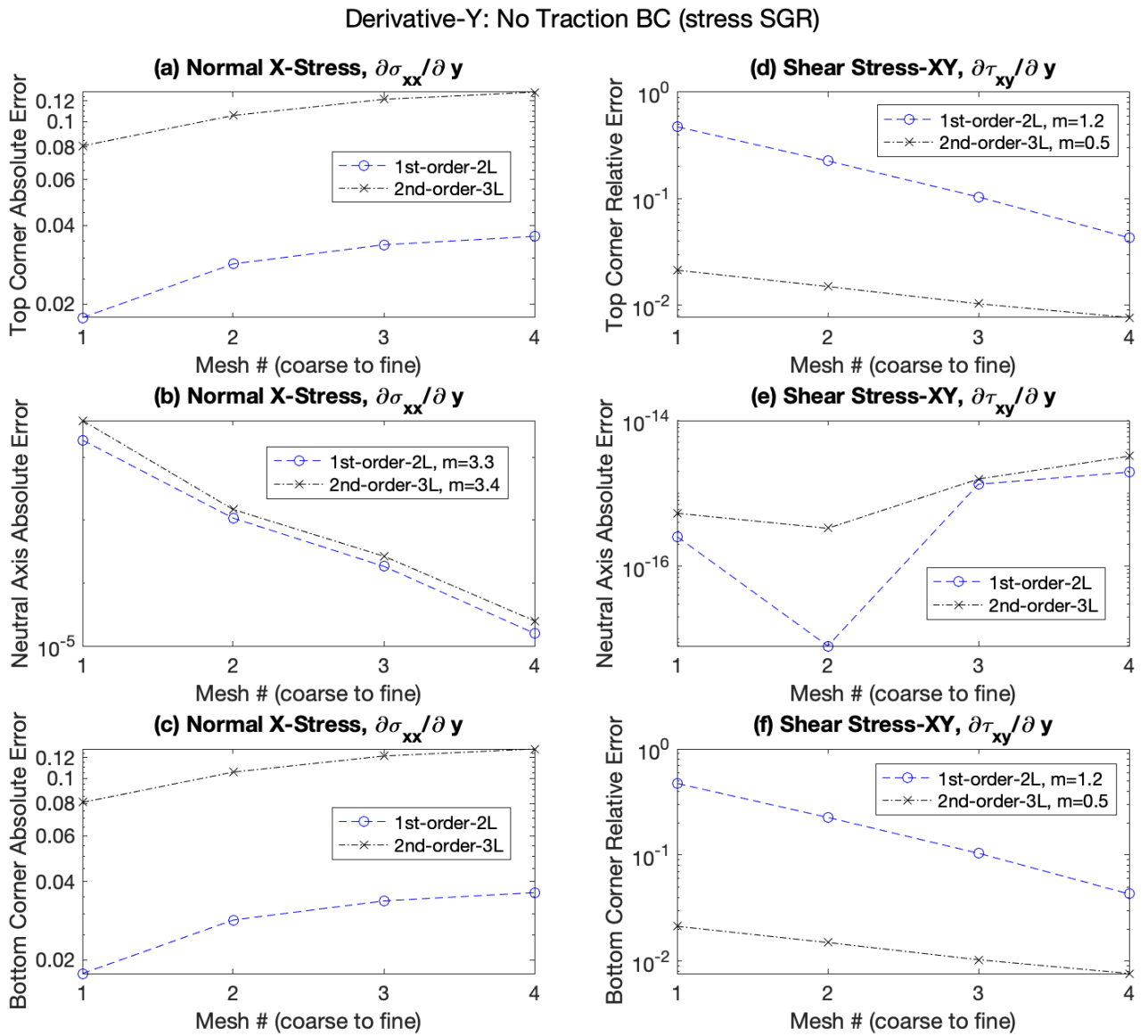


Figure A.12: Stress-Based SGR without Traction Convergence of Y-derivatives at Tip

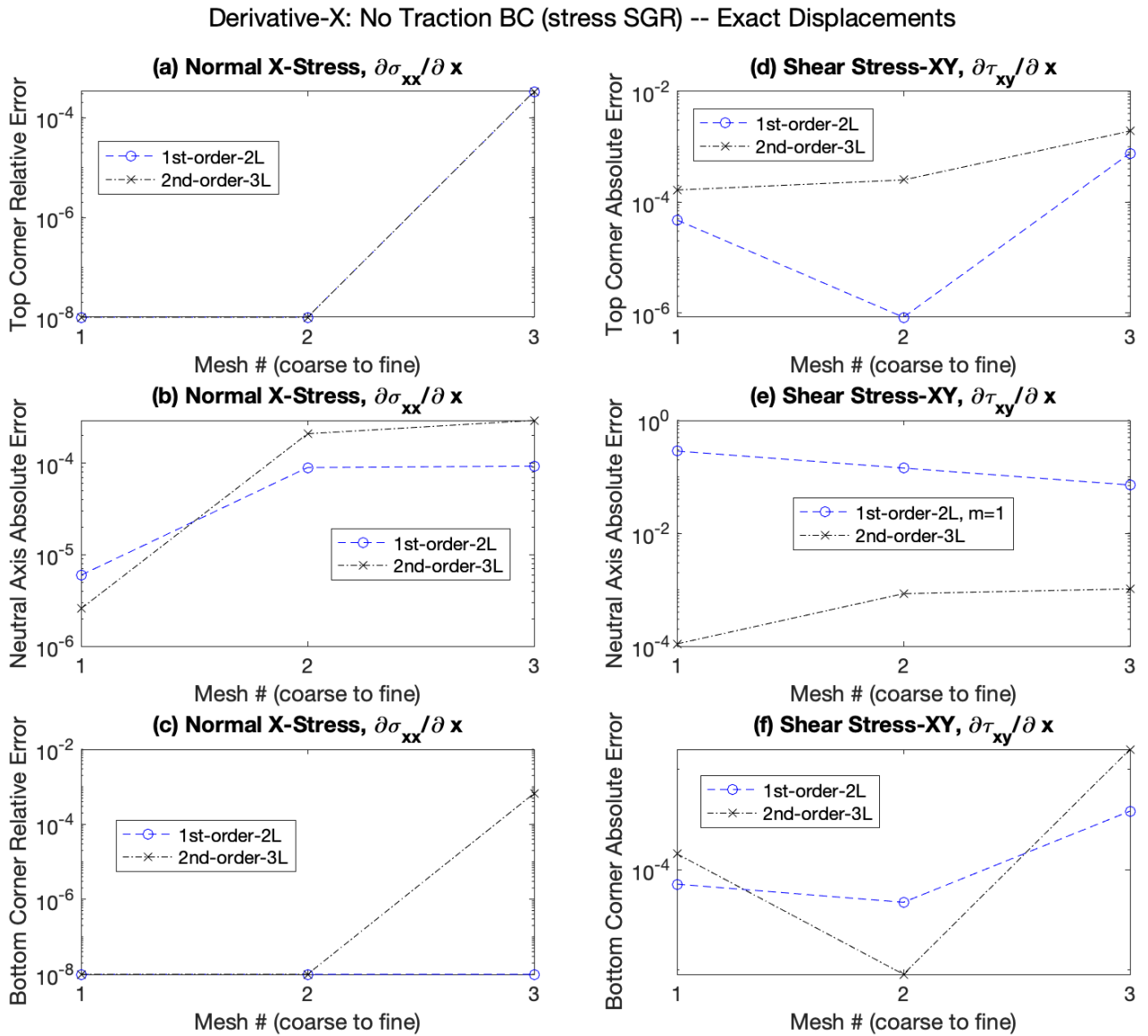


Figure A.13: Exact Displacement, Stress-Based SGR without Traction Convergence of X-derivatives at Tip

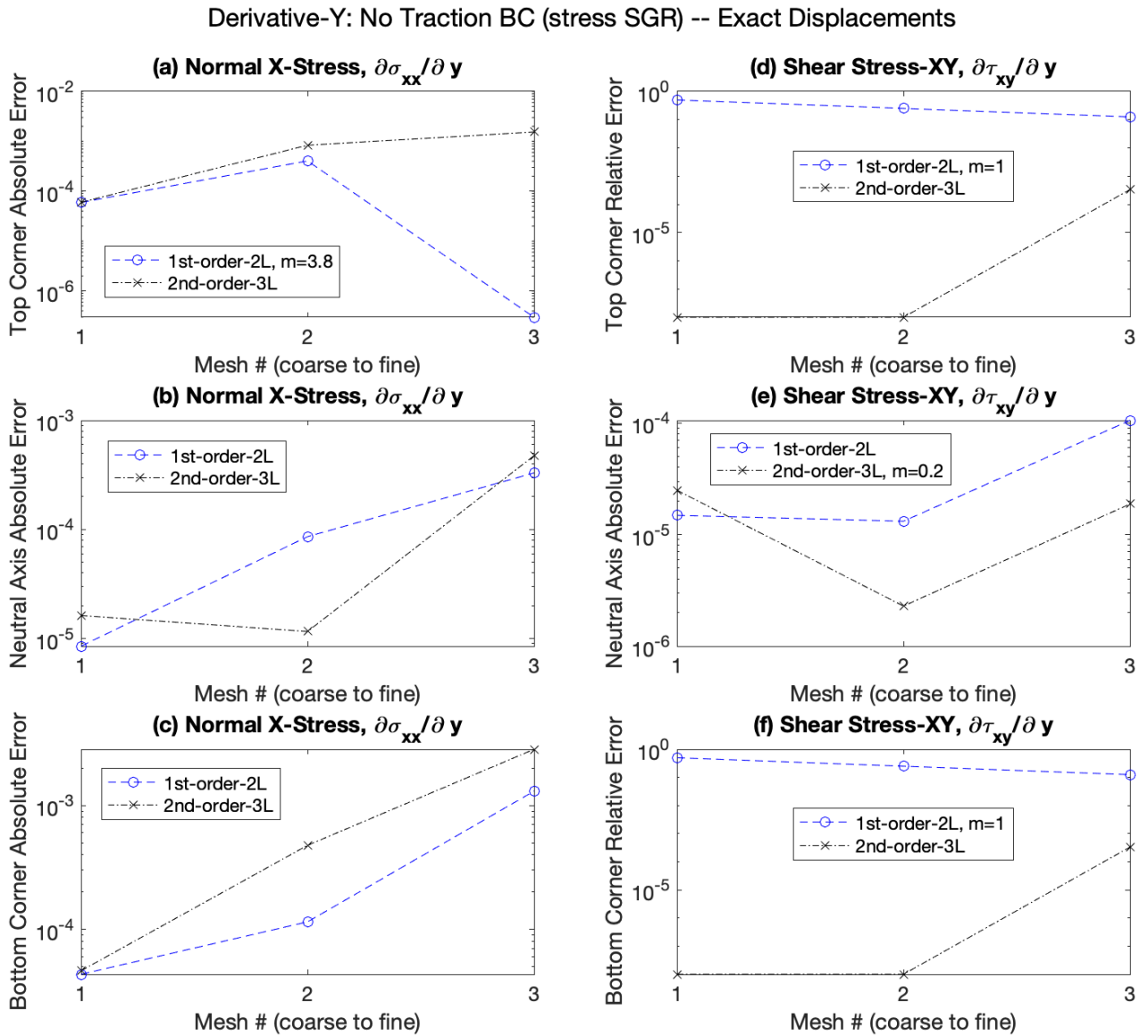


Figure A.14: Exact Displacement, Stress-Based SGR without Traction Convergence of Y-derivatives at Tip

A.3 Convergence Results for Stress-Based SGR with Traction Enforced

Tip Stress - Enforced Traction BC (stress SGR)

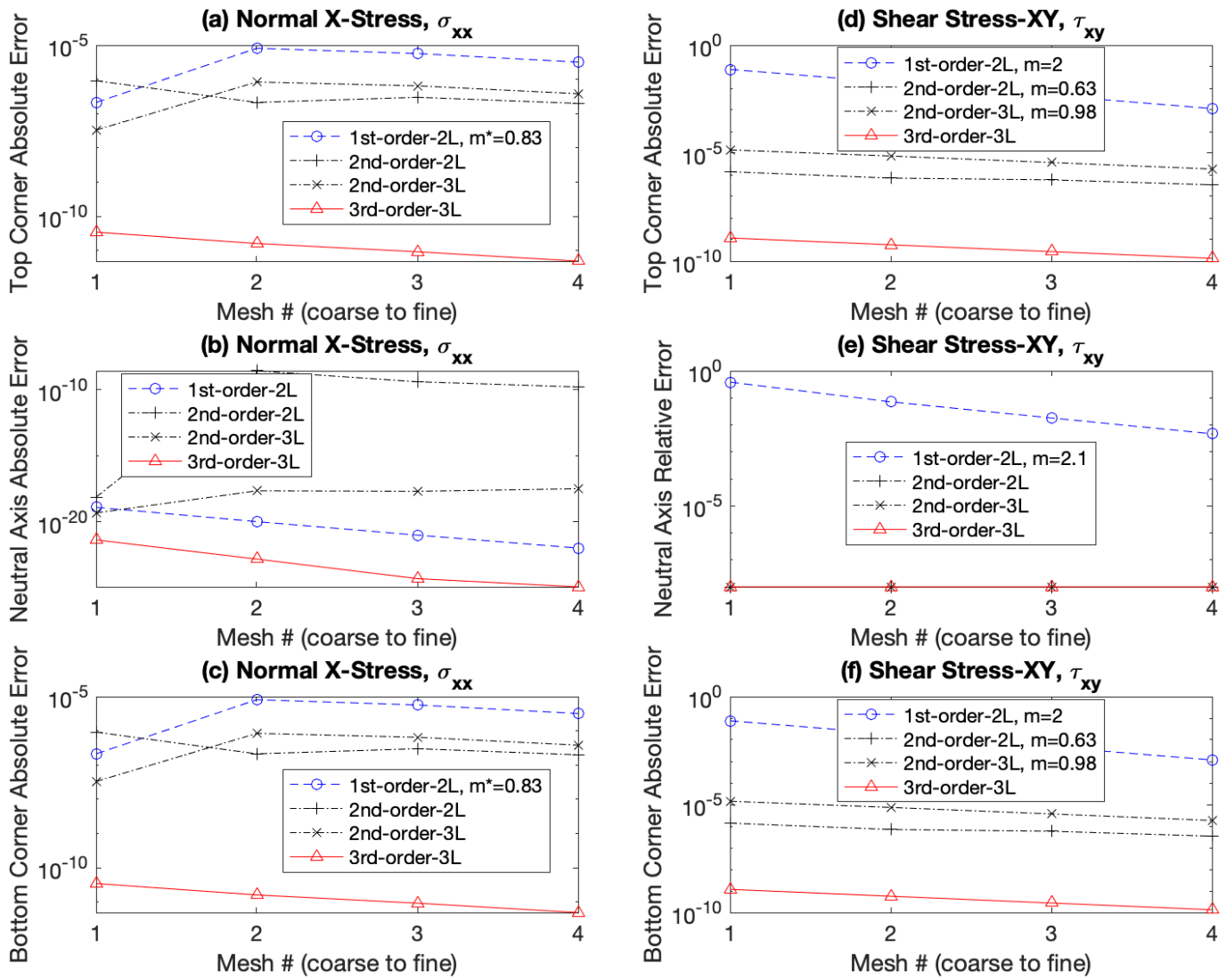


Figure A.15: Traction-Enforced Stress-Based SGR Convergence at Tip

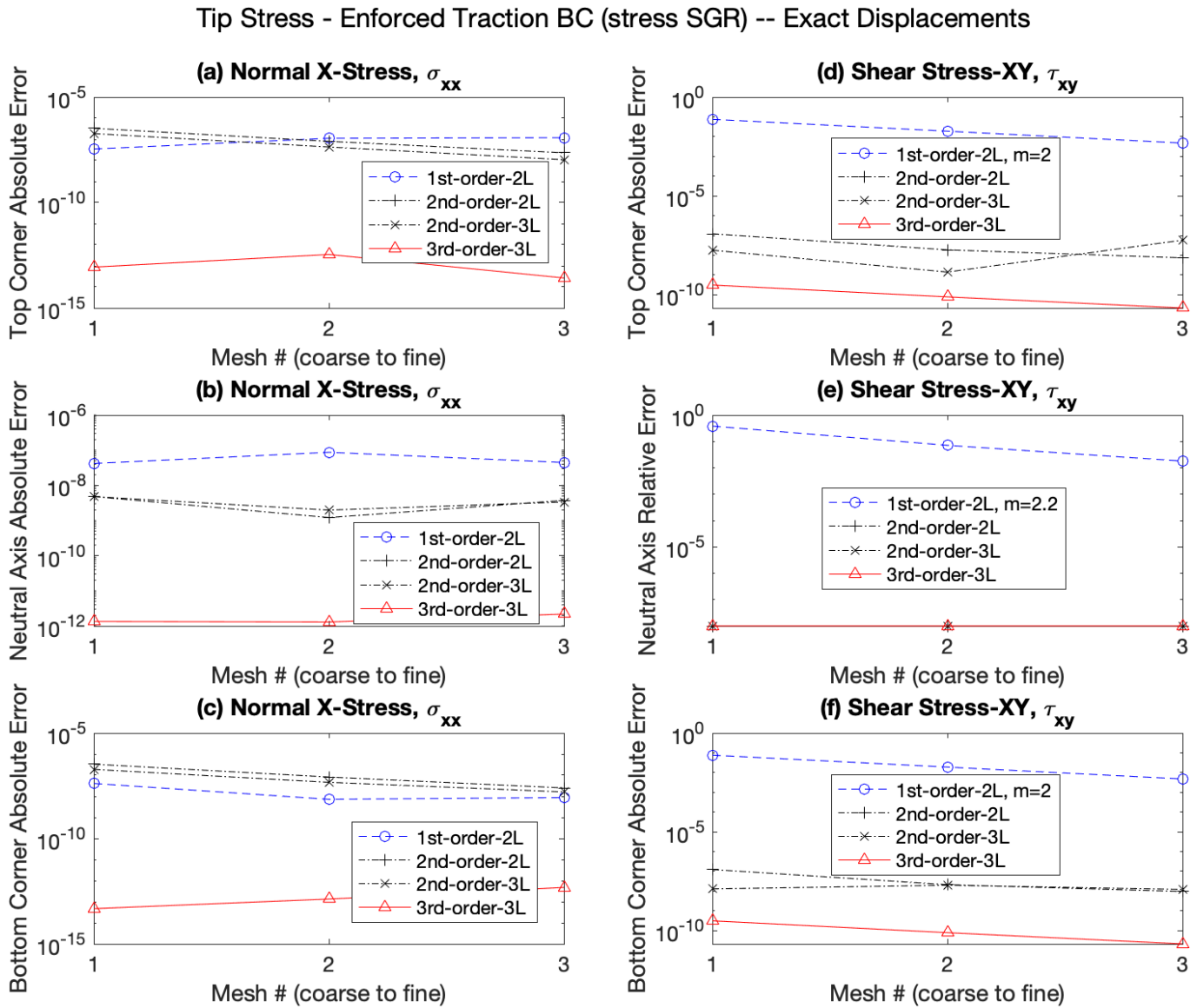


Figure A.16: Exact Displacement, Traction-Enforced Stress-Based SGR Convergence at Tip

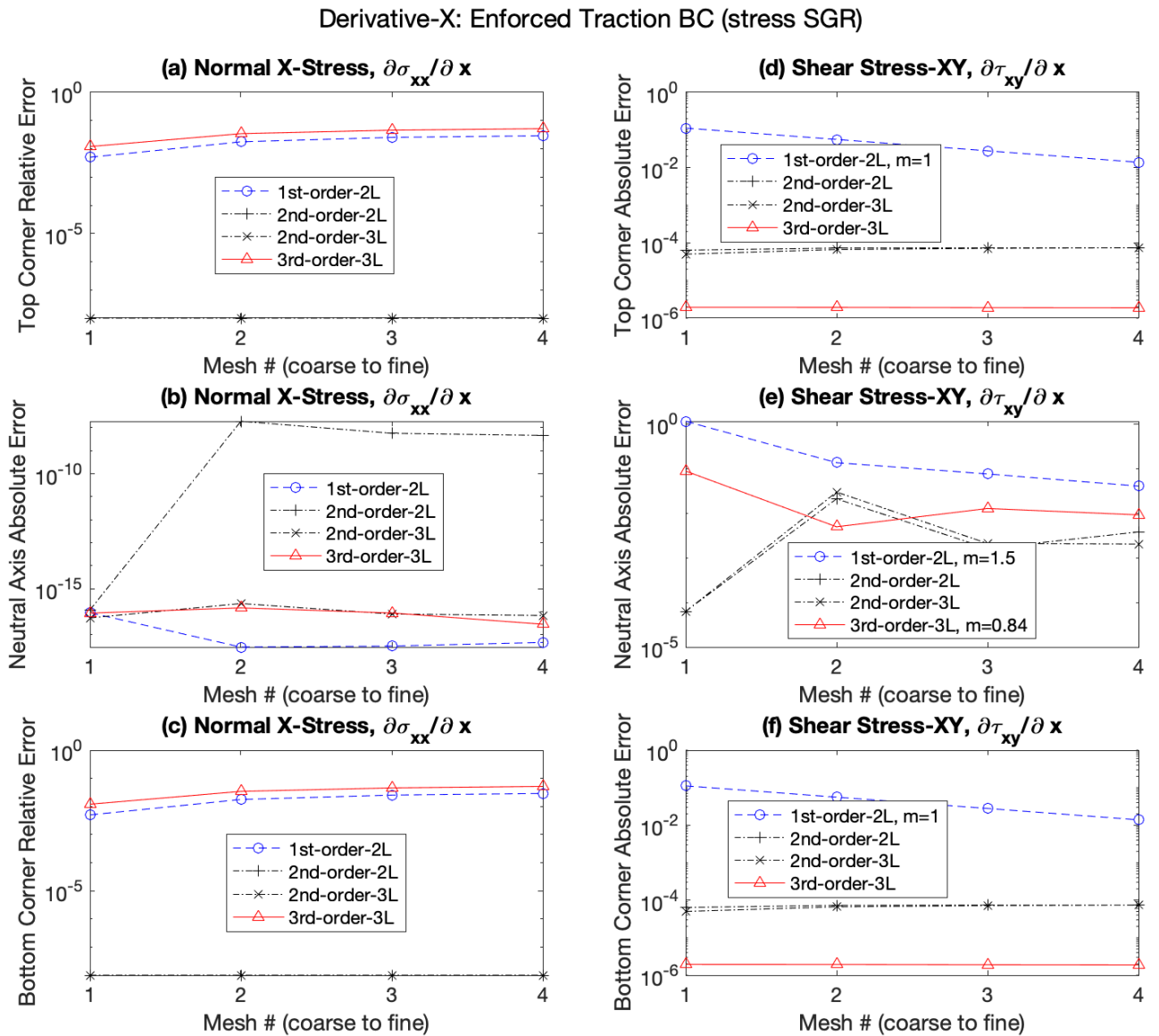


Figure A.17: Traction-Enforced Stress-Based SGR Convergence of X-derivatives at Tip

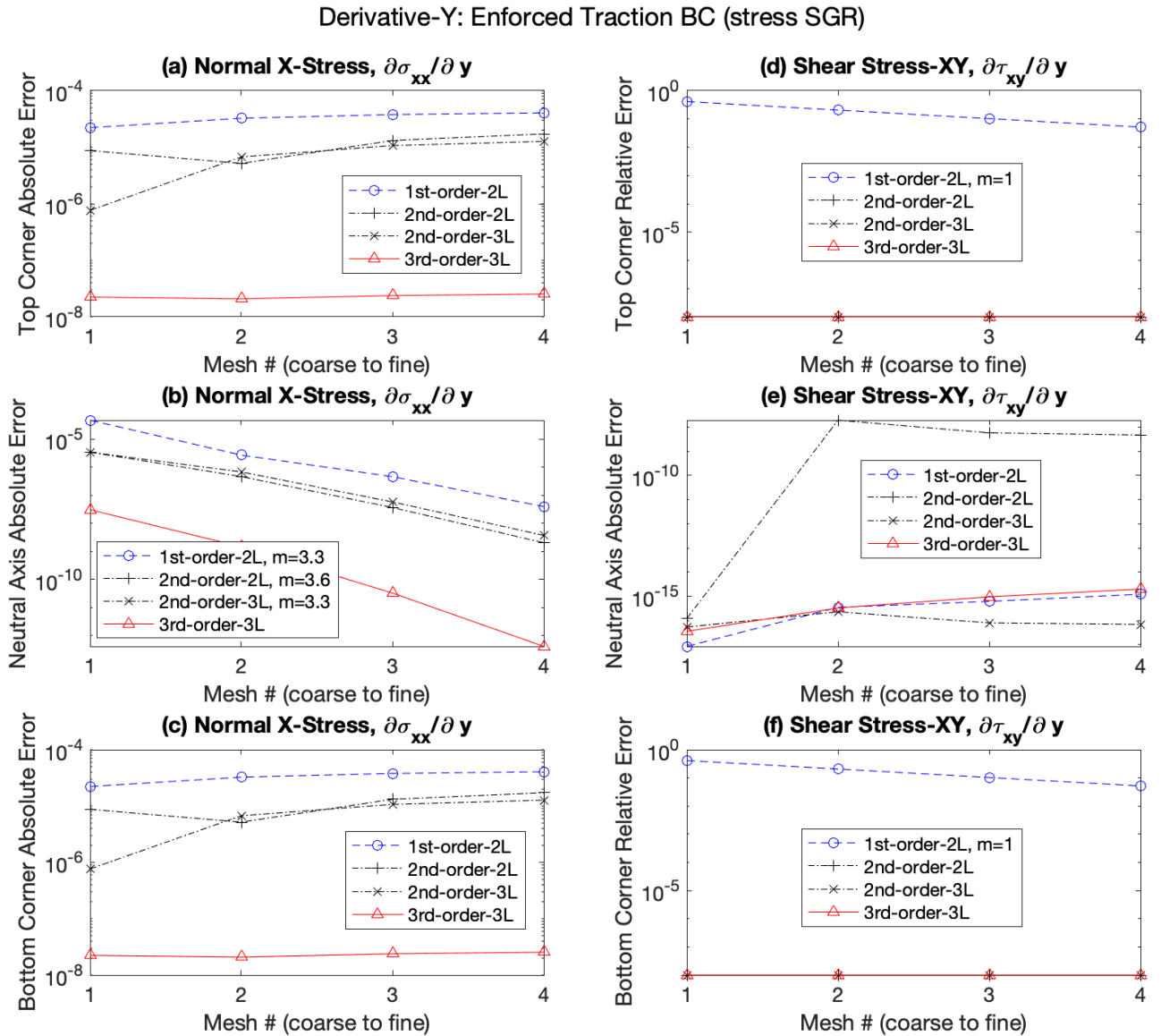


Figure A.18: Traction-Enforced Stress-Based SGR Convergence of Y-derivatives at Tip

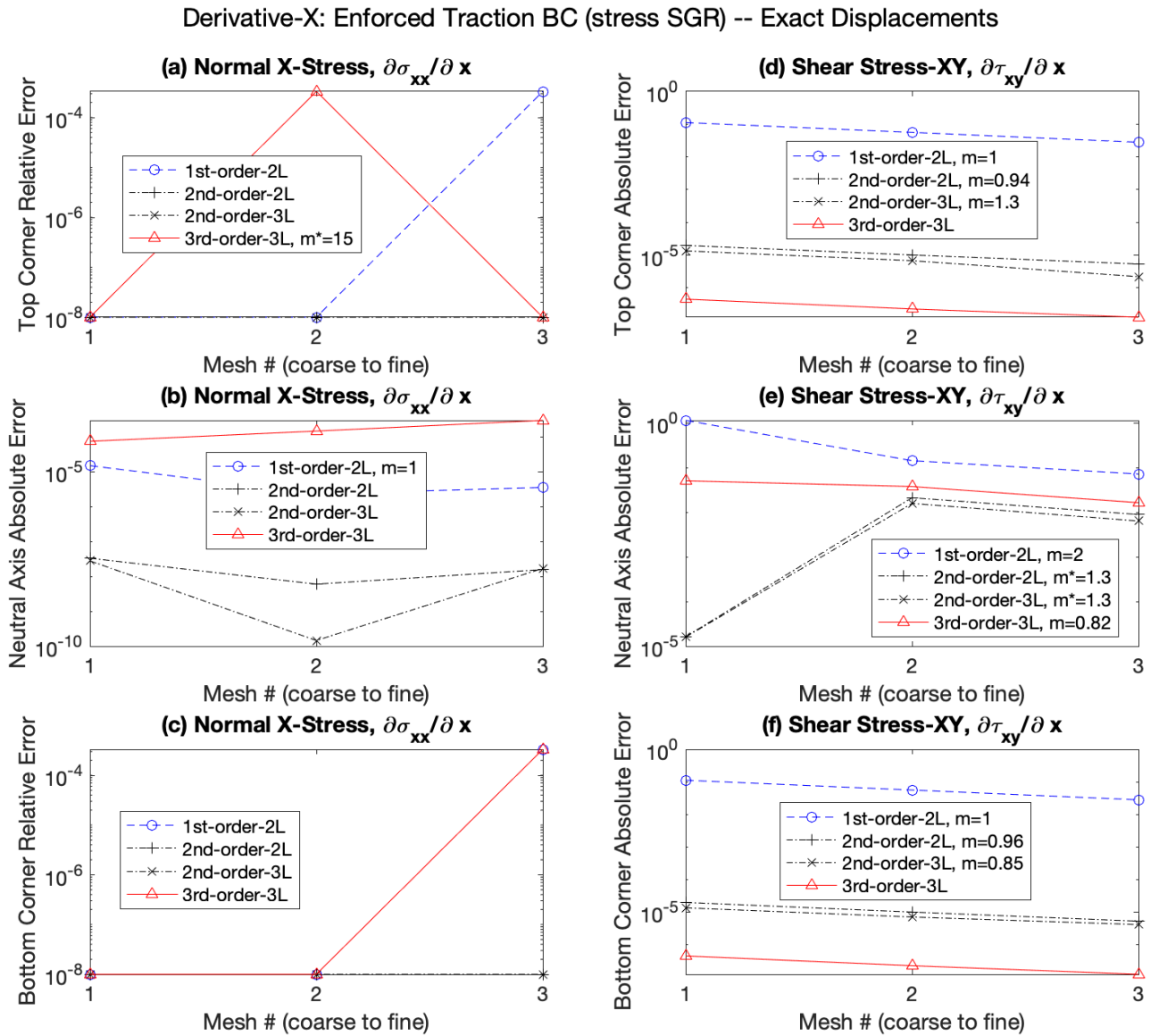


Figure A.19: Exact Displacement, Traction-Enforced Stress-Based SGR Convergence of X-derivatives at Tip

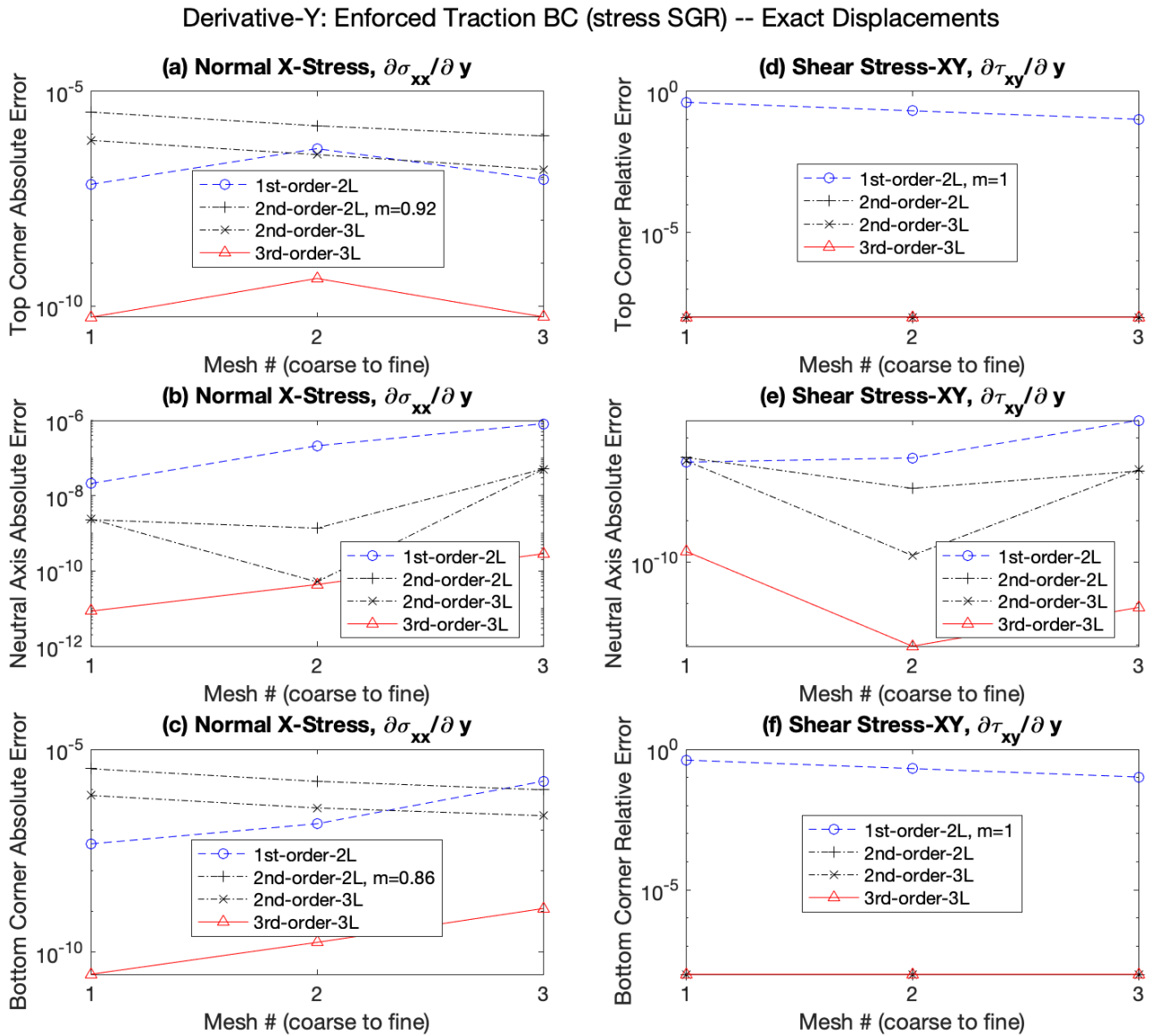


Figure A.20: Exact Displacement, Traction-Enforced Stress-Based SGR Convergence of Y-derivatives at Tip

Robert A. Canfield

B NASTRAN TRIA3 TRIANGULAR MESHES

B.1 Convergence Results for Displacement-Based SGR

Root Stress - Enforced Traction BC (stress SGR) NASTRAN

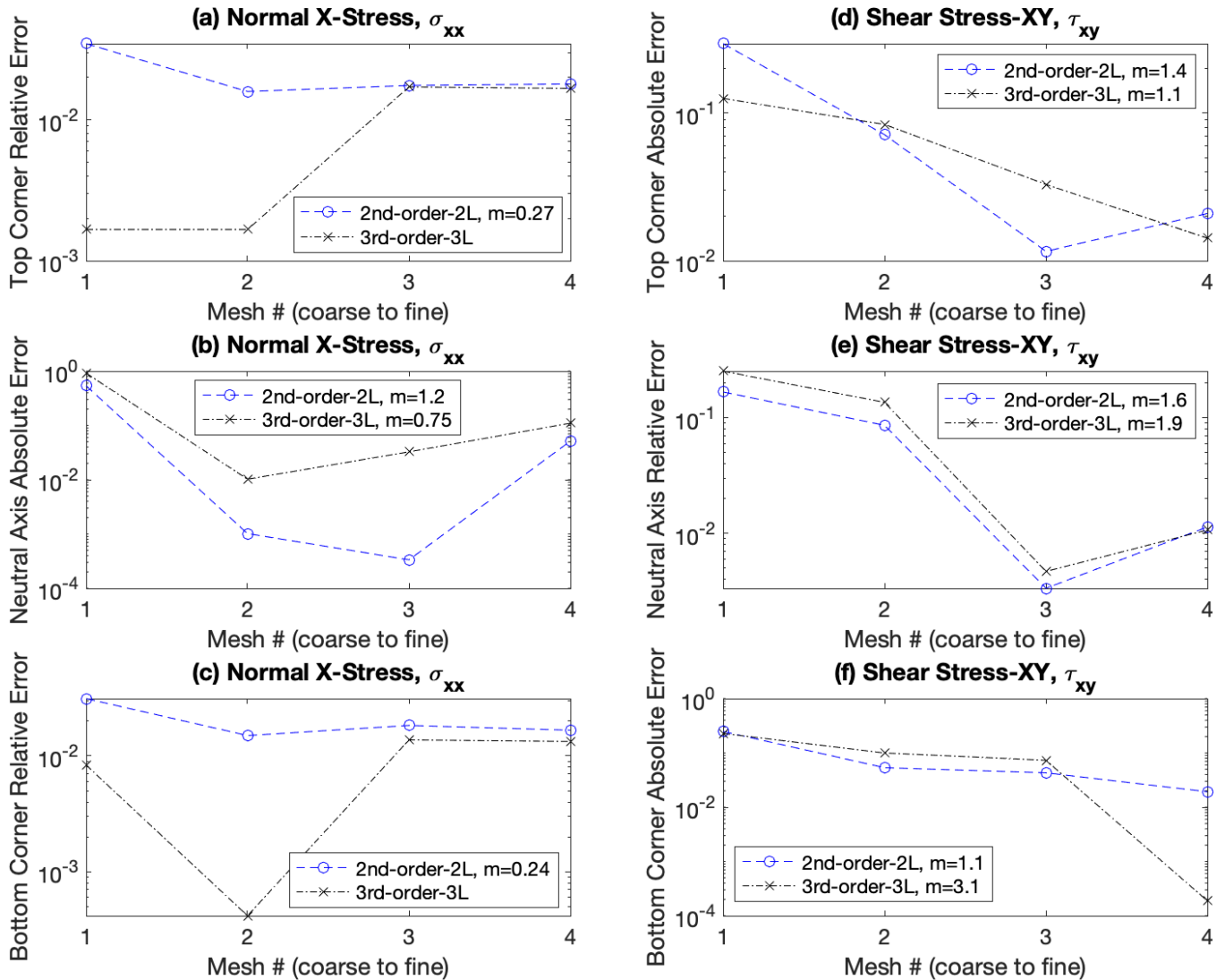


Figure B.1: Displacement-Based SGR Convergence at Root (NASTRA)

Tip Stress - Displacement SGR NASTRAN

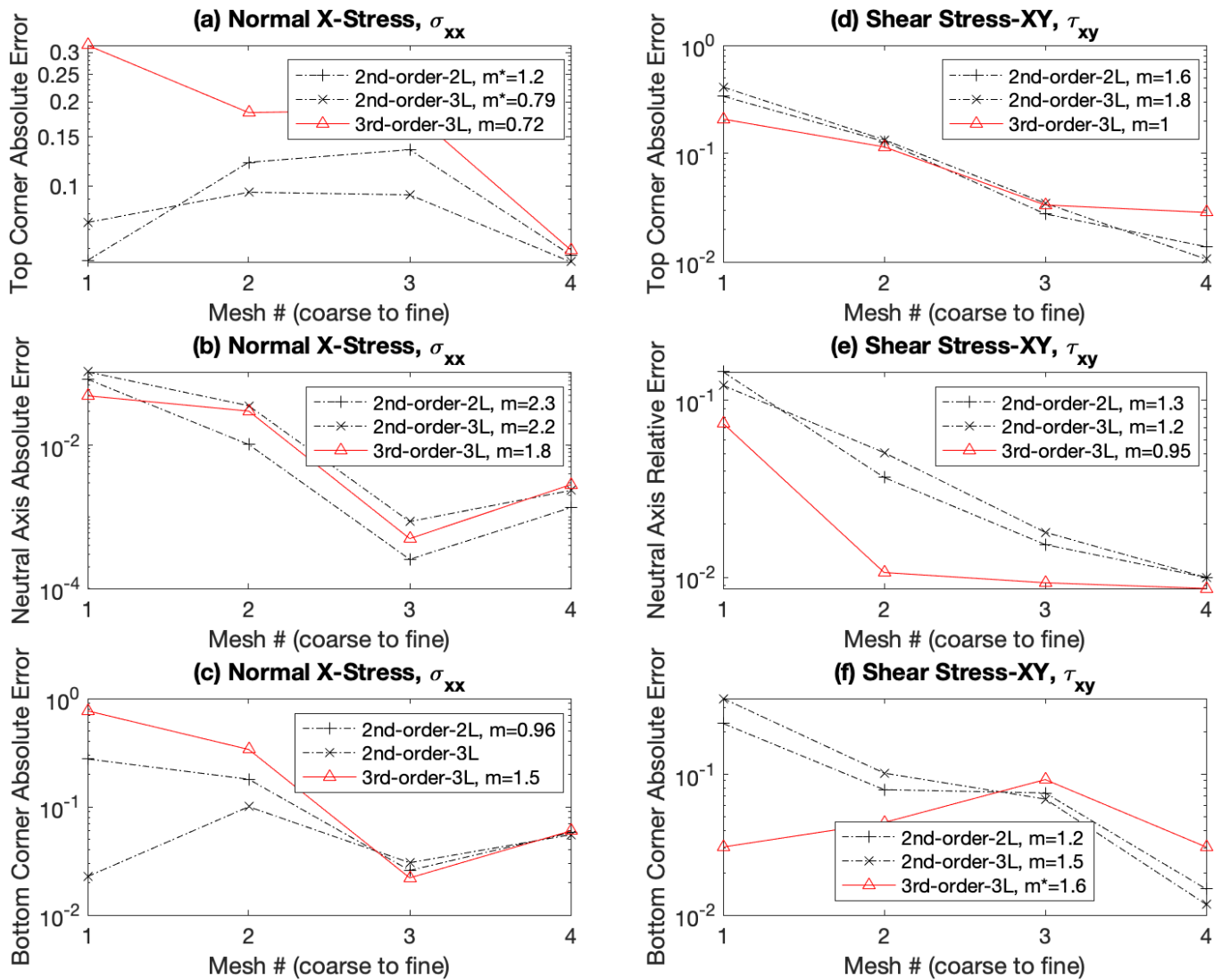


Figure B.2: Displacement-Based SGR Convergence at Tip (NASTRA)

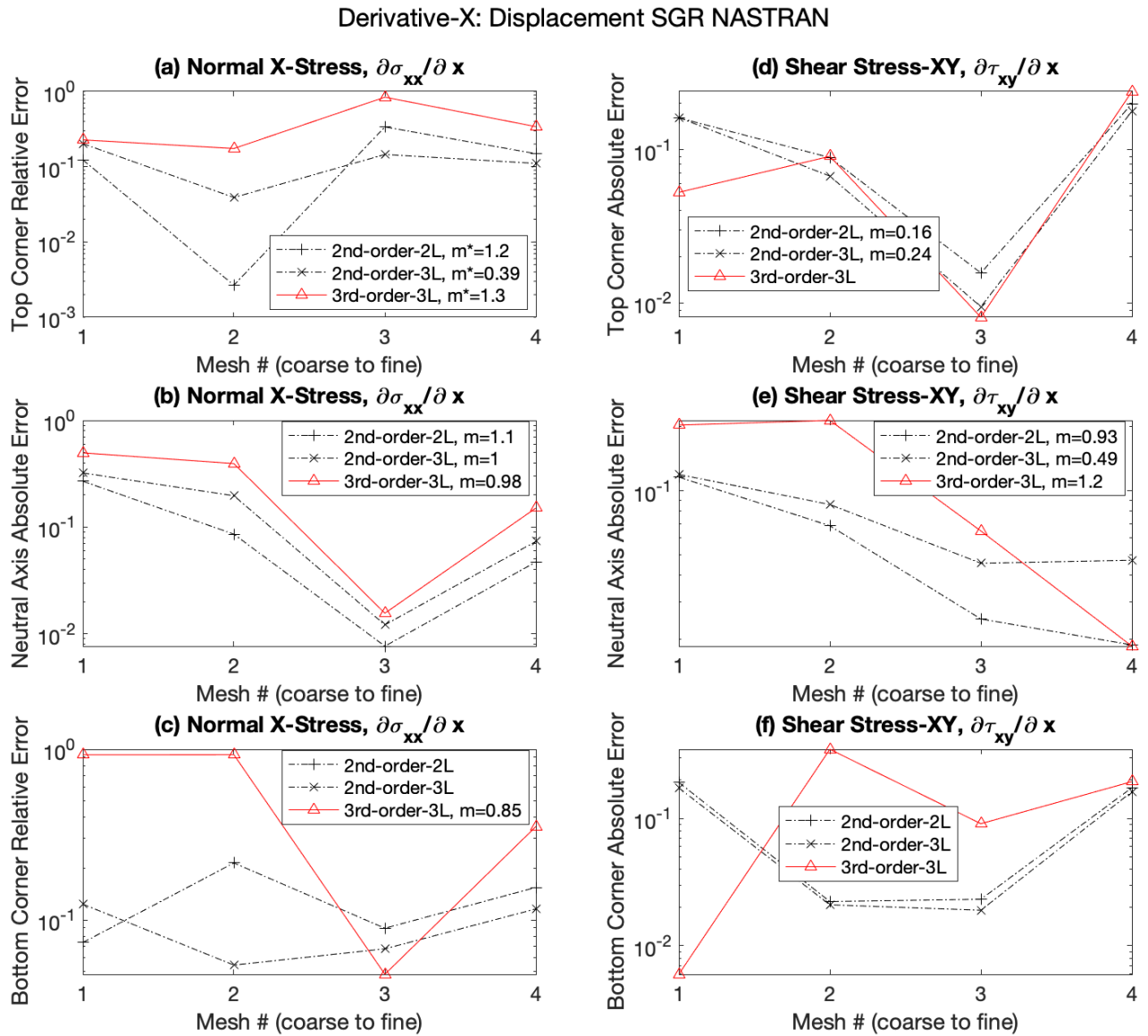


Figure B.3: Displacement-Based SGR of X-derivatives Convergence at Tip (NASTRAN TRIA3)

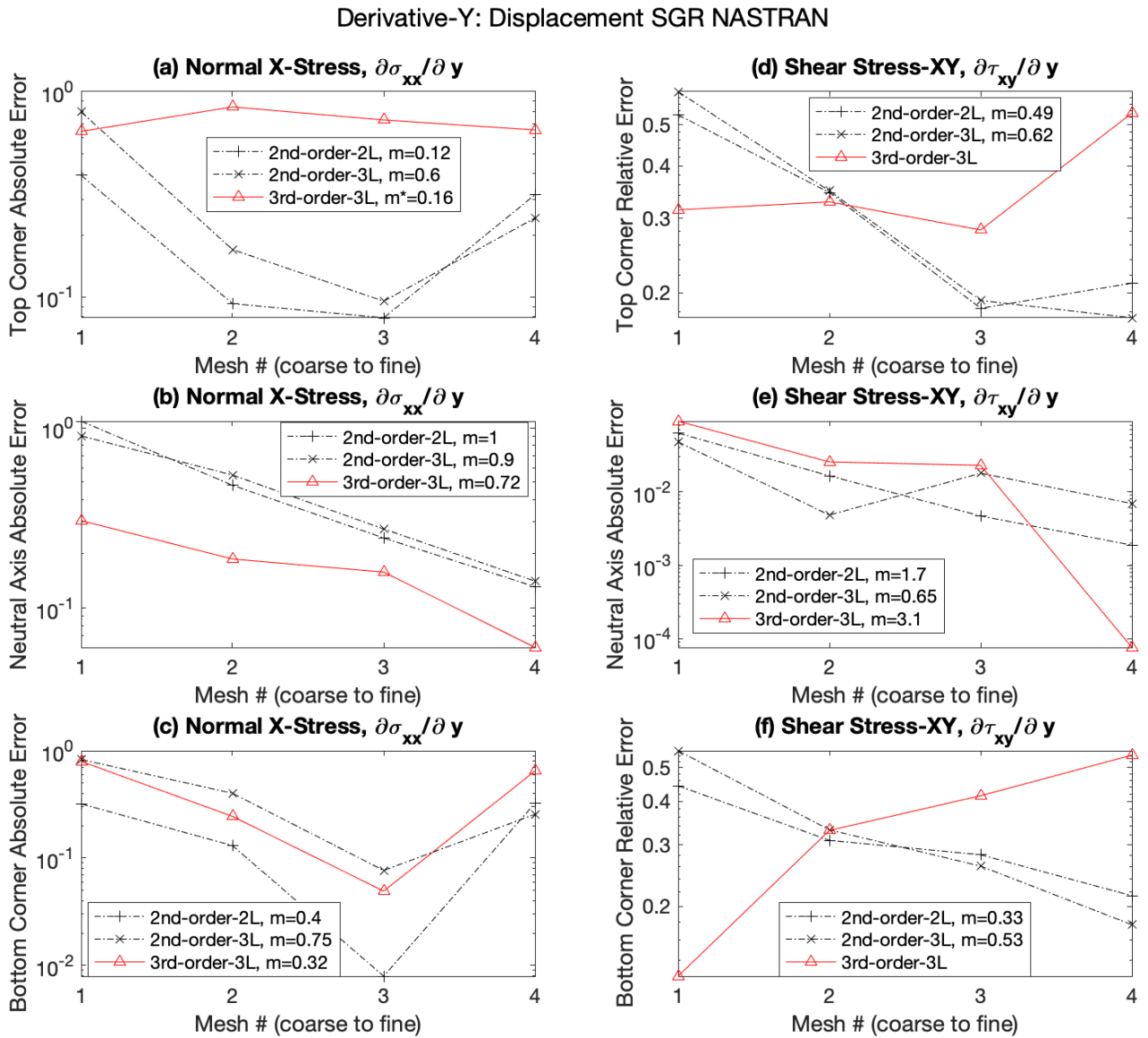


Figure B.4: Displacement-Based SGR of Y-derivatives Convergence at Tip (NASTRAN TRIA3)

B.2 Convergence Results for Stress-Based SGR without Traction Enforcement

Tip Stress - No Traction BC (stress SGR) NASTRAN

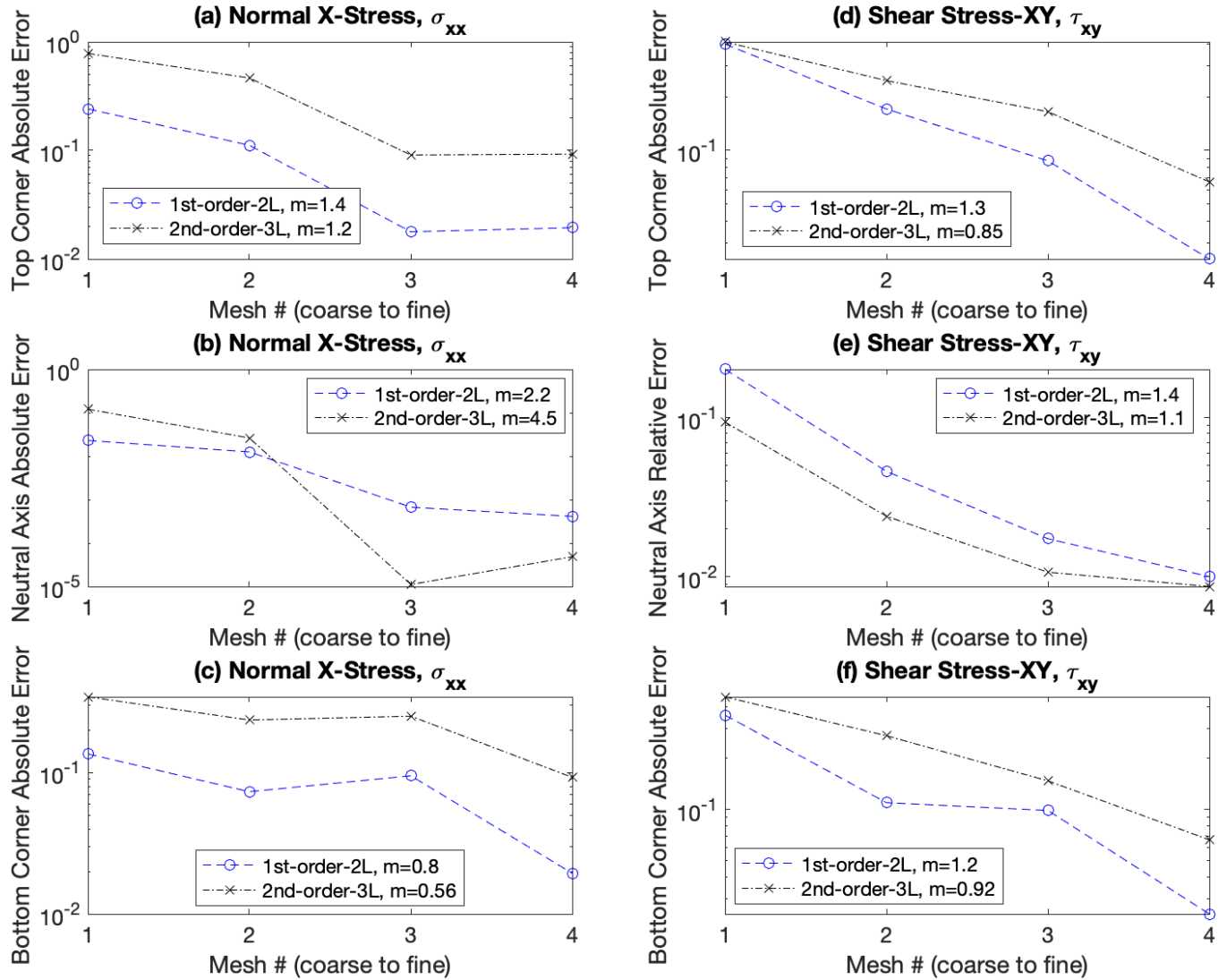


Figure B.5: Stress-Based SGR without Traction Convergence at Tip (NASTRAN TRIA3)

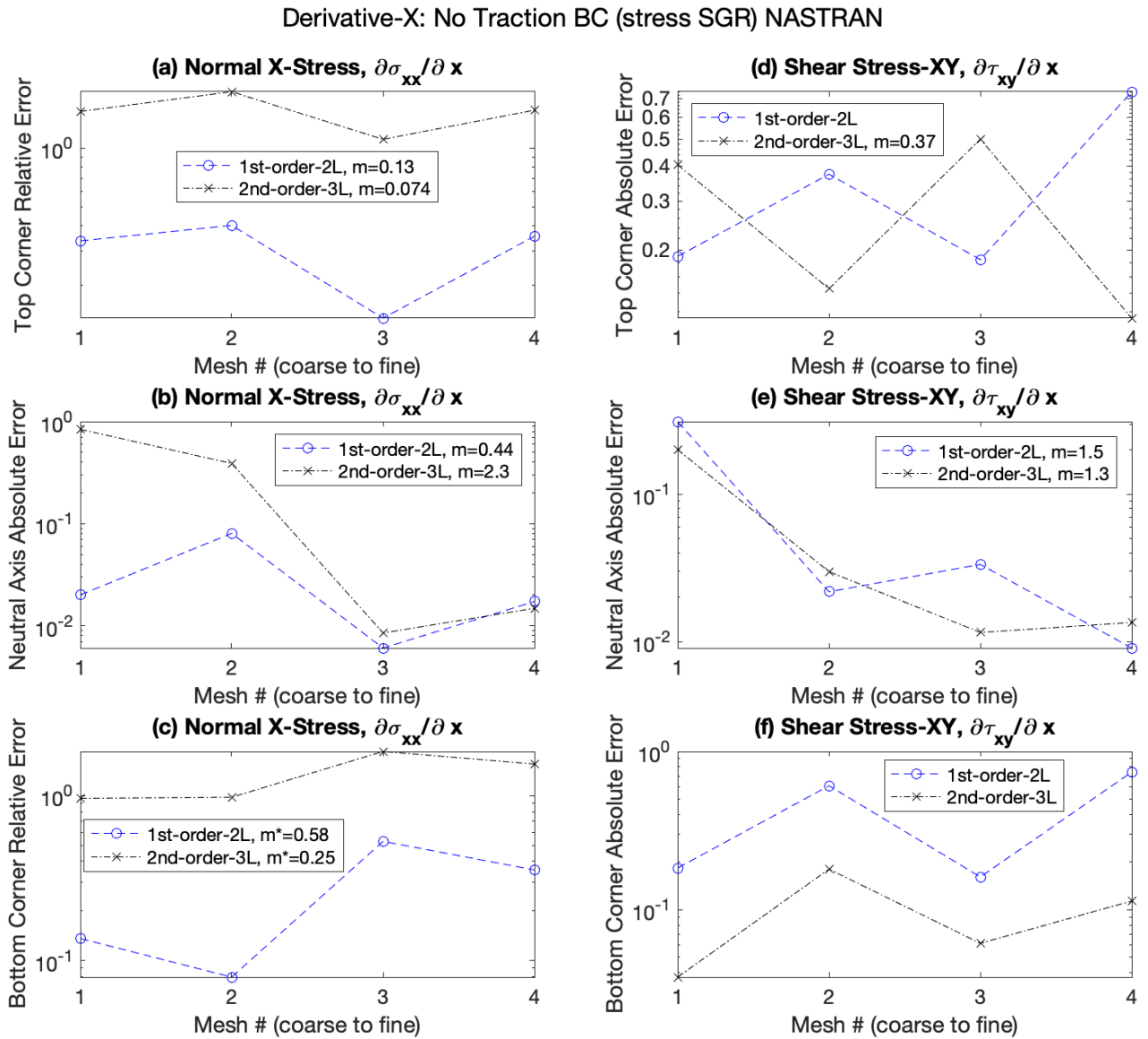


Figure B.6: Stress-Based SGR without Traction Convergence of X-derivatives at Tip (NASTRAN TRIA3)

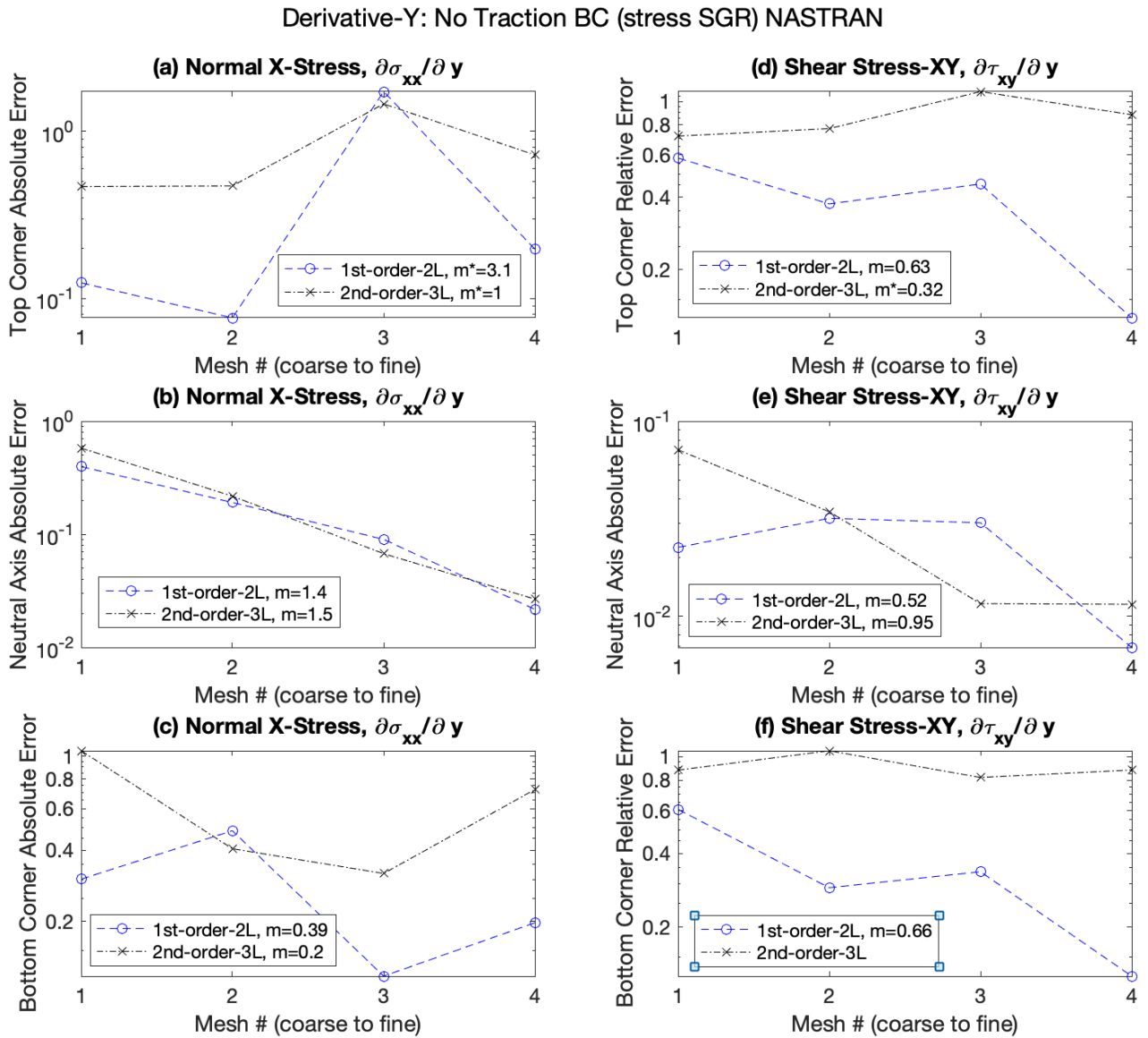


Figure B.7: Stress-Based SGR without Traction Convergence of Y-derivatives at Tip (NASTRAN TRIA3)

B.3 Convergence Results for Stress-Based SGR with Traction Enforced

Tip Stress - Enforced Traction BC (stress SGR) NASTRAN

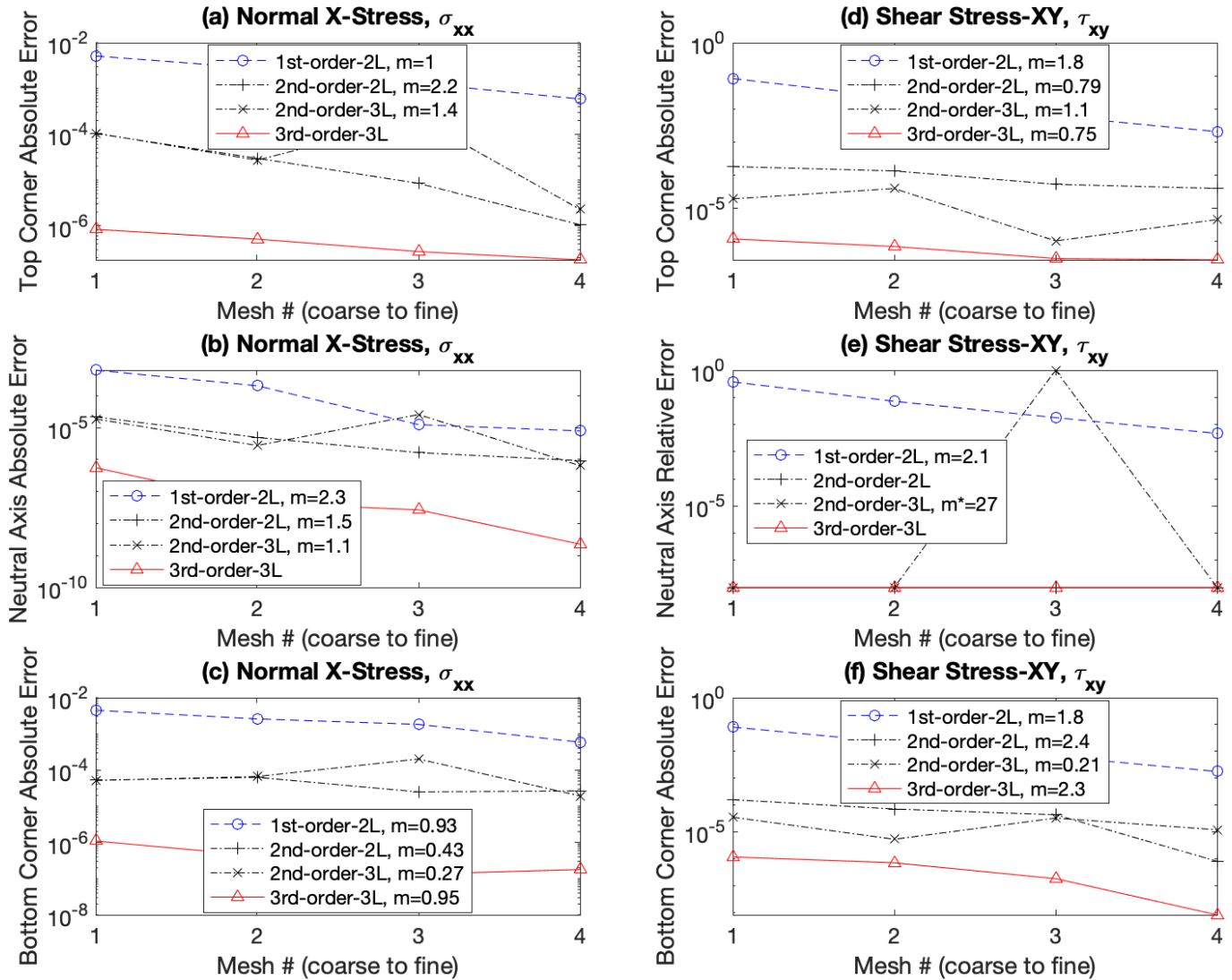


Figure B.8: Traction-Enforced Stress-Based SGR Convergence at Tip (NASTRAN TRIA3)

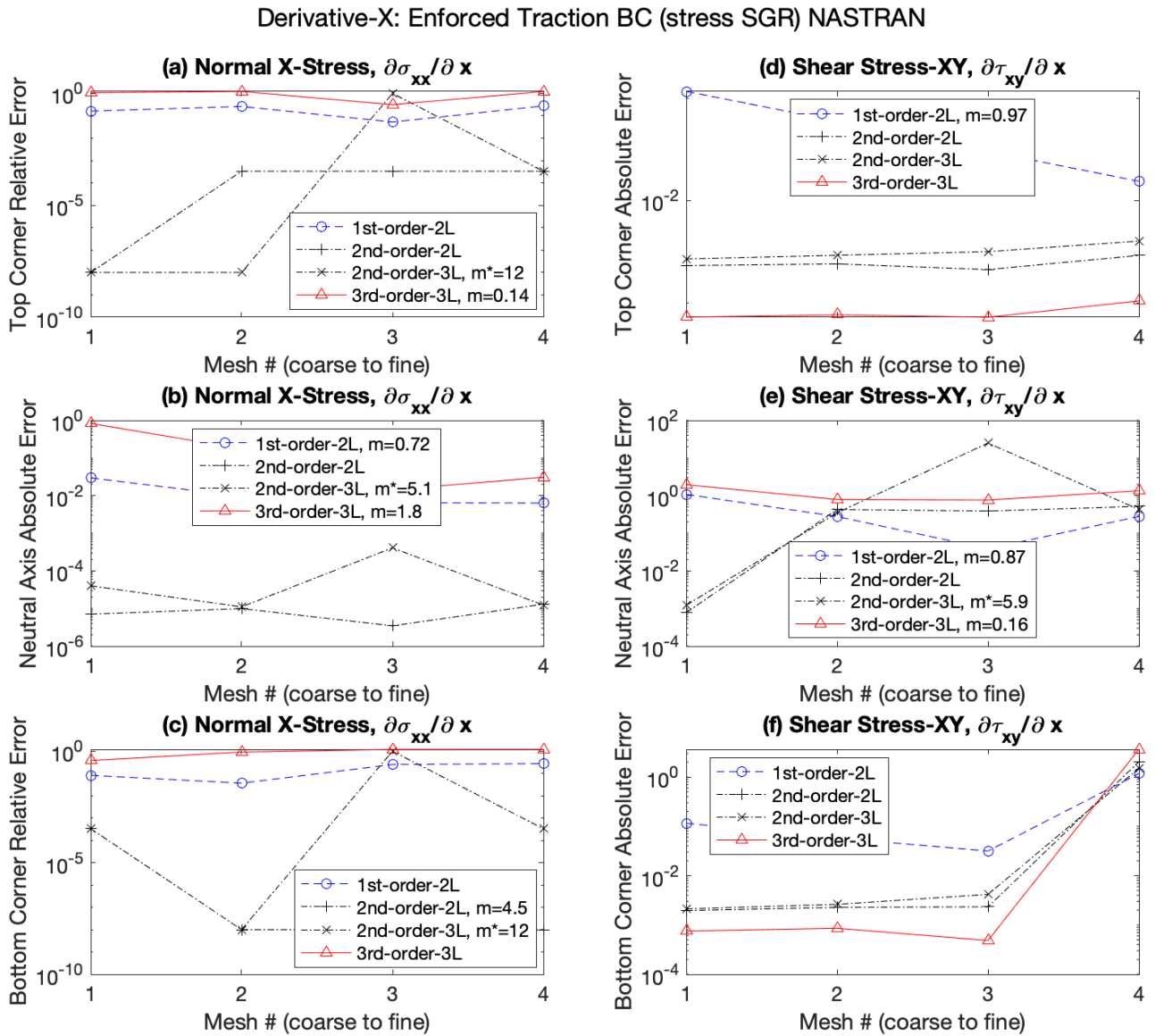


Figure B.9: Traction-Enforced Stress-Based SGR Convergence of X-derivatives at Tip (NASTRAN TRIA3)

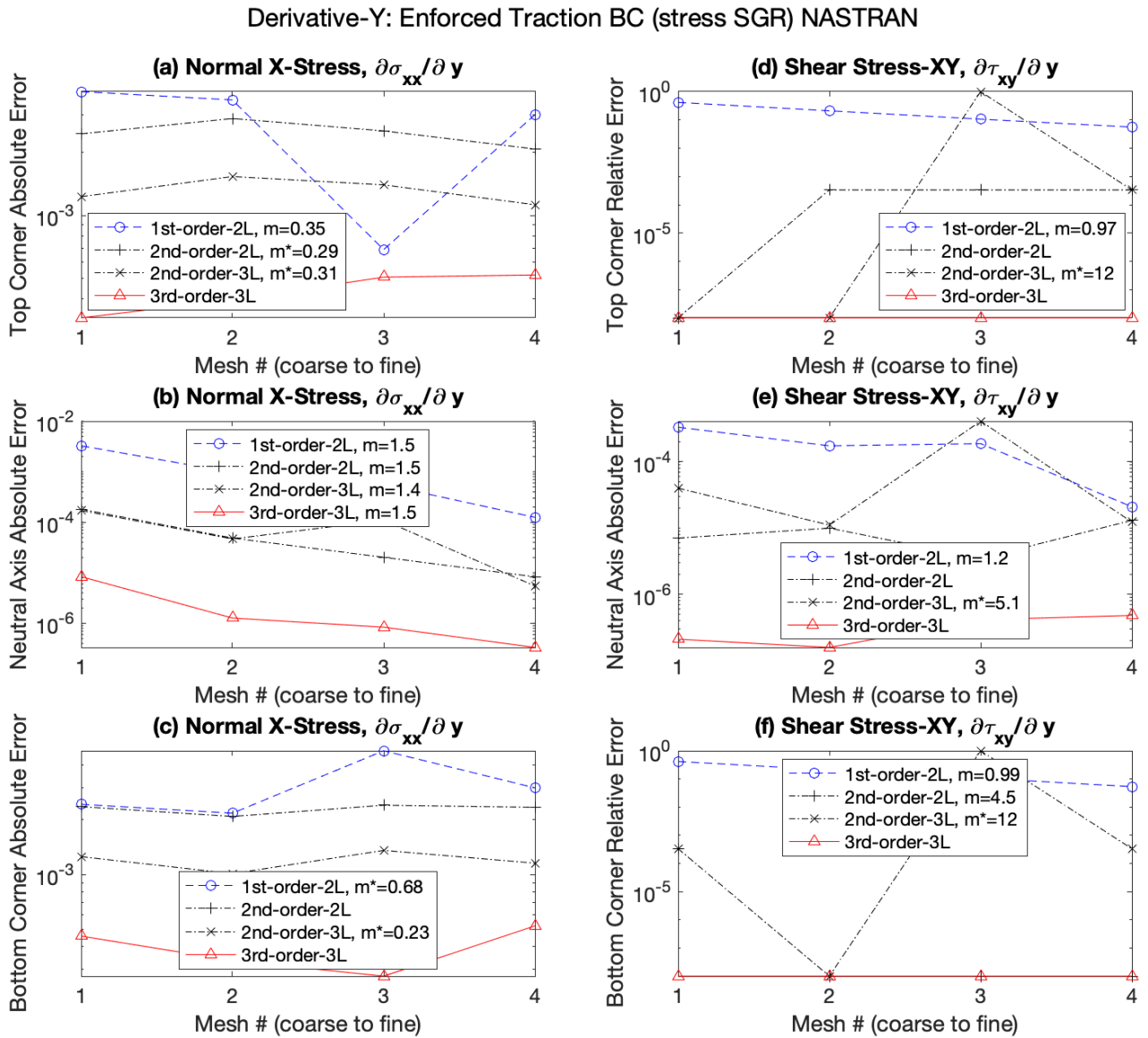


Figure B.10: Traction-Enforced Stress-Based SGR Convergence of Y-derivatives at Tip (NASTRAN TRIA3)

REFERENCES

- [1] K. Dems and Z. Mroz. Variational approach by means of adjoint systems to structural optimization and sensitivity analysis. part ii - structure shape variation. *International Journal of Solids and Structures*, 20:527–552, 1984.
- [2] K. Dems and R. T. Haftka. Two approaches to sensitivity analysis for shape variation of structures. *Mech. Struct. and Mach.*, 16(4):501, 1988.
- [3] J. S. Arora. An exposition of the material derivatives approach for structural shape sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 105: 41–62, 1993.
- [4] K. K. Choi and N.-H. Kim. *Structural sensitivity analysis and optimization*. Springer Science+Business media, New York, 2005.
- [5] J. Borggaard and J. Burns. A pde sensitivity equation method for optimal aerodynamic design. *Journal of Computational Physics.*, 136:366–384, 1997.
- [6] L. G. D. Stanley and D. Stewart. *Design sensitivity analysis : computational issues of sensitivity equation methods*. Academic Press, Philadelphia, 2002.
- [7] R. Duvigneau and D. Pelletier. On accurate boundary conditions for a shape sensitivity equation method. *International Journal for Numerical Methods in Fluids.*, 50 (2):147–164, 2006.
- [8] M. D. Kulkarni, R. A. Canfield, and M. J. Patil. Nonintrusive continuum sensitivity analysis for aerodynamic shape optimization. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2014-2043 in AIAA, Atlanta, GA, June 2014. AIAA. doi:10.2514/6.2014-2043. URL <http://dx.doi.org/10.2514/6.2014-2043>.
- [9] S. Liu and R. A. Canfield. Boundary velocity method for continuum shape sensitivity of nonlinear fluid structure interaction problems. *Journal of Fluids and Structures*, 40:284 – 301, 2013. ISSN 0889-9746. doi:<http://dx.doi.org/10.1016/j.jfluidstructs.2013.05.003>. URL <http://www.sciencedirect.com/science/article/pii/S0889974613001199>.
- [10] D. M. Cross and R. A. Canfield. Local continuum shape sensitivity with spatial gradient reconstruction. *Structural and Multidisciplinary Optimization*, 50(6):975–1000, December 2014.
- [11] D. M. Cross and R. A. Canfield. Local continuum shape sensitivity with spatial gradient reconstruction for nonlinear analysis. *Structural and Multidisciplinary Optimization*, 51(4):849–865, 2015. ISSN 1615-1488. doi:10.1007/s00158-014-1178-8. URL <http://dx.doi.org/10.1007/s00158-014-1178-8>.
- [12] D. M. Cross and R. A. Canfield. Convergence study of local continuum sensitivity method using spatial gradient reconstruction. *AIAA Journal*, 2016. doi:10.2514/1.J053800. URL <http://dx.doi.org/10.2514/1.J053800>.

-
- [13] M. D. Kulkarni, D. M. Cross, and R. A. Canfield. Discrete adjoint formulation for continuum sensitivity analysis. *AIAA Journal*, 2016. doi:10.2514/1.J053827. URL <http://arc.aiaa.org/doi/10.2514/1.J053827>.
- [14] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery (spr) and adaptive finite element refinement. *Computer Methods in Applied Mechanics and Engineering*, 101(1):207–224, 1992.
- [15] N.-E. Wiberg, F. Abdulwahab, and S. Ziukas. Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions. *International Journal for Numerical Methods in Engineering*, 37(20):3417–3440, 1994.
- [16] I. Babuška, T. Strouboulis, and C. S. Upadhyay. A model study of the quality of a posteriori error estimators for finite element solutions of linear elliptic problems, with particular reference to the behavior near the boundary. *International Journal for Numerical Methods in Engineering*, 40(14):2521–2577, 1997.
- [17] P. A. Sackinger, P. R. Schunk, and R. R. Rao. A newton-raphson pseudo-solid domain mapping technique for free and moving boundary problems. *J. Comput. Phys.*, 125(1):83–103, 1996. doi:10.1006/jcph.1996.0081.
- [18] S. Liu and R. A. Canfield. Two forms of continuum shape sensitivity method for fluid structure interaction problems. *Journal of Fluids and Structures*, 62:46 – 64, 2016. ISSN 0889-9746. doi:<http://dx.doi.org/10.1016/j.jfluidstructs.2015.12.013>. URL <http://www.sciencedirect.com/science/article/pii/S0889974616000104>.
- [19] J. Borggaard and J. Burns. A sensitivity equation approach to shape optimization in fluid flows. Nasa contractor report 191598, Langley Research Center, 1994.
- [20] T. Kvamsdal and M. Okstad. Error estimation based on superconvergent patch recovery using statically admissible stress fields. *International Journal for Numerical Methods in Engineering*, 42:443–472, 1998.
- [21] K. M. Okstad, T. Kvamsdal, and K. M. Mathisen. Superconvergent patch recovery for plate problems using statically admissible stress resultant fields. *International Journal for Numerical Methods in Engineering*, 44(5):697–727, 1999. doi:[https://doi.org/10.1002/\(SICI\)1097-0207\(19990220\)44:5<697::AID-NME526>3.0.CO;2-L](https://doi.org/10.1002/(SICI)1097-0207(19990220)44:5<697::AID-NME526>3.0.CO;2-L). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-0207%2819990220%2944%3A5%3C697%3A%3AAID-NME526%3E3.0.CO%3B2-L>.
- [22] R. A. Canfield, R. J. Durscher, D. E. Bryson, and R. M. Kolonay. Shape continuum sensitivity analysis using astros and caps. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-2228.
- [23] S. Timoshenko and J. Goodier. *Theory of Elasticity*. McGraw-Hill, 1951.
- [24] C. E. Augarde and A. J. Deeks. The use of timoshenko’s exact solution for a cantilever beam in adaptive analysis. *Finite Elements in Analysis and Design*, 44(9?10):595–601, 2008.

- [25] C. L. Dym and I. H. Shames. *Solid Mechanics: A Variational Approach, Augmented Edition*. Springer, 2013. ISBN 978-1-4614-6034-3.



FROM MDO TO MANUFACTURING: APPLICATION CASE FOR UNMANNED AERIAL VEHICLES

Luiz F. T. Fernandez^{1,2*}, Murat Bronz², Thierry Lefebvre¹ and Nathalie Bartoli¹

1: ONERA/DTIS,
Université de Toulouse,
31400 Toulouse, France
{luiz.tiberio,thierry.lefebvre,nathalie.bartoli}@onera.fr

2: ENAC,
Université de Toulouse,
31400 Toulouse, France
murat.bronz@enac.fr

Abstract. *This paper explores the concept of coupling an MDO problem directly to a manufacturing process. The design parameters required for various disciplinary analyses are used to automatically generate geometries that comply with the selected manufacturing processes. Such geometries are then manufactured to validate our proposal. This strategy reduces the manufacturing time of unmanned aerial vehicles (UAVs) as the CAD modeling becomes automatic and parametric, also allowing for easier comparison of different geometries. It can also potentially improve the MDO process by closing the information loop with manufacturing constraints. We employ the Engineering Sketch Pad (ESP) and leverage from additive manufacturing techniques usually applied in UAVs. The choice of ESP facilitates reproducibility, as it is an operating system agnostic open source CAD, and gradient based optimization, due to its capability of computing gradients of the geometric outputs with respect to the design parameters. The manufacturing processes of wings and propellers are addressed. For the same representative wing geometry, we present different modeling strategies for mass and inertia prediction and for manufacturing using 3D printing. ESP is also employed to predict wing mass and inertia. The maximum difference in weight between the manufactured wing and its predicted value using ESP is roughly 7%. The propeller is 3D printed in resin with stereolithography (SLA) technique, and is defined by means of its airfoils, radius, and chord and twist angle distributions. We conclude about the applicability of the presented strategy, as well as its potential and limitations. All the scripts are shared with the community so researchers can apply it to validate their own MDO problem.*

Keywords: Additive Manufacturing, MDO of UAVs, UAV design and manufacturing

1 INTRODUCTION

With the establishment of unmanned aerial vehicles (UAVs), a significant increase in the interest in such kind of vehicle has been observed in recent years for several applications. Ranging from widespread ideas such as delivery of goods or search and rescue to more specialized ones, such as fire fighting [1–3], debris mapping [4], covid mitigation [5], and even nutrition delivery in the form of edible vehicles [6]. Within different configurations of UAVs, winged designs stand out when higher flight range and endurance are needed. Several manufacturing techniques have been applied for winged UAVs fabrication. In terms of wing manufacturing, hobbyists predominantly utilize balsa wood, which was primarily employed during the early stages of the UAV era. This conventional approach has subsequently been enhanced by combining balsa with composite materials, as in Chung *et al.* [7], and even full composite wings as in Nugroho *et al.* [8]. Full composite wings tend to be light and of high quality, but the manufacturing process is complicated and time consuming. Recently, there has been attention towards additive manufacturing techniques due to their ability to save time and material, fabricate complex geometries, and achieve higher levels of automation. Such techniques have been successfully applied in several cases. Bronz *et al.* [9] presented a 3D printing strategy for UAVs using Onyx material, a blend of carbon fiber and nylon, which provided the desired structural properties and surface quality while keeping a reasonable weight. Taylor *et al.* [10] and Muir *et al.* [11] coupled optimization processes to additive manufacturing techniques in order to obtain an aircraft structure and a mission optimized UAV, respectively. Laliberté *et al.* [12] used 3D printing to build a biologically inspired micro aerial vehicle. In 2015, Aurora Flight Sciences also presented the first jet-powered 3D-printed UAV [13]. While several other applications of additive manufacturing for UAVs can be found in the survey by Goh *et al.* [14], this technique has also been successfully employed for teaching activities [15] and even in experiments in hypersonic conditions [16]. For propellers, there are several off-the-shelf options commercially available. Their good quality combined with small prices is usually enough for the UAV community, which has not yet explored propeller design and manufacturing with the exception of a few studies, as in Rutkay and Laliberté [17]. We already employed additive manufacturing for propellers in [18], where a specific geometry was needed in order to compare different aerodynamic methodologies. The higher level of automation from 3D printing also facilitates the integration between design and fabrication processes. Such integration might be beneficial in two ways: enriching design and optimization process with manufacturing constraints, anticipating possible problems or even unfeasible designs; and reducing the time needed to complete the full cycle between design and flight. Regarding the insertion of manufacturing information within a multidisciplinary design and optimization (MDO) process, Laan *et al.* [19] used model-based system engineering techniques to account for the manufacturing cost of high lift devices in the design of experiments performed for the initial sizing of an aircraft. Doneli *et al.* [20, 21] also addressed, within the MDO, the choice of raw materials, manufacturing and assembly processes as well as the supply chain, ensuring a more realistic and comprehensive optimization process. In this paper, our focus is to present a viable way to automatically connect the output of a given MDO process to the fabrication of a UAV. The optimization loop is represented by means of its output design parameters or geometry, and the fabrication is performed with 3D printing. We leverage from the size and affordable characteristics of UAVs to fully test our proposal, presenting the entire

Table 1: Optimal design from [22]

Design variables	Value
Root chord	0.12 <i>m</i>
Tip chord	0.095 <i>m</i>
Wing span	0.7 <i>m</i>
Cruise α	10°
Turn α	8°
Battery mass	0.2 <i>kg</i>



Figure 1: 3D model and real vehicle from [22]

process, from modeling to real manufacturing, for wings and propellers. This paper is organized as follows: in Section 2 we present the context in which this study takes place. Section 3 explains how we share the data. The methodology and results are presented in Sections 4 and 5, respectively. Our conclusions and perspectives are then presented in Section 6.

2 CONTEXT

This paper is part of a research line dedicated to the application of MDO strategies to small vertical takeoff and landing (VTOL) vehicles. Apart from being useful for different applications, such vehicles also offer the possibility of experimentally validating design strategies and decisions due to their low cost and easier manufacturing and flight testing (compared to bigger vehicles) procedures. On the other hand, manufacturing the vehicle is usually a time consuming task that includes not only the fabrication itself but also the generation of 3D models, needed to guide the integration of all the systems. Building a good model often requires experience to avoid several iterations and minimize cost. In [22], we developed a simplified MDO problem to design a vehicle for the 2022 International Micro Aerial Vehicle Competition. The outcome of this problem was a set of parameters, shown in Tab.1. In order to "transform" such numbers into a real vehicle, we manually built the 3D model shown in Fig.1. Such process consumed significantly more time than running the MDO. Similarly, in [18] we fabricated propellers to compare different aerodynamic techniques for evaluating their performance. After designing one by defining its airfoils, twist and chord distributions, a manual 3D model was also needed to design the hub and integrate it with the blades in order to have a usable propeller. This papers addresses this gap between the output of a given MDO problem and the generation of a manufacturing compliant 3D model, allowing for time and cost reduction

Table 2: Wing section parameters.

Parameter	Unit
Chord	[m]
Position in X , Y , and Z	[m]
Twist angle	[deg]
Airfoil location of max camber	[percent chord]
Airfoil maximum thickness	[percent chord]
Airfoil maximum camber	[percent chord]

and permitting less experienced designers to build models more easily.

3 DATA SHARING

In the subsequent sections, we outline the methodology utilized and present the results obtained. All files, ESP geometries and Python scripts, are shared so the community can use and improve the methods. They are available in *ENAC drones* group *GitHub* at https://github.com/enac-drones/AeroBest23_MDO_UAV/tree/paper_codes/. We provide two examples for wings manufacturing, one simpler without twist and using a NACA airfoil (under *Wo_twist* folder) and other with twist and Kulfan airfoils (under *With_twist* folder). The scripts using ESP's Python interface for wing weight estimation are also provided. The Python scripts for blade generation and ESP files for propellers are available under the folder *propellers*.

4 METHODOLOGY

Instead of defining an MDO process, we only represent the output of such kind of problem, ensuring that the same strategy can be used for any process with a similar parametrization. As explained by Hajdik *et al.* [23], despite the wide application of free form deformation (FFD) in MDO, as in [24–26], coupling FFD geometries in general design and manufacturing workflows is not straightforward, as information might be lost throughout this process. So, a CAD tool is employed, specifically the *Engineering Sketch Pad* (ESP) [27], an open source and operating system agnostic software. ESP incorporates drivers for aerodynamic and structural tools. Additionally, it is compatible with OpenM-DAO [28] and natively allows for gradient calculation. In order to test our proposal for wings, we select a representative set of design parameters and present the different modeling strategies needed for manufacturing and for mass/inertia estimation. For propellers, the manufacturing process is simpler and relies on *BladeX* by Gadalla *et al.* [29].

4.1 Wing and propeller parametrization

Each wing section is defined by the parameters shown in Tab.2, where the number of sections can also change. ESP has primitives for different airfoil families, but we employ NACA family for simplicity. The structural parameters, shown in Tab.3, are defined for the whole wing. Spars are assumed to be cylindrical as off-the-shelf carbon tubes are usually employed in this form for small UAVs. The rib offset is the angle between the ribs and the chord line. For propeller parametrization we employ three parameters: airfoil(s), chords and twist distributions along the radius.

Table 3: Wing structural parameters.

Parameter	Unit
Number of ribs	-
Thickness of the ribs	[m]
Ribs offset	[deg]
Number of spars	-
Diameter of each spar	[m]
Location of each spar	[percent chord]

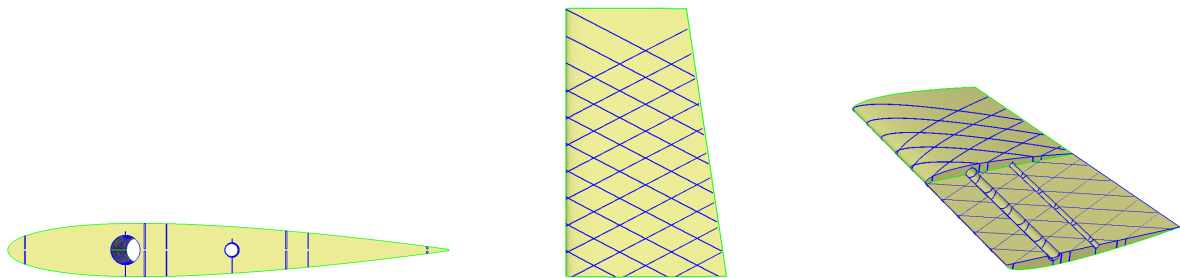


Figure 2: Modeling the wing to comply with the manufacturing technique: solid body with empty internal ribs (in detail on the right) and spars.

4.2 Manufacturing setup

The wings are manufactured with a *Raise3D N2 Pro* by *Raise 3D*¹. We use the lightweight PLA from *colorFabb*² for all tests. Once the geometry has been defined, as an *STL* file for our case, it needs to be "sliced" in order to be printed. There are several open source slicers. We use the *IdeaMaker*³ due to its compatibility with our printers. For the propeller, the 3D printer *Form 3+*, from *FORMLABS*⁴ is employed. The material used is the "Tough" resin, provided by the same company. To prepare the printing process we use the *PreForm*⁵, also open source.

4.3 Modeling wings for manufacturing

The 3D printing process is based on the one presented in [30] and has already been used in [22]. This fabrication strategy ensures a more continuous printing, minimizing the need to remove the nozzle throughout the process, which tends to increase the weight of the piece and reduce its quality. It roughly consists in modeling the wing as a solid body while keeping the ribs as empty cavities. Figure 2 shows the obtained model. *ESP* is capable of directly outputting the resulting geometry as an *STL* file, which is then used to obtain the *G code* needed to print the part. Manufacturing wings with high twist angles and cambered airfoils might require some adaptations to the original method in order to

¹www.raise3d.com

²www.colorfabb.com

³www.ideamaker.io

⁴www.formlabs.com/eu/3d-printers/form-3

⁵www.formlabs.com/eu/software/#preform

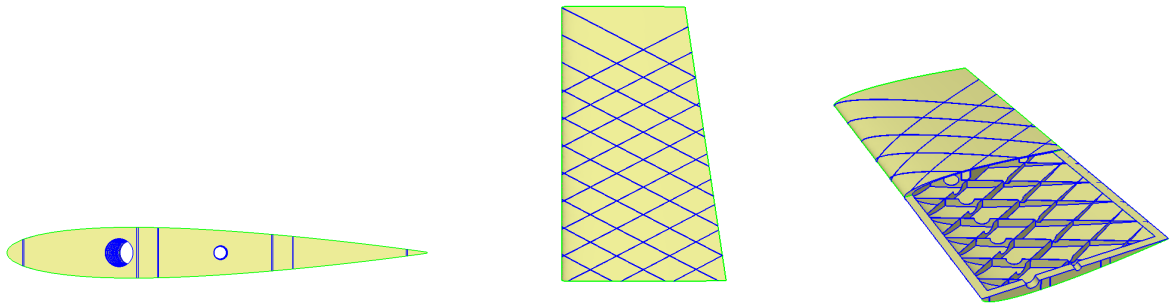


Figure 3: Modeling the real printed wing for mass and inertia estimation, with internal ribs being detailed on the right.

Table 4: *Masstran* material definition.

Parameter	value
Material	Lighweight PLA
Material density	390 [$\frac{kg}{m^3}$]
Material type	Isotropic shell
Membrane thickness	0.74[mm]

correctly generate the ribs. An example for such case is also available for comparison.

4.4 Modeling wings for mass and inertia estimation

Understanding the modelization strategy for mass and inertia estimation is straightforward. The CAD model needs to represent the final manufactured body as accurately as possible, as seen in Fig.3. With such model and the material definition, shown in Tab.4, we calculate mass and inertia with the *Masstran* feature present in *PyCaps* [31]. The membrane thickness was obtained after averaging different printed pieces.

4.5 Modeling propellers

The propeller process starts with the definition of one blade using BladeX [29]. The blade is defined by means of its airfoils and chord and twist distributions. Once the blade is generated, it is exported as an *IGES* file. The ESP script then imports this file to generate both blades. It also models the hub and exports the propeller as an *STL* file to be used for printing.

5 RESULTS

In the following sections we present the results obtained with the manufacturing processes as well as some aspects regarding their validation. We present the comparison between predicted and measured weight for wings and discuss the time needed to obtain the printed pieces.

5.1 Wings

Figure 4 shows two printed wings, different from each other because of the airfoils, geometric twist and number of ribs, and also a zoom in the airfoil and wing internal



(a) Printed wings with and without twist.



(b) Airfoil and internal structure.

Figure 4: Printed wings.



(a) Twisted wing with 3 ribs per side.



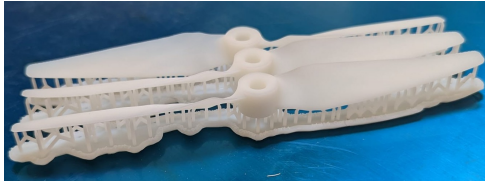
(b) Straight wing with 10 ribs per side.

Figure 5: Wing mass measurements.

structure. The geometry of the wings can for now be validated by means of measuring span and chord. A further analysis with 3D scanners is left for future work. However, any mismatch between the printed wing and the one defined during the optimization is most likely caused by the printing itself, as the geometry generated with ESP matches the one intended. We also compare the expected and observed mass using *Masstran* from *PyCaps* [31]. Each wing mass was measured and the results are shown in Fig.5. Table 5 shows the difference between predicted and measured mass. The accuracy of the outcome is highly dependent on the quality of the mass models. For more complex wings with, for example, command surfaces, servos, and hinges, the prediction might lose precision, depending on how such geometry is modeled. For our case, the simple wing geometry is

Table 5: Mass prediction results.

Number of ribs	Masstran result [g]	Measured mass [g]	Error [%]
3	25.30	26.1±0.1	3.16
10	31.53	29.5±0.1	6.88



(a) Printed propellers straight out of the printer.



(b) After the cleaning and "sanding" process.

Figure 6: Printed propellers.

based on our vehicle Falcon [22] shown in Fig.1, a UAV with four motors and no command surfaces. As *Masstran* also is capable of calculating inertia, this tool can be very useful for flight dynamics analysis. However, any validation with this regard is left for future work. The CAD model and "manufacturing view" are automatically generated once the geometry is known. So the total time to obtain the piece is driven by the manual slicing process, which is straightforward, and the printing time. The printing time depends highly on the geometry, printer, and material. For both wings manufactured for this paper, the process took roughly 4 hours each considering the setup presented in Section 4.2. This result shows a potential of reducing the manufacturing time in days where the final goal is to have only the printing time as constraint. However, the real reduction is hard to estimate because it highly depends on the designer's ability to generate the model and on the complexity of the geometry.

5.2 Propellers

Figure 6 shows the obtained propellers. The printer generates a supporting structure to ensure quality, as shown in Fig.6a. This structure needs to be removed and the propeller cleaned, as shown in Fig.6b. We can also validate propellers by measuring the chords and twist angles. The 3D scan technology is even more important for them as it is difficult to measure the angles properly. However, the similar performance in experiments and analytical methods observed in [18] indicates that the propellers are close to the original geometry definition.

6 CONCLUSION

In this paper, we present a preliminary way of integrating the manufacturing process of wings and propellers to an MDO framework for UAVs. We employ ESP to automatically generate the fabrication and mass prediction geometric "views". Such geometries are then used to manufacture two sample wings and assess their predicted weight. The weight prediction shows good accuracy for our case, mainly because of the simplicity of the wings analyzed. We also present a similar strategy for propeller manufacturing. Such strategy has also been used in [18] and the performance obtained in wind tunnel tests was close to the one expected. It is expected that this solution can improve MDO processes by providing new manufacturing constraints and anticipating potential issues. It can also reduce the time needed to build test vehicles to perform flight and/or wind tunnel test campaigns, thus accelerating the design process. Every script used for this paper is open source so the community can use the methods to build UAVs and subscale vehicles to improve MDO processes.

ACKNOWLEDGEMENTS

This work is part of the activities of ONERA-ISAE-ENAC joint research group.

REFERENCES

- [1] E. Ausonio, P. Bagnerini, and M. Ghio. Drone swarms in fire suppression activities: A conceptual framework. *Drones*, 5(1), 2021. ISSN 2504-446X. doi:10.3390/drones5010017. URL <https://www.mdpi.com/2504-446X/5/1/17>.
- [2] B. Aydin, E. Selvi, J. Tao, and M. J. Starek. Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting. *Drones*, 3(1), 2019. ISSN 2504-446X. doi:10.3390/drones3010017. URL <https://www.mdpi.com/2504-446X/3/1/17>.
- [3] M. A. Akhloufi, A. Couturier, and N. A. Castro. Unmanned aerial vehicles for wild-land fires: Sensing, perception, cooperation and assistance. *Drones*, 5(1), 2021. ISSN 2504-446X. doi:10.3390/drones5010015. URL <https://www.mdpi.com/2504-446X/5/1/15>.
- [4] Y. Taddia, C. Corbau, J. Buoninsegni, U. Simeoni, and A. Pellegrinelli. Uav approach for detecting plastic marine debris on the beach: A case study in the po river delta (italy). *Drones*, 5(4), 2021. ISSN 2504-446X. doi:10.3390/drones5040140. URL <https://www.mdpi.com/2504-446X/5/4/140>.
- [5] A. Restás. Drone applications fighting covid-19 pandemic;towards good practices. *Drones*, 6(1), 2022. ISSN 2504-446X. doi:10.3390/drones6010015. URL <https://www.mdpi.com/2504-446X/6/1/15>.
- [6] B. Kwak, J. Shintake, L. Zhang, and D. Floreano. Towards edible drones for rescue missions: design and flight of nutritional wings. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1802–1809, 2022. doi:10.1109/IROS47612.2022.9981956.
- [7] P.-H. Chung, D.-M. Ma, and J.-K. Shiau. Design, manufacturing, and flight testing of an experimental flying wing uav. *Applied Sciences*, 9(15), 2019. ISSN 2076-3417. doi:10.3390/app9153043. URL <https://www.mdpi.com/2076-3417/9/15/3043>.
- [8] G. Nugroho, A. A. Rafsan Jani, R. R. Trio Sadewo, and M. Satrio. Manufacturing process and flight testing of an unmanned aerial vehicle (uav) with composite material. In *Materials and Technologies in Modern Mechanical Engineering*, volume 842 of *Applied Mechanics and Materials*, pages 311–318. Trans Tech Publications Ltd, 9 2016. doi:10.4028/www.scientific.net/AMM.842.311.
- [9] *Mission-Oriented Additive Manufacturing of Modular Mini-UAVs*, 2020. doi:10.2514/6.2020-0064. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2020-0064>.
- [10] *Design Optimization, Fabrication, and Testing of a 3D Printed Aircraft Structure Using Fused Deposition Modeling*, 2020. doi:10.2514/6.2020-1924. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2020-1924>.

- [11] *The Use of MDO and Advanced Manufacturing to Demonstrate Rapid, Agile Construction of a Mission Optimized UAV*, 2013. doi:10.2514/6.2013-1675. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2013-1675>.
- [12] J. F. Laliberté, K. L. Kraemer, J. W. Dawson, and D. Miyata. Design and manufacturing of biologically inspired micro aerial vehicle wings using rapid prototyping. *International Journal of Micro Air Vehicles*, 5(1):15–38, 2013. doi:10.1260/1756-8293.5.1.15. URL <https://doi.org/10.1260/1756-8293.5.1.15>.
- [13] N. Atlas. World’s first jet-powered, 3d-printed uav debuts at dubai airshow, 2015. URL <https://newatlas.com/worlds-largest-fastest-3d-printed-uav/40293/>.
- [14] G. Goh, S. Agarwala, G. Goh, V. Dikshit, S. Sing, and W. Yeong. Additive manufacturing in unmanned aerial vehicles (uavs): Challenges and potential. *Aerospace Science and Technology*, 63:140–151, 2017. ISSN 1270-9638. doi:<https://doi.org/10.1016/j.ast.2016.12.019>. URL <https://www.sciencedirect.com/science/article/pii/S127096381630503X>.
- [15] *Aero-engine model design and assembly practice teaching based on 3D printing*, 2023. doi:10.2514/6.2023-1020. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2023-1020>.
- [16] I. Rêgo, T. Marcos, D. R. Pinto, R. Vilela, V. Galvão, A. Pivetta, G. Camilo, J. Silva, B. Lima, A. Carvalhal, R. Cardoso, J. Martos, A. Santos, A. Oliveira, and P. Toro. Ground experimentation with 3d printed scramjet inlet models at hypervelocities. *Aerospace Science and Technology*, 55:307–313, 2016. ISSN 1270-9638. doi:<https://doi.org/10.1016/j.ast.2016.06.009>. URL <https://www.sciencedirect.com/science/article/pii/S1270963816302188>.
- [17] B. Rutkay and J. Laliberté. Design and manufacture of propellers for small unmanned aerial vehicles. *Journal of Unmanned Vehicle Systems*, 4(4):228–245, 2016. doi:10.1139/juvs-2014-0019. URL <https://doi.org/10.1139/juvs-2014-0019>.
- [18] L. F. Fernandez, M. Bronz, N. Bartoli, and T. Lefebvre. Assessment of methods for propeller performance calculation at high incidence angles. In *AIAA SCITECH 2023 Forum*, 2023. doi:10.2514/6.2023-2283. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2023-2283>.
- [19] *Bringing Manufacturing into the MDO domain using MBSE*, 2022. doi:10.2514/6.2022-3721. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-3721>.
- [20] G. Donelli, P. D. Ciampa, B. Nagel, G. Lemos, J. Mello, A. P. C. Cuco, and T. van der Laan. A model-based approach to trade-space evaluation coupling design-manufacturing-supply chain in the early stages of aircraft development. In *AIAA AVIATION 2021 FORUM*, 2021. doi:10.2514/6.2021-3057. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2021-3057>.

- [21] G. Donelli, P. D. Ciampa, T. Lefebvre, N. Bartoli, J. M. Mello, F. I. Odaguil, and T. van der Laan. Value-driven model-based optimization coupling design-manufacturing-supply chain in the early stages of aircraft development: Strategy and preliminary results. In *AIAA AVIATION 2022 Forum*, 2022. doi:10.2514/6.2022-3723. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2022-3723>.
- [22] L. F. T. Fernandez, M. Bronz, N. Bartoli, and T. Lefebvre. Development of a mission-tailored tail-sitter mav. In *13th International Micro Air Vehicle Conference*, pages 159–168, Delft, the Netherlands, Sep 2022. URL <http://www.imavs.org/papers/2022/19.pdf>. Paper no. IMAV2022-19.
- [23] *Aerodynamic Shape Optimization with CAD-Based Geometric Parameterization*, 2023. doi:10.2514/6.2023-0726. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2023-0726>.
- [24] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, September 2010. AIAA 2010-9231.
- [25] Y. Shi, C. A. Mader, and J. R. R. A. Martins. Natural laminar flow wing optimization using a discrete adjoint approach. *Structural and Multidisciplinary Optimization*, 64(2):541–562, Aug 2021. ISSN 1615-1488. doi:10.1007/s00158-021-02936-w. URL <https://doi.org/10.1007/s00158-021-02936-w>.
- [26] N. P. Bons, J. R. R. A. Martins, F. I. K. Odaguil, and A. P. C. Cuco. Aerostructural Wing Optimization of a Regional Jet Considering Mission Fuel Burn. *ASME Open Journal of Engineering*, 1, 10 2022. ISSN 2770-3495. doi:10.1115/1.4055630. URL <https://doi.org/10.1115/1.4055630>. 011046.
- [27] J. Dannenhoffer and R. Haimes. The engineering sketch pad, 2022. URL <https://acdl.mit.edu/esp/>.
- [28] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi:10.1007/s00158-019-02211-z.
- [29] M. Gadalla, M. Tezzele, A. Mola, and G. Rozza. BladeX: Python Blade Morphing. *The Journal of Open Source Software*, 4(34):1203, 2019. doi:<https://doi.org/10.21105/joss.01203>.
- [30] T. Station. How i designed a 3d printed wing. URL https://www.youtube.com/watch?v=QJjhMan6T_E.
- [31] R. J. Durscher and D. Reedy. pycaps: A python interface to the computational aircraft prototype syntheses. In *AIAA Scitech 2019 Forum*, 2019. doi:10.2514/6.2019-2226. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2019-2226>.



SKY SAILING OF TETHERED AEROSTATS FOR EFFICIENT AERIAL MONITORING

Andrzej Świercz^{1*}, Cezary Graczykowski^{1*}, Lech Knap^{2*},
Zbigniew Wolejsza^{1*} and Jan Holnicki-Szulc^{1*}

1: Institute of Fundamental Technological Research
Polish Academy of Sciences
Pawinskiego 5B, 02-106 Warsaw, Poland
{aswiercz, cgraczyk, zwolejsza, holnicki}@ippt.pan.pl, <http://www.ippt.pan.pl>

2: Institute of Vehicles and Construction Machinery
Warsaw University of Technology
Narbutta 84, 02-524 Warsaw, Poland
Lech.Knap@pw.edu.edu, <http://www.simr.pw.edu.pl>

Abstract *This contribution introduces the concept of sky sailing, which combines the advantages of airships and standard fixed-wing aircraft, albeit in a vertical plane alignment. The proposed vehicle is equipped with rigid aerodynamic sails and auxiliary engines, enabling navigation and control with minimal power consumption along the desired trajectory. The proper orientation of the airship relative to the wind direction is achieved through the adjustment of the sails' angle of attack and the use of auxiliary lateral engines. Consequently, the system enables efficient maneuvering, particularly in windy conditions, while requiring low energy input. In the current stage of our research, we focus on 2D sky sailing in a horizontal plane. This study formulates mathematical model which employs a combined approach of analytical methods and numerical simulations based on finite volume method. Then, the corresponding control problem aimed at following the desired fly path with the lowest possible energetic cost. The motivation behind this work stems from the potential applications of aerial monitoring, such as crop or forest surveillance.*

Keywords: Airship, flight control, optimization, aerospace

1. INTRODUCTION

Balloons, aerostats and airships are widely applied as High Altitude Pseudo Satellites (HAPS) for environment and aerial monitoring as well as surveillance missions [1-2]. The rapid development of materials and electronic devices has influenced the advances in the field of airborne vehicles [3-5]. Many research works are devoted to solving problem of predicting the movement of airships in both the vertical and the horizontal direction [6-7], as well as to analysis of the changes in volume and corresponding buoyancy force resulting from heating of the envelope by the sun and heat exchange with the environment [8-9]. However, the problem of control of such vehicles remains difficult due to variable atmospheric conditions. Proposed by [10] solution to control the horizontal balloon drift across the wind involved the development of a balloon guidance system (BGS) and its in-flight tests. The idea of this system is based on application of an aerodynamic wing suspended beneath the balloon (1km or more) by means of a tether. The drag forces are controlled using winch system for lowering and raising the wing leading to stabilizing the balloon position. In another application, the software tools were developed focusing on station-keeping of high altitude balloon, exploiting the natural wind conditions [11]. By using altitude control system based on venting and pumping processes the efficient stabilization of the platform within designated district was achieved.

In the previous paper [12], the authors proposed construction of the aerostat based on deployable ultra-light rod-cable tensegrity structure integrated with envelope and equipped with elements of controllable lengths. Such construction enables convenient transport of the aerostat with the use of a balloon or an aircraft to the operational altitude and its automatic deployment at certain location at the atmosphere. On the other hand, the application of adaptable elements enables change of the aerostat volume and shape during the flight. In turn, the procedures for buoyancy control of a tensegrity-based aerostat were presented in [13]. The ultra-light rod-cable tensegrity structure instrumented with electronic devices was proved to enable precise control volume of the aerostat, the corresponding buoyancy force and resulting aerostat horizontal motion. The presented numerical examples have shown the possibility of accomplishing various flight missions by using volume and shape control.

The current contribution introduces the concept of *sky sailing*, which allows to combine the advantages of lighter-than-air vehicles and standard fixed-wing-aircrafts. According to the proposed concept, the aerostat is equipped with a rigid sail of aerodynamic profile and auxiliary engines of small power consumption. The engines provide desired orientation of the aerostat with respect to the wind direction, while the sail causes generation of additional aerodynamic forces influencing motion of the aerostat. As a result, the system enables efficient tacking in the wind and low-energetic motion of the aerostat. In the current stage of the research, for the sake of simplicity we consider a basic problem of sky sailing in the horizontal plane, where the vertical components of aerodynamic forces and buoyancy changes are neglected.

The remainder of the paper is organized as follows. The second section presents a mathematical model describing the *sky sailing* concept, which is based on equations of motion of the aerostat in horizontal plane and numerically determined changes of aerodynamic forces in terms of the system configuration and the wind velocity. The third section shows results of the conducted computational fluid dynamics (CFD) analyses including air velocity fields and aerodynamic forces resulting from the wind flow around the aerostat for various angles of attack. In the fourth section we formulate a control problem aimed at following the desired flight path with minimal energetic cost of the engines operation and analyse its basic features. Finally, in the fifth section general conclusions about modelling, control and advantages of the *sky sailing* concept are presented.

2. MATHEMATICAL MODEL OF THE SKY SAILING

The proposed concept of sky sailing effectively combines the advantages of airships and standard fixed-wing aircraft. The advantages of the airship are achieved by using lighter-than-air gas (helium or hydrogen) in order to obtain buoyancy force and stable position at the assumed altitude under certain atmospheric conditions. In addition, the proposed airship is equipped with rigid sails of aerodynamic profile aligned in the vertical plane (Fig. 1a, side view) and auxiliary lateral engines (Fig. 2b, horizontal view). The combination of the vertical sails and auxiliary lateral engines enables effective usage of aerodynamic forces and obtaining the effect of drifting in the wind in the direction deviated from the wind direction. The proper orientation of the airship relative to the wind direction and its energy saving motion in space are achieved via proper adjustment of the engines' power and sails' angle of attack. Consequently, the system enables efficient maneuvering and navigation, particularly in windy conditions, while requiring low energy input.

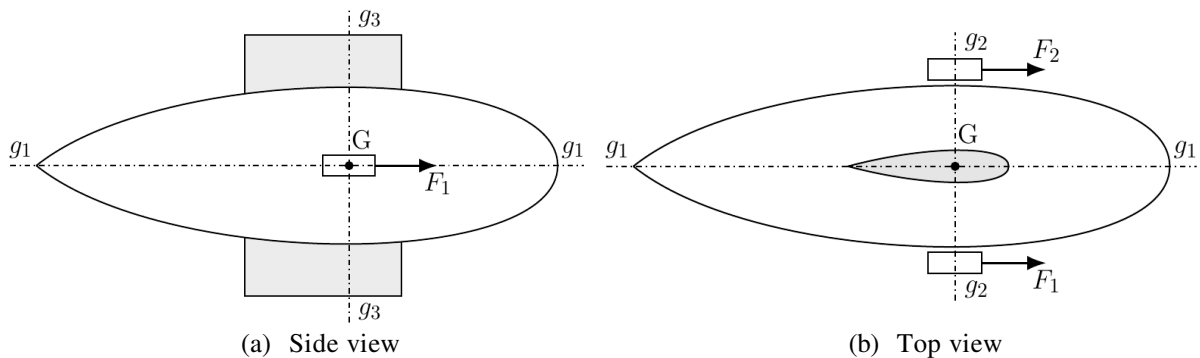


Figure 1. Airship model.

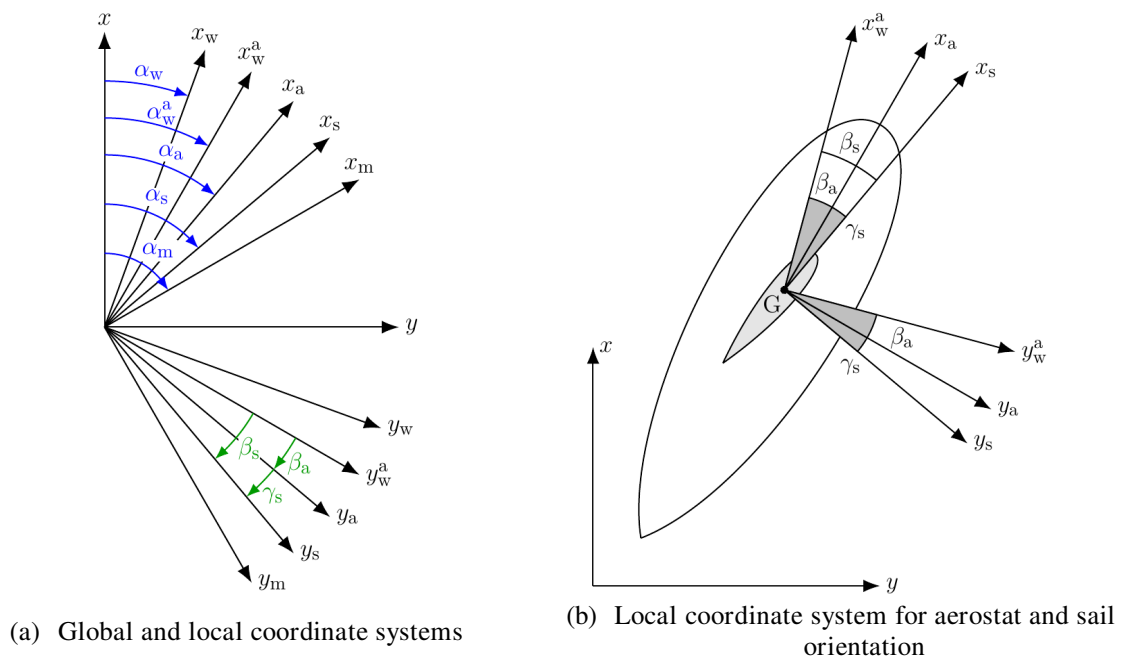


Figure 2. Coordinate systems using in mathematical model of the aerostat

The fundamental mathematical model of the *sky sailing* concept refers to the motion of the aerostat in the horizontal plane. In such a model we consider exclusively aerodynamic forces

acting in horizontal plane, while the variations of buoyancy and perturbation of vertical forces (e.g. due to temperature changes) are neglected. The starting point for development of the mathematical model is proper definition of the auxiliary coordinate systems and their orientation angles with respect to global coordinate system. In particular, Fig. 2a presents four angles denoted by α which are referred to global coordinate system, namely: α_w – the wind direction, α_a – the aerostat main axis direction g_1 (see Fig. 1a), α_s – the sail axis direction, and α_m – the aerostat motion direction. In addition, α_w^a is the wind direction relative to the direction of the aerostat motion, which results from both the direction α_w and velocity of the wind v_w as well as the direction α_m and velocity of the aerostat motion v_m in a global coordinate system:

$$\alpha_w^a = f(\alpha_w, v_w, \alpha_m, v_m). \quad (1)$$

On the other hand, the angles denoted by β are referred to relative wind direction, namely: β_a – the direction of the aerostat with respect to relative wind direction and β_s – the direction of sail with respect to relative wind direction. Both the β_a and β_s denote the angles of attack (of the aerostat and sail, respectively) and they can be used for simple and intuitive parameterization of the results obtained from CFD analyses. Moreover, both these angles can be expressed in terms of angles in global coordinate system:

$$\beta_a = \alpha_a - \alpha_w^a, \quad \beta_s = \alpha_s - \alpha_w^a. \quad (2)$$

Finally, γ_s denotes angle between the sail and the aerostat:

$$\gamma_s = \alpha_s - \alpha_a = \beta_s - \beta_a. \quad (3)$$

Let us note that according to the above discussion, we have distinguished eight angles and four relations between them, which reveals that four angles remain independent. The set of independent angles can be conveniently selected as: $\alpha_w, \alpha_a, \gamma_s, \alpha_m$, where: α_w – is a given time-dependent quantity, α_a – is one of the main unknowns of the system, γ_s – is one of the control variables, while α_m – can be expressed in terms of basic unknown components of aerostat motion x and y .

The next step of model development is computation of aerodynamic forces and moments acting on the aerostat with respect to the angles of attack β_a and β_s as well as the wind velocity relative to the aerostat v_w^a . These forces and moments are determined using the CFD model of the flow around aerostat described in the next section. The forces acting in the direction parallel and perpendicular to the wind direction (P_1 and P_2) as well as torque acting on the aerostat M_p can be expressed in general forms:

$$P(\beta_a, \beta_s, v_w^a) = P(\alpha_a, \alpha_s, \alpha_w^a, v_w^a) = P(\alpha_a, \gamma_s, \alpha_w, v_w, \alpha_m, v_m) = P(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}), \quad (4)$$

where: $P = \{P_1, P_2, M_p\}$. Further, forces referred to global coordinate system (P_x and P_y) can be calculated using standard transformation:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = T(\alpha_w^a) \begin{bmatrix} P_1(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) \\ P_2(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) \end{bmatrix} = \begin{bmatrix} P_x(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) \\ P_y(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) \end{bmatrix},$$

while the torque M_p does not require transformation to global coordinate system and thus $T(\alpha_w^a)$ is the transformation matrix with dimensions 2 by 2. Let us note that determined forces depend on angle of aerostat rotation α_a , angle of the sail γ_s (one of system control variables), known direction and velocity of the wind α_w and v_w as well as unknown components of aerostat velocity \dot{x}, \dot{y} . Similarly, forces generated by the auxiliary lateral engines in local coordinate system of the aerostat $\{F_1, F_2\}$ can be transformed to global coordinate system and expressed as follow:

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = T(\alpha_a) \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} F_x(F_1, F_2, \alpha_a) \\ F_y(F_1, F_2, \alpha_a) \end{bmatrix}. \quad (5)$$

Moreover, the torque corresponding to the operation of the engines M_F can be calculated in a standard manner.

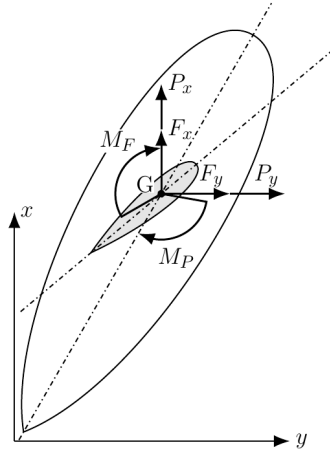


Figure 3. Force systems acting on the airship (top view).

The above dependencies allow to define equations of motion governing the process of *sky sailing* in the horizontal plane:

$$\begin{aligned} m\ddot{x} &= P_x(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) + F_x(F_1, F_2, \alpha_a) \\ m\ddot{y} &= P_y(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) + F_y(F_1, F_2, \alpha_a) \\ I\ddot{\alpha}_a &= M_P(\alpha_a, \gamma_s, \alpha_w, v_w, \dot{x}, \dot{y}) + M_F(F_1, F_2, \alpha_a) \end{aligned} \quad (6)$$

where m is the total mass of the airship, I is the moment of inertia of the airship, P_x and P_y are components of aerodynamic forces acting on the airship in the x and y direction, respectively, F_x and F_y are forces generated by the propulsion system in the x - and y -direction, respectively, while M_F and M_P are torques generated by aerodynamic forces and propulsion system, respectively. The presented form of the governing equations reveals the dependencies of generated forces and torques on the main unknowns of the problem x, y, α_a , known parameters of the wind α_w, v_w as well as control variables γ_s, F_1, F_2 .

3. NUMERICAL ANALYSES OF THE FLOW AROUND AEROSTAT

Numerical studies related to the determination of the values of forces and torques acting on the aerostat model and used in equations of aerostat motion (Eq. 6) were carried out using Ansys Fluent software. The obtained results allowed to create the database established for given, discretized values of aerostat hull and sails angles of attack wind velocity as well as wind velocities. For the purpose of solving the Eq. 6 the intermediate values of forces and torques were interpolated.

The full 3D numerical model of the airship consists of a hull and two sails in the form of aerofoils mounted rotatably in a controlled way with respect to a vertical axis passing through the centre of gravity G of the entire structure. The position of the centre of gravity G was determined based on the design of the structure equipped with the basic systems necessary to perform the assumed mission. The profile model of the aerostat section axis has a length of 12 m in the longitudinal and a diameter of maximum value of 3 m. The value of the aerostat's lateral area is equal to 91.56 m², while the projected area of the aerostat's

surface is 7.08 m^2 in the longitudinal direction and 28.22 m^2 in the transverse direction.

The aerostat is equipped with two sails mounted vertically under and above the hull with possible rotation about the vertical axis g_3 (see Fig 1a). A single sail has a length of about 2.97 m, a maximum width of 0.551 m and a height of 1.995 m. The value of the lateral area of the entire sail is 12.31 m^2 , and the projected area of the aerostat surface in the longitudinal direction is 0.895 m^2 and in the transverse direction is 5.98 m^2 . It can be seen that the values of the projected areas of the sail are relatively large compared to the projected areas in the corresponding directions.

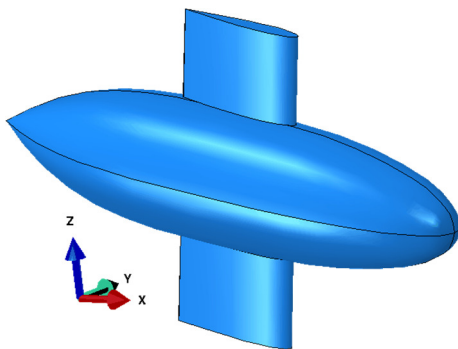


Figure 4. Designed airship geometry used for finite volume method analysis.

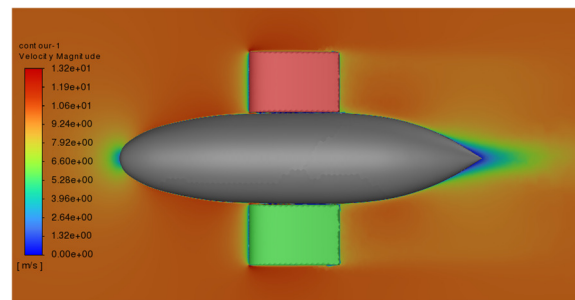


Figure 5. Airflow velocity distribution in the XZ plane: $\beta_s = 0$, $\beta_a = 0$, $v = 10 \text{ m/s}$.

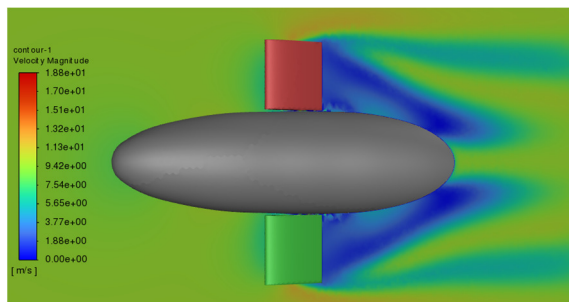


Figure 6. Airflow velocity distribution in the XZ plane: $\beta_s = 15$, $\beta_a = 15$, $v = 10 \text{ m/s}$.

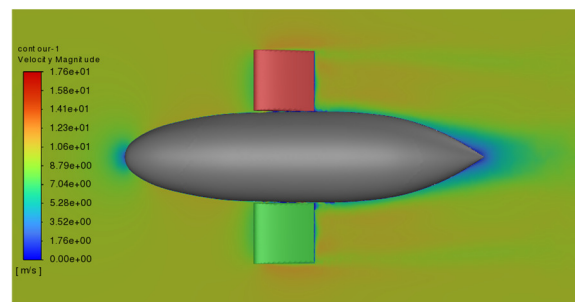


Figure 7. Airflow velocity distribution in the XZ plane: $\beta_s = 15$, $\beta_a = 0$, $v = 10 \text{ m/s}$.

During the analyses, it was assumed that the aerostat is surrounded by air with a density resulting from a given altitude, which was selected based on the NASA atmosphere modelⁱ. It was assumed that in each direction the space surrounding the aerostat have a size of 40 by 60 m, whose length ratio is between 3.5 and 5 values of the aerostat's length. The conducted simulations have shown that this is a sufficient value for analysis. The numerical model of the aerostat and the surrounding air consist of 3.6 million nodes, 4.4 million surface elements refined in the areas of geometry changes. The mesh in the vicinity of the aerostat and sails was also refined. Further away from the model at the edges of the fluid model, a larger elements dimensions were used. Based on the aforementioned nodes and surface elements, a fluid mesh of about 660 thousand polyhedral cell volume elements was created. In order to reflect the basic fluid behaviour in the boundary layer, a boundary layer consisting of 5 volume elements was built.

The numerical model of the aerostat including sails was parameterized by using the following assumptions concerning parameters variations:

ⁱ The model is available at NASA's web page (accessed 2023 May 31): <https://www.grc.nasa.gov/www/K-12/airplane/atmosmet.html>.

- The parameter describing the angle of deflection of the aerostat in the horizontal plane – during the analyses, the variation in the range of -30 to $+30$ degrees from the direction of airflow was allowed;
- The parameter describing the angle of deflection of the sails relative to the fuselage – during the analyses, the angle was allowed to vary in the range of -15 to $+15$ degrees relative to the hull;
- The air velocity was varied in the range of 2.5 – 10 m/s. The range of such speeds was due to the assumption of the limitation of wind speed in real conditions during the planned experimental tests.

During the numerical tests, the air flow was assumed to occur in the horizontal XY plane in the X-axis direction. The vertical direction was described by the Z-axis. The simulations were carried out for various combinations of the above-mentioned parameters, which allowed to determine the values of forces acting in each direction and the values of moments with respect to each mentioned axis. The values of moments were determined with respect to axes passing through the centre of gravity G located in the axis of the aerostat at a distance of 5.75 m from its tip.

Figures 5-7 show the distribution of velocity in the XZ plane around the aerostat for different hull and sail deflection angles. It can be seen that both changing the angle of the hull relative to the wind and the sail causes significant changes in the streamlining of the aerostat with the sails by the air.

4. FORMULATION OF THE CONTROL PROBLEM FOR SKY SAILING

The control problem corresponding to the sky sailing concept involves following the assumed trajectory of aerostat horizontal motion with minimal energetic cost. The most general form of the objective function includes the integral term denoting global error of the trajectory tracking and integral term denoting work done by auxiliary engines. The control is conducted with respect to actual angle of the sail and forces generated by the engines. The constraints of the problem are equations of motion of the system. The complete mathematical formulation of such a control problem takes the form:

$$\text{Minimize: } \int_0^T [s(x, y) - \bar{s}(x, y)]^2 dt + q \int_0^T [F_x \dot{x} + F_y \dot{y}] dt$$

$$\text{with respect to: } \gamma_s(t), F_1(t), F_2(t)$$

$$\text{subject to: } m\ddot{x} = P_x + F_x, \quad m\ddot{y} = P_y + F_y, \quad I\ddot{\alpha}_a = M_P + M_F$$

where $\bar{s}(x, y)$ denotes the assumed trajectory of motion in the horizontal plane, while q is the weighting coefficient of the control cost term. Let us note that one of the proposed control variables $\gamma_s(t)$ corresponds to the semi-active control of the proposed system, while two remaining control variables $F_1(t)$ and $F_2(t)$ correspond to the active control. Therefore, the presented formulation can be considered as a joint problem of semi-active and active control. However, depending on the value of the weighting coefficient of the control cost term q , two opposite cases can be distinguished. In the first case, by setting relatively large value of the weighting coefficient, the active control terms are eliminated and the formulation becomes a semi-active control problem with respect to angle of the sail $\gamma_s(t)$. In the second case, when the weighting coefficient is set to zero the control problem is aimed at optimal following of the assumed trajectory regardless of the control cost.

The above formulated control problem can be solved using various approaches, semi-

analytical or numerical methods. One of the possibilities is application of the so called *inverse dynamics method* known from the problems of adaptive impact absorption. The possibilities of applying such method to both above mentioned versions of the control problem will be the subject of the further research.

5. CONCLUSIONS

The article presents the concept of an airship propelled by two engines and equipped with vertical sails with adjustable angle of attack. The objective of the work is to present an effective approach for controlling the airship motion by using aerodynamic loads generated by the sails while minimizing the engines' usage. In order to solve this problem, a 2D mathematical model with three degrees of freedom (airship rotation angle and two displacements) is developed, where the control is achieved by changing sails' angle of attack and engines' thrusts. The mathematical model is supported by a three-dimensional finite volume model developed in the Ansys Fluent, which is used to create a reference database containing aerodynamic forces acting on the airship for selected discrete values of wind directions and sails attack angle. Eventually, the problem of airship sailing along a pre-defined mission path at a fixed altitude is formulated, and potential methods of its solution are proposed. Such defined problem, particularly the numerical analysis of controlled airship movement, will be the subject of further work of the authors.

ACKNOWLEDGMENTS

The authors acknowledge the support of the National Centre for Research and Development and the National Science Centre, Poland, granted in the framework of the TANGO 4 programme (project TANGO-IV-C/0001/2019-00) and the project DEC-2017/25/B/ST8/01800 of the National Science Centre, Poland.

REFERENCES

- [1] C. T. Chaplain et al. Future aerostat and airship investment decisions drive oversight and coordination needs, Tech. rep. GAO-13-81. United States Government Accountability Office, 2012. URL: <https://www.gao.gov/assets/650/649661.pdf>.
- [2] J. J. S. Rees (ed), Recent development efforts for military airships, The Congress of the United States, Congressional Budget Office, 2011,
- [3] URL: <https://fas.org/irp/program/collect/cbo-airship.pdf>.
- [4] M. S. Smith, E. L. Rainwater, Optimum designs for superpressure balloons, *Advances in Space Research*, vol. 33(10), pp. 1688-1693, 2004, doi:10.1016/j.asr.2003.07.042.
- [5] J. Hu, Y. Li, W. Chen, et al., Uniaxial mechanical properties of multi-layer thin films in use for scientific balloons, *Advances in Space Research*, vol. 62(5), pp. 1165-1176, 2018, doi:10.1016/j.asr.2018.06.003.
- [6] B. S. Kumar et al., Mechanical properties of ANTRIX balloon film and fabrication of single cap large volume balloons, *Advances in Space Research*, vol. 42(10), pp. 1691-1697, doi:10.1016/j.asr.2008.03.030.
- [7] B. S. Kumar, N. Nagendra, D. K. Ojha, G. S. Peter, R. Vasudevan, D. Anand, P. M. Kulkarni, V. A. Reddy, Development of Ultra-Thin Polyethylene Balloons for High Altitude Research up to Mesosphere, *Journal of Astronomical Instrumentation*, 2014, doi:doi.org/10.1142/S2251171714400029.
- [8] M. Gai, G. Guglieri, M. G. Lattanzi, A. Lombardi, M. Mana, L. Masserano, I. Musso, P. Navone, A Scientific Mission Based on a High Altitude Stratospheric Balloon,

- International Journal of Aerospace Sciences, 2014, 3(1): 18-29
- [9] Ö. Kayhan, M. A. Hastaoglu, Modeling of Stratospheric Balloon Using Transport Phenomena and Gas Compress-Release System, Journal of Thermophysics and Heat Transfer, 2014, vol. 28, no. 3.
- [10] S. Saleh, H. E. Weiliang, New design simulation for a high-altitude dual-balloon system to extend lifetime and improve floating performance, Chinese Journal of Aeronautics, 2018, vol. 31, no. 5, pp. 1109-1118.
- [11] K. T. Nock, K. M. Aaron, M. K. Heun, A. A. Pankine, Aerodynamic and mission performance of a winged balloon guidance system, Journal of Aircraft, vol. 44 (6), pp. 1923-1938, 2007, doi:10.2514/1.31922.
- [12] H. Du, M. Lv, J. Li, W. Zhu, L. Zhang, Y. Wu, Station-keeping performance analysis for high altitude balloon with altitude control system, Aerospace Science and Technology, vol. 92, pp. 644-652, 2019, doi:10.1016/j.ast.2019.06.035.
- [13] L. Knap, A. Świercz, C. Graczykowski, J. Holnicki-Szulc, The concept of self-deployable helium-filled aerostats based on tensegrity structures, AeroBest 2021, International Conference on Multidisciplinary Design Optimization of Aerospace Systems, Lisboa, pp.3-13.
- [14] L. Knap, A. Świercz, C. Graczykowski, J. Holnicki-Szulc, Self-deployable tensegrity structures for adaptive morphing of helium-filled aerostats, Archives of Civil and Mechanical Engineering, vol. 21(4), no.159, 2021, doi:10.1007/s43452-021-00292-6.



EXPLORING THE POTENTIAL OF DEEP LEARNING IN OPTIMIZING AN AERIAL PHOTOGRAMMETRY MISSION

Shahab Sotouni¹, Matthew Tucsok², Kumaraditya Gupta³, Iraj Mantegh⁴
and Homayoun Najjaran^{1*}

1: Department of Mechanical Engineering
Faculty of Engineering and Computer Science
University of Victoria
3800 Finnerty Rd, Victoria, BC, V8P 5C2, Canada
{shahabsotouni, najjaran}@uvic.ca, <https://www.uvic.ca/>

2: Okanagan School of Engineering
Faculty of Applied Science
University of British Columbia
3333 University Way, Kelowna, BC, V1V 1V7, Canada
tucsok@student.ubc.ca, <https://ok.ubc.ca/>

3: Department of Electrical and Electronics Engineering
Birla Institute of Technology Science, Pilani
Vidya Vihar, Pilani 333031, Rajasthan, India
f20190776@pilani.bits-pilani.ac.in, <https://bits-pilani.ac.in/>

4: Aerospace Research Centre
National Research Council Canada
5145 Decelles Ave., Montreal, QC, H3T 2B2, Canada
iraj.mantegh@cnrc-nrc.gc.ca, <https://nrc.canada.ca/>

Abstract. *In this paper, we investigate how deep learning can improve different parts of aerial photogrammetry. We focus on creating flexible waypoints on-the-fly (Next Best View planning) and dealing with challenges when working with multiple drones. By checking if real-time adaptive waypoint generation for drone navigation is possible and exploring issues in multi-drone systems like task assignment and formation changes, we hope to find ways that deep learning neural networks can enhance aerial photogrammetry, ultimately making data collection more efficient and accurate, and moving the field forward.*

Keywords: Uncrewed Aerial Vehicles (UAVs), Aerial Photogrammetry, Deep Reinforcement Learning, Next Best View Planning, Multi-Agent Systems, Proximal Policy Optimization

1 INTRODUCTION

Uncrewed aerial vehicles (UAVs) have revolutionized aerial photogrammetry, enabling more precise and efficient data collection. However, this field still has untapped potential waiting to be explored. This paper delves into the power of deep learning in enhancing various aspects of aerial photogrammetry, focusing on creating adaptable and real-time waypoint generation (Next Best View planning) and managing multi-drone systems safely.

In the first part of this paper, we present a novel framework for Next Best View (NBV) planning for 3D reconstruction using UAVs. This framework comprises two deep learning models. The first model is trained on 3D reconstruction from single images and sorts views based on reconstruction performance [1]. Then, Oriented FAST and Rotated BRIEF (ORB) [2] feature filtering is used to find distinct features from the selected views based on their reconstruction performance in the previous step. The second model, which is called the Graph Best View Finder (GBVF) predicts the NBV for aerial image acquisition by leveraging the learned representation of the first model. More precisely, the GBVF model generalizes unseen objects from the classes it has been trained on.

Conventional photogrammetry techniques depend on pre-determined waypoints tailored to individual or multiple agents. However, with the advent of innovative approaches, such as the on-the-fly waypoint generation method presented in the first section of this paper or any other method that does not rely on detailed prior knowledge of the environment and target, there is an increasing demand for a system capable of adaptively managing path planning for multi-agent photogrammetry missions. This system must consider safety measures, optimization strategies, and constraints specific to each mission.

In the second part of this paper, we explore the potential of Reinforcement Learning (RL) in managing multi-drone systems in a simplified environment. Reinforcement Learning is a suitable approach for managing multi-drone systems due to its ability to handle dynamic and uncertain environments, making it ideal for real-time decision-making and adaptation. The environment is a simple grid-world that includes multiple agents navigating to waypoints while remaining within the mission geofence and avoiding obstacles and other UAVs. We leverage a Proximal Policy Optimization (PPO) [3] Reinforcement Learning algorithm, which shows great promise for further development and application to more complex tasks. PPO has been selected for its relatively less complicated implementation and robustness to hyper-parameters.

Our research demonstrates the significant benefits of incorporating deep learning into aerial photogrammetry. This technology drives the field forward and improves efficiency by reducing the time-consuming pre-planning stage.

2 NEXT BEST VIEW (NBV) PLANNING

To improve any aerial photogrammetric 3D reconstruction, one must consider how images are captured by the drone during data acquisition. Attempts have been made to generate waypoints based on the geometry of pre-defined boundaries or by considering sequential images' horizontal and vertical overlaps [4]. However, these methods often fail to capture areas of the scene that are occluded and neglect what the drone is currently viewing when proposing new waypoints and orientations to take the next images. For this reason, integrating a method for predicting the NBV into the data acquisition phase of the photogrammetry pipeline holds significant promise for optimizing the reconstruction

process, minimizing manual intervention during flight operations, and achieving higher-quality reconstructions with greater ease.

In the following subsections, we introduce a novel supervised deep learning framework for approaching the NBV problem within the context of aerial photogrammetry.

2.1 Related Works

Deep learning approaches have recently emerged as a viable method for addressing the NBV planning problem in 3D reconstruction. These systems use sophisticated neural networks to learn successful view selection and path planning strategies and have occasionally outperformed classic probabilistic algorithms.

The work in [5] proposes a volumetric-based planning method for single-object modelling. Before this approach, NBV computing was viewed as a search problem [6, 7] in which the goal was to locate the view that maximized a metric. Unlike prior approaches, [5] aimed to forecast the NBV directly using the information provided by the partially rebuilt model. They create a discrete NBV search space and employ a classification strategy to generate a collection of possible classes. A sphere tessellation is used to produce a distinct set of viewpoints on which a model predicts NBV classes using a 3D-Convolutional Neural Network (CNN) architecture. The results demonstrated that NBV-Net could recreate unknown items not visible during training and over the majority of the surface. This model furthered the state-of-the-art NBV approaches.

Other approaches like [8] offered an NBV planning method that uses a 3D-CNN to assess the utility of a viewpoint. As an input to the CNN, the approach used a multi-scale voxel representation of a partially analyzed scene. It trains utility scores of opinions from an oracle with access to real-world data. Recent work in [9] also compares the volumetric information gain measures and introduces several information gain functions for volumetric viewpoint evaluation. They counted unobserved voxels to quantify observable information and tested the proposed information gain functions' performance using various modelling situations.

A recent review of the methods for NBV planning [10] concludes that, while there is a large body of literature on AI-based path and coverage planning, viewpoint planning for 3D reconstruction based on AI and deep learning approaches is still in its infancy, with much room for further research.

2.2 Framework Overview

To develop a deep learning-based NBV planning that can be trained for automatic UAV-based 3D reconstruction, a novel framework comprising two parts is proposed; both deep learning models can be trained end-to-end. The first of these two models is a single-view autoencoder-like network adapted from [1], trained on 3D voxel reconstruction from single views. This model is used to create a labelled dataset which creates a sequence of canonical views for training the second model. The second end-to-end model fuses a modified graph-based camera relocalization network from [11], which is called the Graph Feature Extractor (GFE), with a Long Short-Term Memory (LSTM) [12] network for next pose prediction. The GFE serves to embed features from views that are spatially separated, while the LSTM learns the sequential capturing behaviour over the course of aerial photogrammetry data acquisition. The combination of the GFE and LSTM forms the Graph Best View Finder (GBVF).

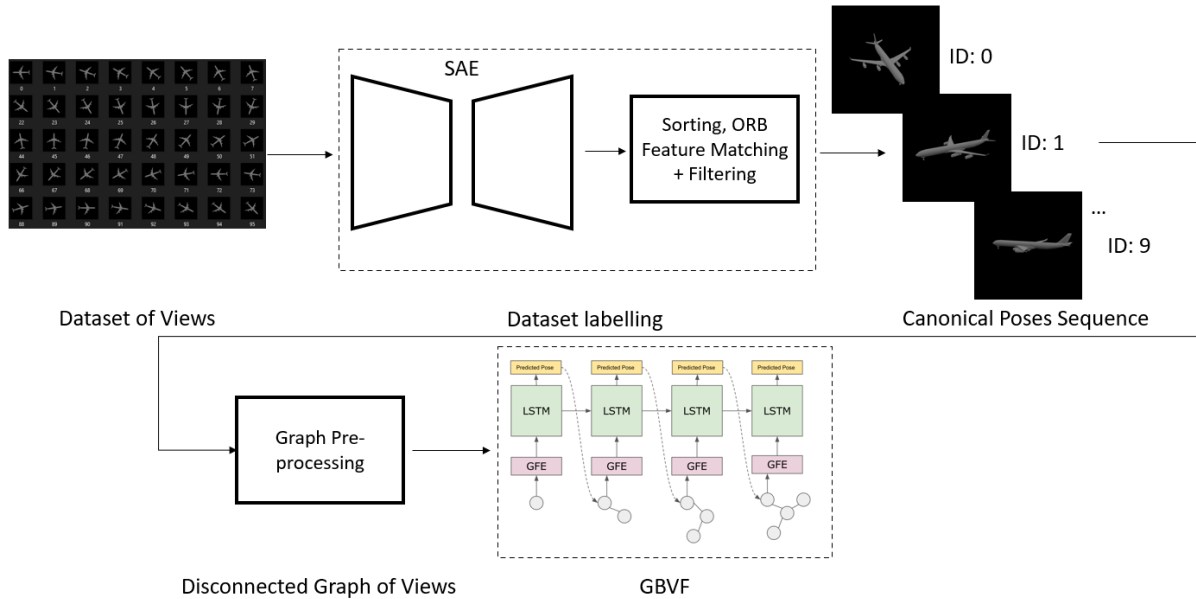


Figure 1: An overview of the framework for training the Graph Best View Finder (GBVF).

2.3 Dataset

The dataset to train the Sparse Auto-Encoder (SAE) is generated using ModelNet40 objects [13]. ModelNet40 does not have the same number of objects for each class, so the datasets used to train the SAE are balanced by sampling the same number of objects from each class. The first dataset contains 100 objects from 10 classes, and the larger dataset contains 800 objects from 20 classes. BlendTorch [14] was used to generate the datasets by taking approximately 300 images of each object in Blender tessellated evenly on an upper hemisphere centred on the object.

2.4 SAE Canonical Pose Creation

The first part of canonical pose creation requires an SAE model [15], which gives a crude 3D reconstruction of an object from a single view. This reconstruction, though not of great quality, helps determine the utility of a particular view by creating a 3D occupancy grid of the object of interest by comparing it to the ground truth. In the results of previous works [1], it is observed that for a deep-learning 3D reconstruction from multiple views, sampling views which have the highest Intersection-over-Union (IoU) and also have a threshold for the number of similar ORB features amongst each other, have better reconstruction accuracy than other subsets of views. The poses of the sampled views were used as the label for the sequence of canonical views for a given object. As a result, the SAE acts as a dataset labeller for the GBVF model.

2.5 Graph Construction and Graph Neural Network (GNN) Architecture

A dense graph of the collected views is iteratively constructed, starting with a random initial pose. A latent space representation is learned for the graph using a pre-trained image feature extractor network and a GNN. The choice of the use of GNN is motivated by recent papers having shown that they take advantage of geometric relationships between images of the same object through message-passing networks [11, 16, 17]. The node

message and edge representations are all constructed and learned using a soft-attention mechanism [18].

A graph is commonly denoted as $G = (V, E)$, where $V = \{v_i\}$ and $E = \{e_{ij}\}$ symbolize the nodes and edges, respectively. Each node v_i has an attribute x_i , while each edge $e_{ij} = (v_i, v_j)$ links the nodes v_i and v_j .

Each node's representation is kept as a 3D tensor to accommodate the spatial correlations amongst image features, which are crucial for knowledge exchange across multiple perspectives. Consequently, each feature map $x_i \in \mathcal{R}^{H \times W \times C}$ (with H , W , and C standing for height, width, and channels) can be integrated into the graph without any loss of information. As a result, the message passing and node updating functions are modified to facilitate information exchange in the form of 3D tensors. Initially, a dense directed graph is built by linking every pair of nodes with edges. This approach allows temporally separated images to influence each other. The graph model enables global information distribution to each node.

The GNN architecture used is similar to and inspired by [11]. The message is determined from features at the node using convolutions to maintain spatial associations. The generation function computes message $m_{j \rightarrow i}$ from x_i and x_j . Both x_i and x_j are concatenated along the channel dimension and then fed into two convolution layers with a kernel size of 3×3 to generate the message. The output channels of both layers are identical to x_i , rendering $m_{j \rightarrow i}$ the same size as x_i and x_j . These operation steps are also shown in Fig.2.

$$m_{j \rightarrow i} = f_m(x_i, x_j) \quad (1)$$

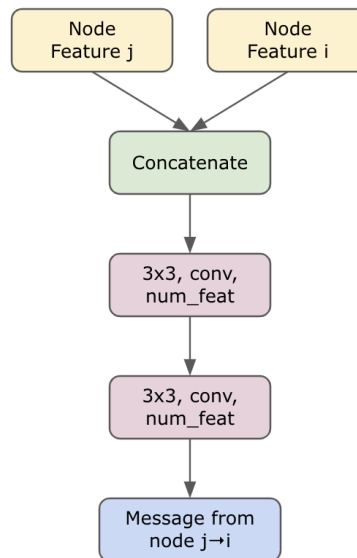


Figure 2: The f_m function which generates the message $m_{j \rightarrow i}$.

An attention mechanism also determines the channel-wise correlations between connected nodes.

$$a_{j \rightarrow i} = f_{\text{atten}}(x_i, x_j) \quad (2)$$

$$a_{j \rightarrow i}^{(k)} = \sigma \left(\text{cs} \left(\text{vec} \left(x_i^{(k)} \right), \text{vec} \left(x_j^{(k)} \right) \right) \right) \quad (3)$$

This function computes the cosine similarity between each vectorized k^{th} channel of linked features x_i and x_j . The attention value is subsequently normalized to the range (0, 1) by the sigmoid function σ .

The message m_i^{agg} for node v_i is compiled by collecting messages from all originating nodes with soft attention incorporated, as follows:

$$m_i^{\text{agg}} = \frac{1}{N_i} \sum_{e_{ij} \in E} a_{j \rightarrow i} \otimes m_{j \rightarrow i} \quad (4)$$

where N_i represents the count of originating nodes linked to vertex v_i and \otimes symbolizes the channel-wise multiplication.

The feature of v_i is enhanced by the aggregated message function, which maintains the same structure as the message generation function f_m , but employs a different set of parameters. These operations are shown in Fig.3

$$x'_i = f_u(x_i, m_i^{\text{agg}}) \quad (5)$$

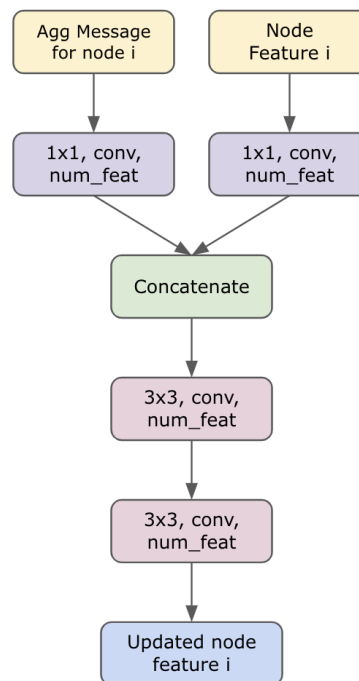


Figure 3: The f_u function which outputs the updated node feature x'_i from the aggregated message and previous feature.

2.6 Graph Feature Extractor (GFE)

The GFE uses the GNN block described in Section 2.5 with pre-trained ResNet34 layers [19] for iterative feature extraction. Following each GNN block, edge pooling is executed to eliminate redundant edge connections. The GFE design is modified from the camera pose regression architecture used in [11].

As the information is gradually passed amongst multiple views from the lower to higher levels, there is a limited requirement to retain connections with minimal values. In this context, the network adapts and makes decisions based on the correlation between linked nodes, as follows:

$$c_{j \rightarrow i} = \text{cs}(\text{maxpool}(x_i), \text{maxpool}(x_j)) \quad (6)$$

The correlation $c_{j \rightarrow i}$ is determined using cosine similarity on the downsampled features of x_i and x_j . For each node, only k neighbours with the highest correlations are retained, while the others are discarded. Figure 4 shows the architecture to obtain the final graph feature from the image graph with n nodes.

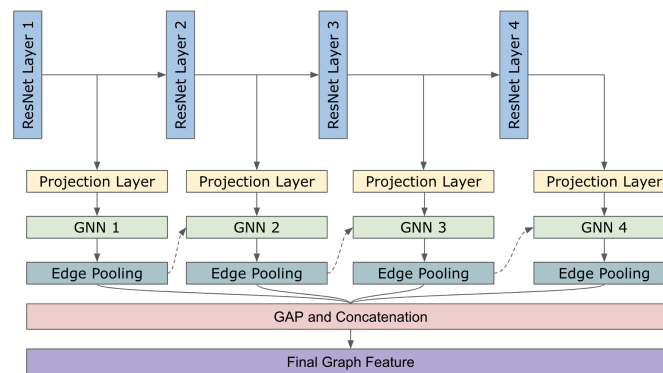


Figure 4: The Graph Feature Extractor (GFE) architecture depicts the ResNet34 layers, GNNs, and dynamic edge pooling.

The features from four levels contribute to capturing both high-level and low-level features depicted in the scene. These features are concatenated and pooled to construct a single feature vector for the graph, representing the n initial images.

2.7 Graph Best View Finder (GBVF)

To mimic data acquisition using a UAV in a real-world scenario, where only the rough location of the target of interest is known, a single image captured from a random initial pose is given to the GFE block to generate a feature vector. This vector is given as input to the LSTM block, which has a hidden size of 2048. This LSTM block outputs the prediction for the next canonical pose. The image from this pose is used to construct a graph, now comprising two nodes, and is once again passed through the GFE and LSTM to predict the next view. This process is iterated n times to predict n canonical poses using the GBVF.

The network is trained with $n = 10$ on the dataset of 800 objects. During training, a method akin to teacher-forcing is adopted, i.e. the correct canonical pose is incorporated into the graph each time, even if the LSTM predicted different pose values in the previous step. This aids the network in achieving faster convergence. Figure 5 shows the architecture to obtain these predicted poses from the LSTM output.

Prior to training the GBVF network, the dense graphs are generated from the canonical pose views that will be inputted into the LSTM to reduce the number of redundant graph creations. For a single object, the graphs are stored in a single file. A batch of objects is

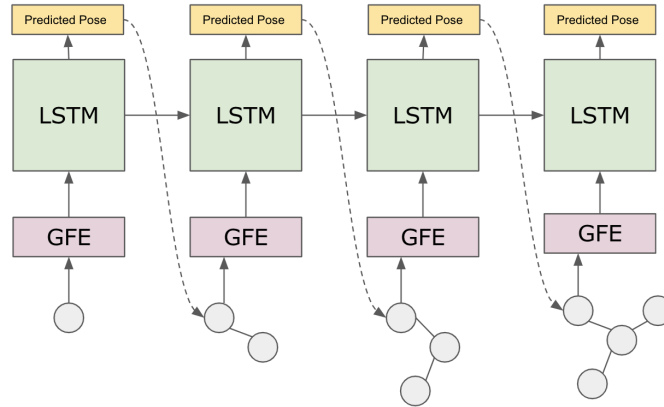


Figure 5: The Graph Best View Finder (GBVF) architecture depicts the use of the GFE to create the input feature to the LSTM.

created at the time of training using the PyTorch Geometric library [20], which forms a larger disconnected graph for the m objects ($m = \text{batchsize}$).

The loss function employed to train the model is similar to the function used to balance the position and orientation errors where $\zeta_i = (\mathbf{t}_i, \mathbf{r}_i)$ and $\hat{\zeta}_i = (\hat{\mathbf{t}}_i, \hat{\mathbf{r}}_i)$ represent the ground-truth and predicted 6-Degree-of-Freedom (DoF) absolute poses. Three values are for the position and four for the quaternion orientation. β_p and γ_p are the parameters balancing the position and orientation errors.

$$\mathcal{L} = \frac{1}{N_v} \sum_{v_i \in V_{\text{pose}}} d(\zeta_i, \hat{\zeta}_i) \quad (7)$$

$$d(\zeta_i, \hat{\zeta}_i) = \|\mathbf{t}_i - \hat{\mathbf{t}}_i\|_1 e^{-\beta_p} + \beta_p + \|\mathbf{r}_i - \hat{\mathbf{r}}_i\|_1 e^{-\gamma_p} + \gamma_p \quad (8)$$

2.8 Training

The network is trained to predict 10 poses corresponding to the length of the canonical view sequences from the SAE and feature-filtering model. During training, the GBVF was observed to generalize better when it had more objects from a single class. Training on merely 100 object graphs did not lead to a significant reduction in loss. However, with 800 objects, the network could generalize to unseen objects and predict viable next canonical poses. The loss graph for the training and validation set is depicted in Fig.6.

2.9 Inference

Only objects from the validation set are considered to test the inference performance of the GBVF. As shown in Fig.7, the GBVF successfully generates a valid sequence of views, where the target object is being viewed from all around, while capturing the important features.

To evaluate the reconstruction performance of these views, TSDF-Fusion [21] is used as a lightweight reconstruction method which can visualize the coverage performance from a sequence of views. As shown in Fig.8, the reconstruction resulting from the set of 10 views predicted by the GBVF is not significantly worse than that produced by a hemispherical tessellation with 269 views.

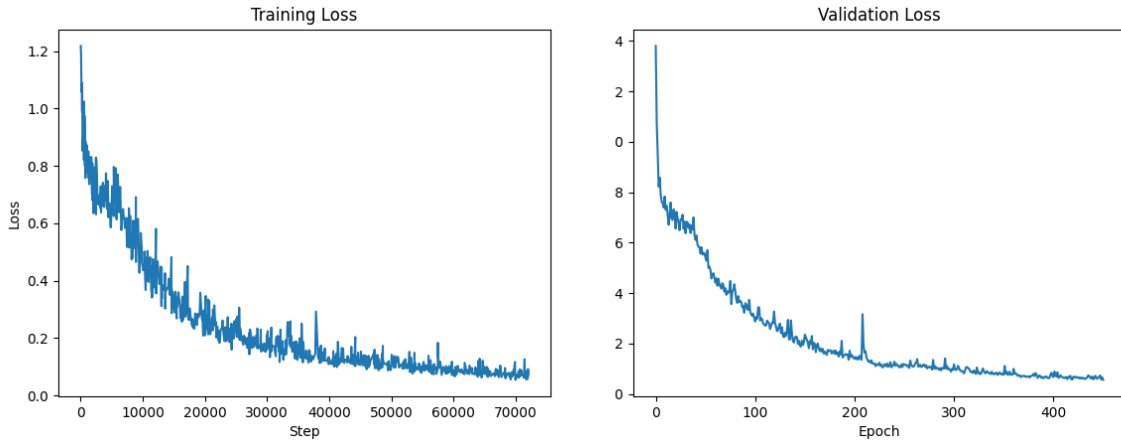


Figure 6: The training and validation loss versus steps (left) and epochs (right).

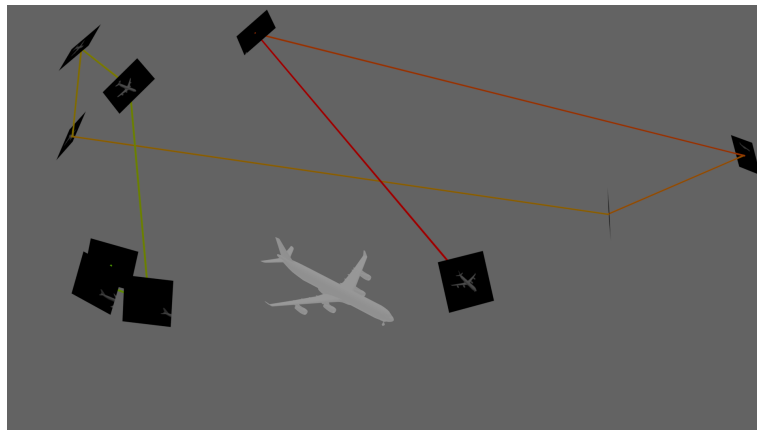


Figure 7: The GBVF predicts 9 views sequentially when given only the best canonical view identified by the SAE for ModelNet40 airplane_0029.

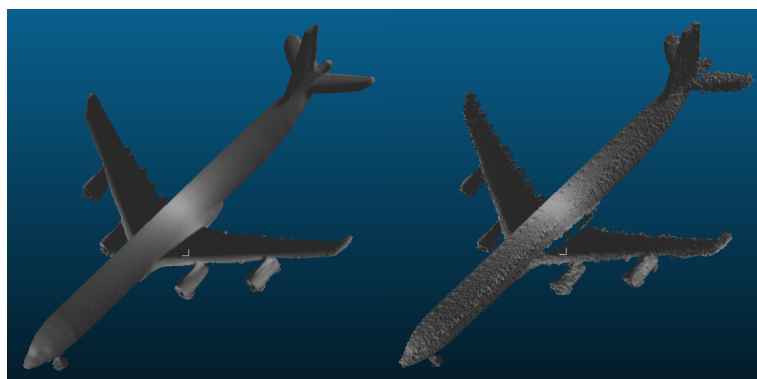


Figure 8: A reconstruction using TSSDF-Fusion [21] of ModelNet40 airplane_0029 from 269 views using the tessellated hemisphere method (left) versus the resulting reconstruction from the sequence of 10 views predicted by the GBVF Fig.7 (Right).

3 REINFORCEMENT LEARNING FOR MULTI-AGENT GUIDANCE

As we venture into multi-drone systems in aerial photogrammetry, the complexity of managing these systems increases exponentially. Traditional methods, which rely on

pre-determined waypoints [22], fail to provide the flexibility and adaptability required in dynamic environments. This necessitates exploring novel approaches that can handle the intricacies of multi-agent missions while ensuring safety, optimization, and adherence to mission-specific constraints. This section examines the potential of Reinforcement Learning (RL) as a promising solution to these challenges. With its ability to handle dynamic and uncertain environments, RL is presented as an ideal framework for real-time decision-making and adaptation in multi-drone systems. The exploration of a Proximal Policy Optimization (PPO) [3] RL algorithm within a simplified environment is presented here, demonstrating its potential for further development and application to more complex tasks.

3.1 Related Works

Advancements in Uncrewed Aerial Vehicle (UAV) designs have opened up a myriad of applications, ranging from surveillance [23] to firefighting [24], cellular networks, and delivery [25] applications. A notable approach towards UAV trajectory planning was proposed by Khalil et al. [26], who developed REPlanner, an economic game-inspired multi-agent RL algorithm. This method focuses on solving the path planning problem as a multi-agent economic game where agents (UAVs) cooperate and compete for resources.

A different approach was introduced by Westheider et al. [27], where they proposed a multi-agent informative path planning method based on deep RL for terrain monitoring scenarios using UAV teams. Their method leverages a counterfactual baseline to learn cooperative behaviour, demonstrating superior performance to state-of-the-art non-learning-based methods and adaptability to varying team sizes and communication constraints.

Huang et al. [28] focused on collision avoidance in UAV swarms. They proposed a multi-agent critic-actor learning scheme called MACA for UAV swarm collision avoidance. The key feature of their method is also the use of a counterfactual baseline, which evaluates the importance of an agent in the joint observation-action space. Simulations demonstrated that MACA significantly outperformed state-of-the-art multi-agent RL algorithms regarding average reward, failure rate, and response time.

In the context of using multiple UAVs for wireless data harvesting from distributed Internet of Things (IoT) devices, Bayerlein et al. [29] proposed a deep RL approach which adapts to significant changes in mission parameters without the need for expensive re-computations or relearning control policies. They highlighted that their method can effectively cooperate, adapt to large complex environments, and make movement decisions that balance data collection goals, flight-time efficiency, and navigation constraints. Qie et al. [30] developed a Simultaneous Target Assignment and Path Planning (STAPP) system based on a Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. Their system can effectively handle dynamic environments with the execution only requiring the locations of the UAVs, targets, and threat areas.

Lastly, Chen et al. [31] addressed the issue of incomplete information and low efficiency in multi-UAV path planning. They introduced a centralized training–decentralized execution RL method to solve the path planning problem, demonstrating successful autonomous path planning in an unknown environment. Each of these studies provides a unique perspective on using RL for managing multi-drone systems in various applications, forming a strong foundation for exploring more advanced techniques in this field.

3.2 The Environment and Reward Function

This work represents the environment as a two-dimensional grid of 20×20 squares. Three agents are introduced into this environment, each able to move North, East, South, West, or remain stationary. A penalty is associated with each movement to promote efficiency, and remaining stationary is penalized to encourage exploration.

Five waypoints are randomly generated and distributed across the grid. The environment also contains an obstacle around which the agents must navigate. The characteristics of this obstacle, including its size and location, are randomly generated for each training episode.

When an agent reaches a waypoint, it is marked as completed, indicating the accomplishment of that specific goal. Each agent is rewarded upon reaching a waypoint, and an additional reward is given to all agents if all available waypoints in the environment are visited. This additional reward has a base value for all agents and increases proportionally with each agent's contribution to goal achievement.

Penalties are imposed under certain conditions to encourage safe and efficient navigation. If two agents attempt to move to the same grid cell, a harsh penalty is applied to maintain a safe distance between them. Penalties are also imposed if agents attempt to move outside the grid area, collide with the obstacle, or reverse their last action, discouraging idling and promoting momentum in a particular direction. Time is also a critical factor with penalties applied for the excessive time taken to reach a waypoint, fostering efficient navigation of the environment.

Each agent's observation is represented by a $20 \times 20 \times 6$ tensor that comprises six 20×20 binary matrices. Each matrix represents the following:

1. Agent location channel - the cell containing the agent's state is marked as 1, the rest as 0.
2. Agent previous location channel - the cell containing the agent's previous state is marked as 1, the rest as 0.
3. Obstacle channel - cells containing obstacles are marked as 1, the rest as 0.
4. Friends channel - cells containing other agents' states are marked as 1, the rest as 0.
5. Friends' previous location channel - cells containing other agents' previous states are marked as 1, the rest as 0.
6. Targets channel - target cells are marked as 1, the rest as 0.

These matrices are depicted in Fig.9 for a showcase example.

3.3 Proximal Policy Optimization Algorithm (PPO)

The development of Policy Gradient Methods for RL with Function Approximation [32] marked a significant advancement in RL. These algorithms estimate the policy gradient and subsequently apply a gradient ascent algorithm to this estimate. The gradients are obtained through automatic differentiation software, which operates on a surrogate loss objective. These gradients are then backpropagated through the neural network to update

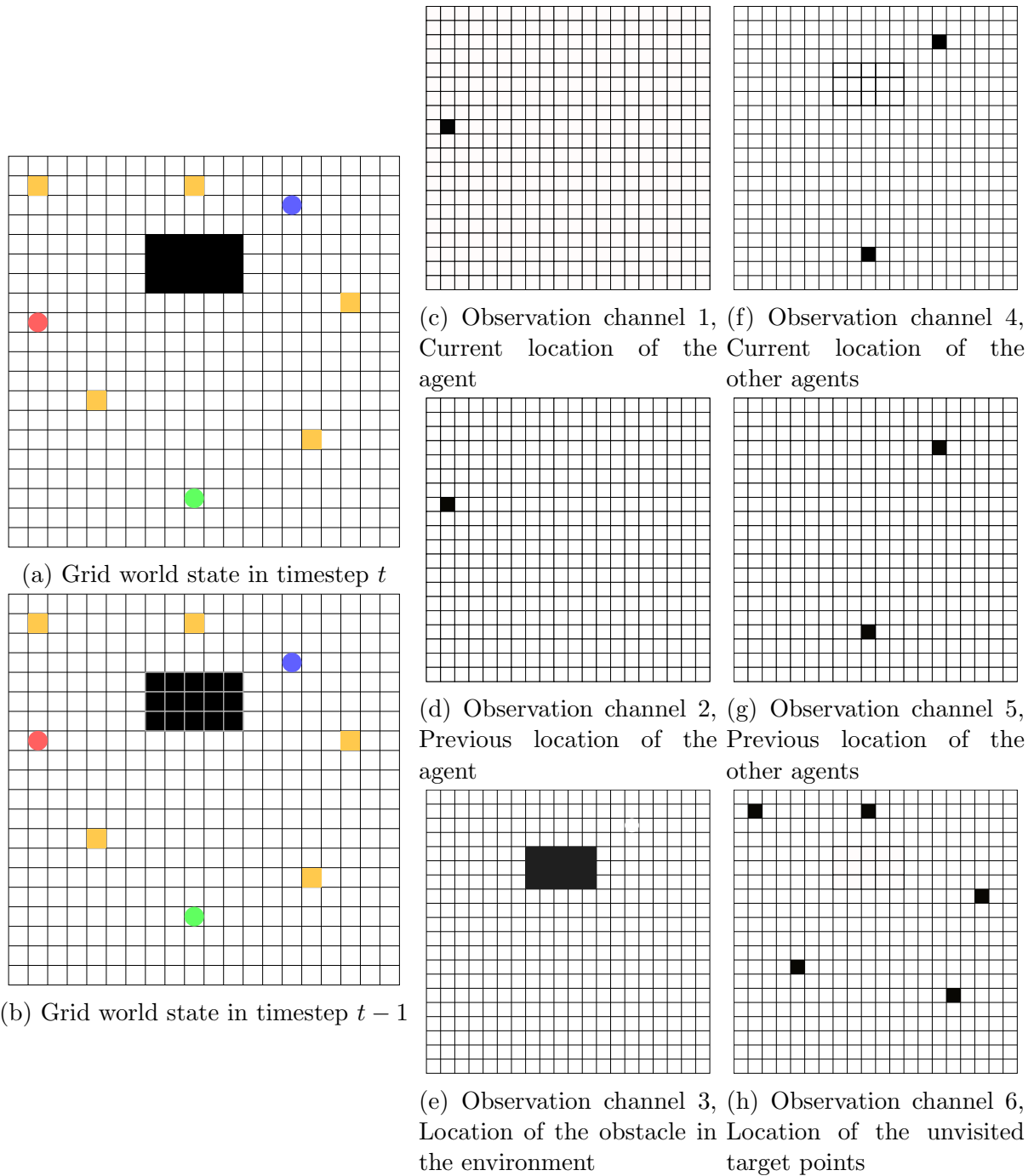


Figure 9: Figures 9a and 9b depict the current and previous states of the environment at timestep t , along with the observations for Agent 1 (represented in red). Obstacles are illustrated in black, target points are marked in yellow, and Agents 1, 2, and 3 are denoted by circles in their respective colours: red for Agent 1, green for Agent 2, and blue for Agent 3.

the parameters. PPO [3] is a method that falls under the category of policy gradient algorithms. It is characterized as a model-free, on-policy, actor-critic, policy-gradient method. The implementation of policy gradient methods involves the construction of an objective function. The gradient of this function serves as the policy gradient estimator,

as shown in Eq.(9) for the policy objective function:

$$L^{PG}(\theta) = \hat{\mathbb{E}}[\log\pi_{\theta}(a_t|s_t)\hat{A}_t] \quad (9)$$

The aim of PPO is to maintain the reliable performance of Trust Region Policy Optimization (TRPO) [33] algorithms, which guarantee gradual improvements by considering the Kullback-Leibler (KL) divergence [34] of policy updates, while only using first-order optimization [35]. The underlying principle of this approach is that by performing a gradient ascent step on this function, the agent is encouraged to take actions that yield higher rewards and avoid actions that could be detrimental. However, this approach has its challenges, particularly in relation to the step size. The training process becomes excessively slow if the step size is too small. Conversely, the training process is highly variable if the step size is too large. PPO addresses this issue by constraining the policy update by introducing a new objective function, the clipped surrogate objective function. This function uses a clip to restrict the policy change within a small range, avoiding extensive, potentially destructive, parameter updates.

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip} \left(r_t(\theta), 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right) \right] \quad (10)$$

Implementing a clipping mechanism on the ratio prevents the policy update from becoming excessively large, as the current policy cannot deviate significantly from the older one, leading to a more stable and efficient learning process.

3.4 Architecture and Training

The PPO agent employs an actor-critic approach. The actor and critic share a similar architecture composed of six primary layers. The architecture begins with a convolution layer with 16 8×8 filters, followed by a convolution layer with 8 4×4 filters. These are followed by a series of fully connected layers that utilize a Rectified Linear Unit (ReLU) [36] activation function. The final layer for the actor consists of five outputs, which undergo a softmax operation. In contrast, the critic's final layer comprises a single output without applying softmax.

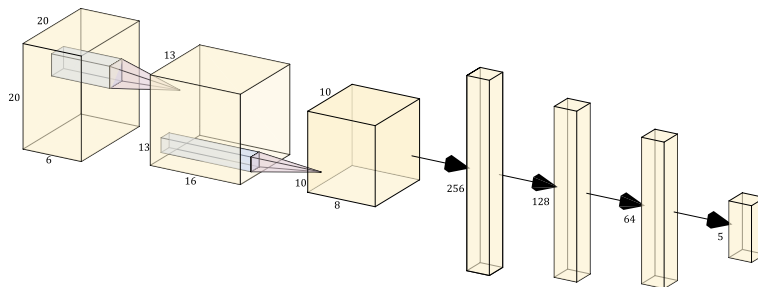


Figure 10: Neural network architecture used for the actor network.

The ADAM optimizer [37] is used during training. The training process is centralized, and experiences are shared amongst the agents. This approach leads to an improvement in the overall efficiency and effectiveness of the training process.

3.5 Results

The performance results of the agents after 1,000 and 4,000 training episodes are presented in Table 1. These results are based on tests against 100 random initial states.

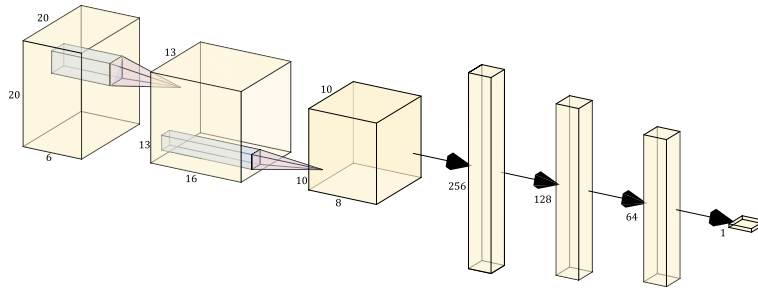


Figure 11: Neural network architecture used for the critic network.

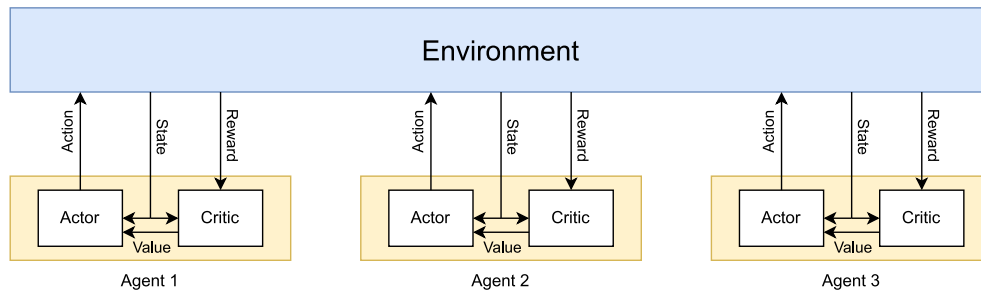


Figure 12: Multi-agent actor-critic system diagram.

The training process demonstrates a significant improvement across all metrics. Notably, agents with more training exhibit a 60% increase in the total number of waypoints visited. The number of tests resulting in visiting all waypoints quadruples, and the average number of time steps required to visit all waypoints is reduced by 23%. The metrics used to evaluate performance are as follows:

- Average Achievement Percentage: This metric represents the average number of waypoints achieved in relation to the total number of available waypoints.
- Total Success Rate: This metric quantifies the number of tests in which the agents visited all waypoints.

Table 1: Performance enhancement with training progress

Number of Training Episodes	Average Achievement Rate	Total Success Rate
1000	48%	11%
4000	77%	49%

4 CONCLUSIONS

In conclusion, an investigation of the application of deep learning in optimizing multi-drone aerial photogrammetry missions has been presented in this paper. The power of neural networks has been harnessed to address several challenges associated with the operation of multiple drones in this field.

The proposed Next Best View planning approach has demonstrated the potential of deep learning in enhancing the efficiency of data collection in aerial photogrammetry. Training neural networks to predict views that yield more information has shown that high-quality reconstructions can be achieved with fewer views that are more strategically selected.

The incorporation of deep learning into multi-agent navigation brings significant benefits. The proposed approach, which utilizes a MAPPO RL algorithm, exhibits considerable potential for further development. It can be adapted to manage tasks traditionally handled by separate subsystems, such as path planning, collision avoidance, and task assignment, in a single comprehensive subsystem.

Overall, this research showcases new possibilities that deep learning can bring to aerial photogrammetry. By leveraging the capabilities of deep learning, data collection processes can be revolutionized, making them more efficient, accurate, and adaptable. These findings are anticipated to inspire further advancements in this domain and contribute to the ongoing progress of aerial photogrammetry.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the National Research Council (NRC) Canada funding under the grant agreement DHGA AI4L-129-2 (CDB #6835) and Mitacs Globalink internship support.

REFERENCES

- [1] M. Tucsok, S. H. Gazani, K. Gupta, and H. Najjaran. 3d reconstruction from 2d images: A two-part autoencoder-like tool. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 538–543. IEEE, 2022.
- [2] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.
- [3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [4] Drone Harmony. <https://www.droneharmony.com/>. Accessed: May 30, 2023.
- [5] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.
- [6] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart. Receding horizon” next-best-view” planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE, 2016.
- [7] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. Towards New Computational Principles for Robotics and Automation’*, pages 146–151. IEEE, 1997.

- [8] B. Hepp, D. Dey, S. N. Sinha, A. Kapoor, N. Joshi, and O. Hilliges. Learn-to-score: Efficient 3d scene exploration by predicting view utility. In *Proceedings of the European conference on computer vision (ECCV)*, pages 437–452, 2018.
- [9] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza. A comparison of volumetric information gain metrics for active 3d object reconstruction. *Autonomous Robots*, 42(2):197–208, 2018.
- [10] M. Maboudi, M. Homaei, S. Song, S. Malihi, M. Saadatseresht, and M. Gerke. A review on viewpoints and path planning for uav-based 3d reconstruction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–26, 2023. doi:10.1109/JSTARS.2023.3276427.
- [11] F. Xue, X. Wu, S. Cai, and J. Wang. Learning multi-view camera relocalization with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11372–11381. IEEE, 2020.
- [12] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.
- [13] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [14] C. Heindl, L. Brunner, S. Zambal, and J. Scharinger. Blendtorch: A real-time, adaptive domain randomization library. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, pages 538–551. Springer, 2021.
- [15] A. Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.
- [16] A. Elmoogy, X. Dong, T. Lu, R. Westendorp, and K. Reddy. Pose-gnn: Camera pose estimation system using graph neural networks. *arXiv preprint arXiv:2103.09435*, 2021.
- [17] X. Li and H. Ling. Gtcar: Graph transformer for camera re-localization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X*, pages 229–246. Springer, 2022.
- [18] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL <http://arxiv.org/abs/1512.03385>.
- [20] M. Fey and J. E. Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

- [21] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.
- [22] E. Salahat, C.-A. Asselineau, J. Coventry, and R. Mahony. Waypoint planning for autonomous aerial inspection of large-scale solar farms. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 763–769. IEEE, 2019.
- [23] R.-J. Wai and A. S. Prasetia. Adaptive neural network control and optimal path planning of uav surveillance system with energy consumption prediction. *Ieee Access*, 7:126137–126153, 2019.
- [24] V. Spurny, V. Pritzl, V. Walter, M. Petrlik, T. Baca, P. Stepan, D. Zaitlik, and M. Saska. Autonomous firefighting inside buildings by an unmanned aerial vehicle. *IEEE Access*, 9:15872–15890, 2021.
- [25] M. Khosravi and H. Pishro-Nik. Unmanned aerial vehicles for package delivery and network coverage. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.
- [26] A. A. Khalil, A. J. Byrne, M. A. Rahman, and M. H. Manshaei. Efficient uav trajectory-planning using economic reinforcement learning. *arXiv preprint arXiv:2103.02676*, 2021.
- [27] J. Westheider, J. Rückin, and M. Popović. Multi-uav adaptive path planning using deep reinforcement learning. *arXiv preprint arXiv:2303.01150*, 2023.
- [28] S. Huang, H. Zhang, and Z. Huang. Multi-uav collision avoidance using multi-agent reinforcement learning with counterfactual credit assignment. *arXiv preprint arXiv:2204.08594*, 2022.
- [29] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert. Multi-uav path planning for wireless data harvesting with deep reinforcement learning. *IEEE Open Journal of the Communications Society*, 2:1171–1187, 2021.
- [30] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang. Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning. *IEEE access*, 7:146264–146272, 2019.
- [31] Y. Chen, Q. Dong, X. Shang, Z. Wu, and J. Wang. Multi-uav autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method. *Drones*, 7(1):10, 2022.
- [32] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.
- [33] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

- [34] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [35] E. Bøhn, E. M. Coates, S. Moe, and T. A. Johansen. Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization. In *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 523–533. IEEE, 2019.
- [36] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [37] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.



COMPUTATIONAL CHALLENGES IN THE MODULAR DESIGN OF FUTURE AIRCRAFT CONCEPTS WITH FLUTTER CONSTRAINTS

Alvaro Cea^{1*}, Peter Nagy², Nicolas Roussouly³, Rafael Palacios^{1*}, Marco Fossati²

1: Department of Aeronautics
Imperial College London
London, SW7 2AZ, United Kingdom

2: Department Mechanical and Aerospace Engineering
University of Strathclyde
Glasgow, G1 1XJ, United Kingdom

3: Institute of Technology IRT Saint-Exupéry
3 Rue Tarfaya, 31400 Toulouse, France

Abstract. *An approach to integrate flutter constraints in an existing environment for aircraft design optimization is discussed, together with a methodology to generate parametric models of varying fidelities that is used to underpin the development. Geometrically nonlinear effects in the aeroelastic response are included in the analysis and the approach blends medium fidelity aerodynamic models with CFD data using a novel infrastructure based on parametric ROMs and sectional polars reconstruction. A strut-braced wing aircraft designed without stability constraints is shown to undergo a strut-driven instability and possible solutions are assessed. Lastly, the developed multidisciplinary tools are deployed in a proof-of-concept of optimization with flutter constraints.*

Keywords: design optimization, aeroelasticity, high-aspect-ratio wings, bi-level MDO formulation

1 INTRODUCTION

The next generation of commercial transport aircraft needs to be capable to respond to the exponentially increasing air traffic volume while enhancing its sustainability and ensuring safety. A step change in aircraft performance is therefore required for which the advancement in ultra-high aspect ratio wings and airframe (UHARW) configurations becomes key. In recent years, some efforts have been put forward to explore the potential of UHARW, including the development of high-aspect ratio concepts based on strut-braced wings (SBWs), which will be studied in this work. The concept of SBW was a common choice in the early years of aircraft design history for low-speed, short-range aircraft. The concept has continued to be used in general aviation aircraft for over a century. However, the use of SBW in long-range, high-speed transport aircraft has not yet been realised, although it was a subject of research since 1980's [1]. The SBW concept offers several synergistic design advantages. The strut reduces the maximum bending moment in the wing by significant amounts, typically up to 50%, which can greatly reduce wing weight. This allows for the increase in the wing span and reduction of the wing thickness and sweep, without increase in structural weight. The higher wing span reduces induced drag and lower sweep helps more laminar flow on the wing using natural and/or hybrid flow control technologies. Early conceptual studies [2] showed an estimation of up to 9% reduction in MTOW and a fuel burn saving of up to 16% for a wide body aircraft. However, the concept introduces the strut as new component, adding to lift, drag, and weight. Therefore, strut bracing creates a complex, multidisciplinary design task that bears the risk of additional wave drag and induced drag caused by the junctions between strut, wing and fuselage [3]. This sustains the common view in the studies of new configurations for the need of Multidisciplinary Design Optimization (MDO) methods. They are designed to explore the interactions between all these components and to study the tradeoffs between competing physical effects, thus enabling design space exploration. In this work we show the early onset of flutter in SBW configurations and hence the must for including it in any aeroelastic MDO; as it has been recently done in [4], where a multifidelity MDO is presented both on a conventional High-Aspect-Ratio Wing (HARW) and a SBW, that includes geometric nonlinearities and flutter constraints. The non-linear flutter calculation was based on the linear DLM method and thus limited to the deformed shape of the FSI calculation. Precisely early work in [5] revealed that critical flutter and loads in the analysis of a SBW would only be discovered when doing the analysis around the nonlinear trim equilibrium. Another methodology and analysis for SBWs has been developed in [6] consisting of geometrically nonlinear FE beams and medium-fidelity aerodynamic tools enhanced with 2D corrections. However, besides the aeroelastic challenges in the design of SBW, the aerodynamic interference between the wing, fuselage, and the strut is a major challenge not tackled with medium-fidelity approaches: the high-fidelity aerodynamic optimization shown in [7] led to a design with a strut producing negative lift force while reducing the overall drag by nearly 15%. This was due to the reduction of both shocks and separation near the wing-strut attachment. Moreover it reflects non-intuitive outcomes of these designs and limits the use of 2D aerodynamic corrections which would not capture critical aerodynamic effects.

Based on this, we look for an approach that captures geometrically nonlinear effects, including linearization around trimmed flight and the subsequent stability; high fidelity

corrections from the full aircraft model; and a flexible architecture that allows different disciplines to be brought together into an MDO process. A wide variety of MDO architectures or formulations have been proposed in last decades, such as monolithic (MDF, IDF, All-At-Once), and distributed (BLISS, ATC, CSSO etc.) [8], involving decomposition of analyses that may be coupled, and decomposition of optimizations into parallel disciplinary optimization tasks coordinated by a system-level optimizer. For large scale industrial problems, the latter ones have been identified as more suitable for a direct use since it preserves a certain autonomy to the disciplines. As a result, a distributed formulation where the optimization is split into sub-optimizations that take advantages of softwares and strategies specificities, is preferred in our case. The proposed approach is based on the bi-level family [9] implemented into the framework GEMSEO, a Generic Engine for Multidisciplinary Scenarios Exploration and Optimization (<https://gemseo.readthedocs.io/en/stable/>) [10] whose the purpose is to decompose the optimization problem in order to efficiently solve it, without having access to the full jacobian matrix including the coupled terms.

The need for a good software architecture in complex MDO problems has been well described in [11]. It is attained herein via a combination of the Common Parametric Aircraft Configuration Schema (CPACS [12], a component-based data definition system for aircraft and has already been used in various MDO-related problems [13–15]) for the model parameterization, GEMSEO for the process implementation, and a modular description of the aircraft object for the consistency of the models.

A variety of tools is employed for the multifidelity analysis of two different SBW models. First, the SU2 suite [16], for the high fidelity aerodynamic analysis of the rigid configuration and to correct the aeroelastic models. Secondly, aeroelastic simulations are carried out with SHARPy [17], which couples a Unsteady Vortex Lattice Method (UVLM) for the aerodynamics, to a geometrically-exact composite beam model (GEBM) for the primary structural components. This approach can capture arbitrarily large wing deformations, within the assumptions of potential-flow theory. Linearization of both structural and aerodynamic equations about nonlinear equilibrium conditions also enable the effect of wing deformations in flutter and other dynamic characteristics of the vehicle [18]. Recently corrections from 3D CFD computations have been introduced [19], which will be further explain herein. Flutter computations are also performed using the Doublet Lattice Method (DLM) [20], implemented in MSc Nastran.

This paper is structured as follows: Sec. 2 presents the developed computational tools, their interactions and workflow. Two different exercises are shown in Sec. 3: first an optimized SBW model that suffers from flutter instability on the strut is described; possible solutions to push the flutter boundary are assessed. Secondly, a new preliminary SBW design is analysed in which it is the wing that becomes unstable but at speed much higher than the flying envelope. This is not the case when the design is optimized, and the investigation to remove the flutter via a set of moving masses is illustrated. We summarise the overall process and main challenges in Sec 4.

2 METHODOLOGY AND TOOLS

In this section the main tools in the developed methodology are described. Equally important the interaction between the tools and the data flow is highlighted. Ultimately the aim is to produce an efficient framework which a) can capture geometric nonlinearities in trimmed flight; b) includes 3D aerodynamic corrections from CFD simulations of the

full aircraft; c) seamlessly integrates in MDO chains. This is laid out next.

2.1 Aircraft aeroelastic response

The full aeroelastic system is formed by coupling a geometrically-exact composite beam model (GEBM) formulation [21] with the Unsteady Vortex Lattice Method (UVLM) [22] and linearising around arbitrary configurations. This has been implemented into the code SHARPy [17], a modular framework for aeroelastic simulations. This aerodynamic panel method solves the inviscid, incompressible flow equations, which can be cast into the form of the Laplace potential equation. Thus it needs a correction to account for 1) thickness effects; 2) viscous forces; 3) compressible forces and nonlinearities in the flow. The aerodynamic corrections are obtained through a data-driven model order reduction tool, from 3D CFD snapshots. This tool enables an efficient prediction of aerodynamic loads at new design points. The correction process implements also a new method to extract sectional polars from 3D solution reconstructions that are then fed into the aeroelastic solvers. This process has been outlined in [19] and is further explained below. Once the (enhanced) static aeroelastic equilibrium is found, a linearization is done and a state-space system is built from which dynamic loads and stability of the configuration can be calculated. The importance of geometrical nonlinear effects on new configurations can only be stated after carrying out the simulations with those features. On SBW configurations, for instance, the interplay between wing and strut can lead to physics not captured by linear analysis. A proof of concept is illustrated in Figure 1 from [23] where a static and dynamic analysis of a very flexible SBW was performed. The main modelling characteristics are highlighted in the picture, i.e. nonlinear beams on the structure elastic axis, bound and shed aerodynamic panels following the structure, and on top of that now we add the corrections from high fidelity data.

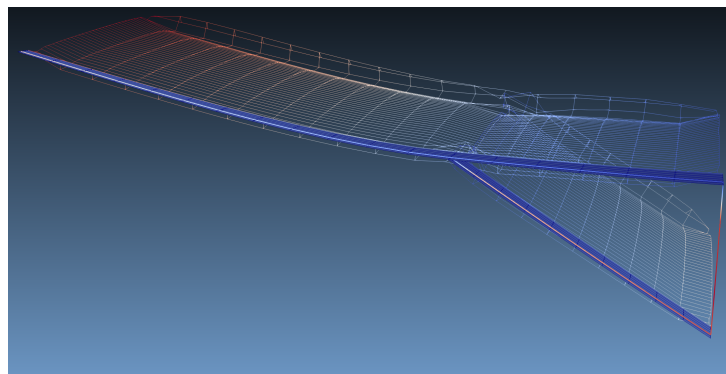


Figure 1: A very flexible SBW undergoing large displacements as design in [23]

2.2 High fidelity aerodynamic corrections

2.2.1 Aerodynamic ROMs

During an MDO exercise, the ROM database(s) can be queried to reconstruct surface solutions corresponding to a target geometry. In this context the reconstructions refer to the geometry of the aircraft – or a specific surface such as the wing or strut – as well as aerodynamic loads (pressure and shear-stress) on the respective surfaces.

Snapshot-based Proper Orthogonal Decomposition (POD) [24] is used to extract bases

of the relevant quantities from an ensemble of steady-state flow solutions (or snapshots). This can be combined with a non-intrusive recovery of new solutions, demonstrated in [25]. In this work approximate solutions at new design points are reconstructed via Radial Basis Function (RBF) interpolation. This data-driven approach is implemented in the code RAZOR.

2.2.2 Interface to aeroelastic solvers

As outlined in section 2.1 corrections are provided to the aeroelastic solvers in the form of sectional polars of aerodynamic coefficients. The polars are provided at spanwise sections corresponding to the nodes of the structural model. Figure 2 shows the correction methodology which is performed every system-level iteration. The outputs of this process are fed to SHARPy which applies the corrections internally.

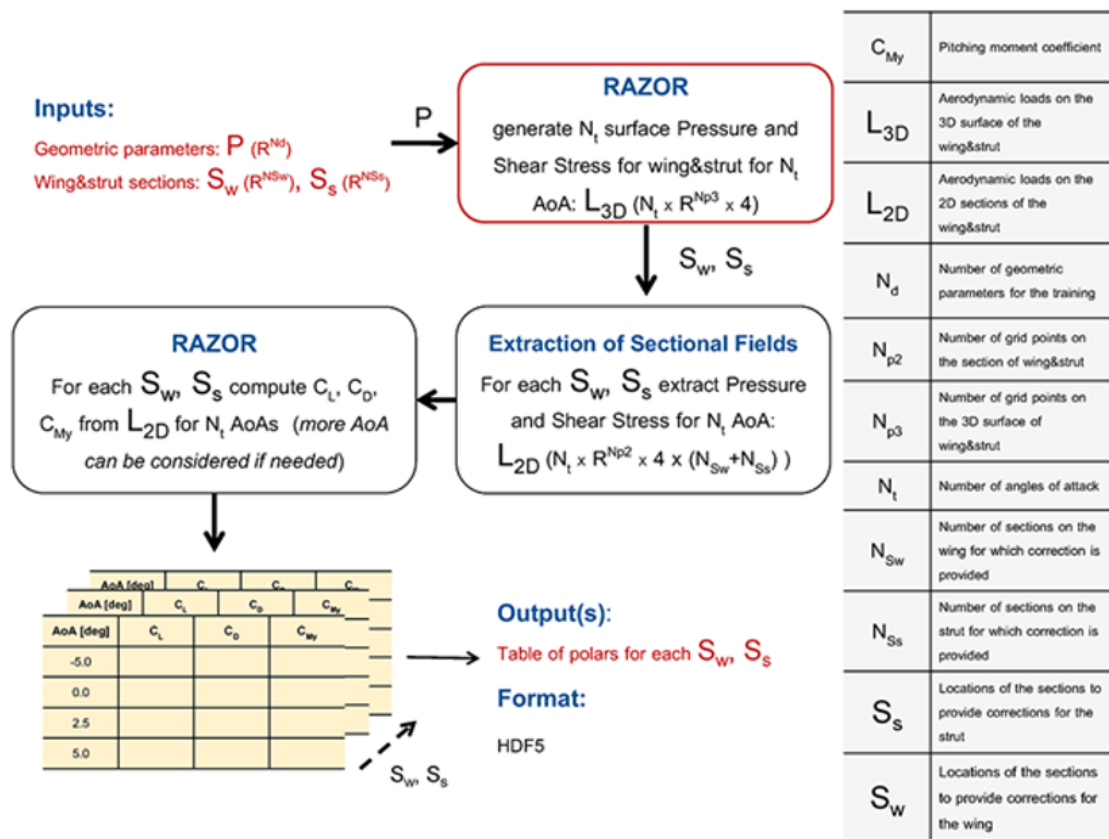


Figure 2: Correction methodology for panels-based aerodynamics

Whenever a new point is queried, platform parameters (denoted as P) are input to RAZOR which computes reconstructions for this target geometry at all angles of attack available in the ROM (N_t), the surface quantities, Pressure and Shear Stress, are recovered along with the new geometry. The set of 3D reconstructions, L_{3D} are then post-processed (via Paraview) to extract sections of the wing and strut components at the locations specified by S_W and S_S . The resulting L_{2D} data (geometry and aerodynamic loads corresponding to each section) is then processed again with RAZOR, so that N_{S_W} and N_{S_S} ROMs are built, with one parameter (AoA) and N_t snapshots each. The solutions are recovered at N_t AoA or can be enriched with more angles. For each 2D solution the

aerodynamic coefficients C_L , C_D , C_{M_y} are computed and assembled into a table of polars which is then passed to the aeroelastic solvers.

2.3 Flutter optimization framework: the bi-level MDO approach

Getting more detailed designs based on higher fidelity models while keeping an affordable computational time is key in the development of new configurations and can be achieved through multidisciplinary optimization (MDO) formulations. It is the final goal in the overall process that we are developing for the current application and for this, a methodology is being built such that different solvers are wrapped into GEMSEO which has been specifically developed to solve MDO problems. The solvers are in charge of computing both static and dynamic aeroelastic responses, also including high fidelity aerodynamic corrections as described in section 2.2 for the dynamic case. Within this environment, the solvers are used through several disciplines and organized in order to provide the necessary outputs depending on the workflow and dataflow.

Even though a lot of progress has been made in the MDO literature to implement coupled gradients between different physical models, it is not always possible to efficiently compute all the coupled terms of the gradients in an intrusive way. It is the case here, as often, regarding static and dynamic responses. For this reason, a bi-level approach developed in [9] is explored that consists in nesting two levels of optimization:

- the system level treats the common design variables that mostly drive the global parameters of the model;
- the local level treats private design variables involved in each discipline.

The overall outputs for the system level are treated through a gradient-free algorithm, and when the dedicated solvers provide the gradients in a faster way (such as via adjoints), it can be taken into account within sub-level optimizations using gradient-based algorithms.

In the current bi-level formulation, we are precisely interested in taking into account the non-linear flutter constraint as described in section 2.1 in addition to any kind of static aeroelastic responses coming from a different solver. A simplified schema of such a bi-level architecture is presented in figure 3 where the system level optimizes the model with respect to the planform variables, the local static aeroelastic scenario optimizes with respect to the structural thicknesses and the airfoil profiles and the local dynamic aeroelastic scenario of interest tries to prevent the flutter to occur through the addition of control masses (values and location). In this process, a first MDA takes place before the local optimizations in order to compute the couplings between the different solvers while the second MDA updates the balance between disciplines from sub-level optimal solutions and returns the objective function and the constraints to the global optimizer. The figure 4 gives a more detailed overview of the process with the eXtended Design Structure Matrix (XDSM) [26]. Note details of potential static aeroelastic solvers are not discussed in this paper as it is not relevant to the problems considered.

The correction based on high fidelity modeling (cf. section 2.2) is included within the first MDA, and is then passed to the solvers involved in the local level. As a consequence, it is assumed that the correction only depends on the planform variables and so remains constant through the local optimization stages. The details of the correction for the flutter is described in section 2.2.

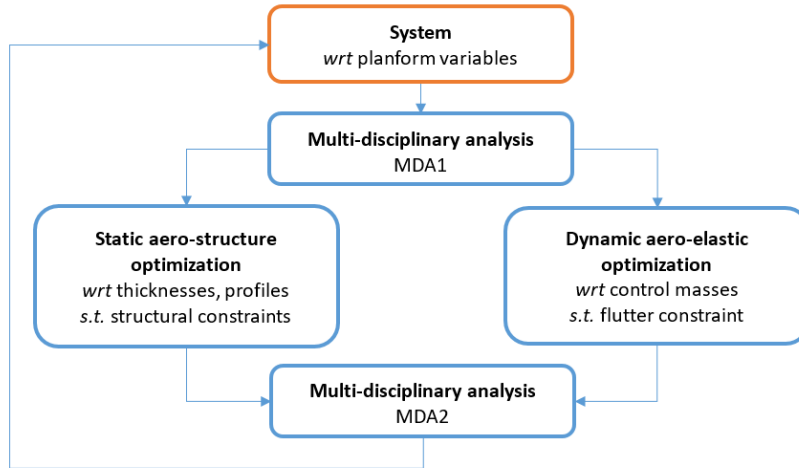


Figure 3: Simplified schema of the bi-level process

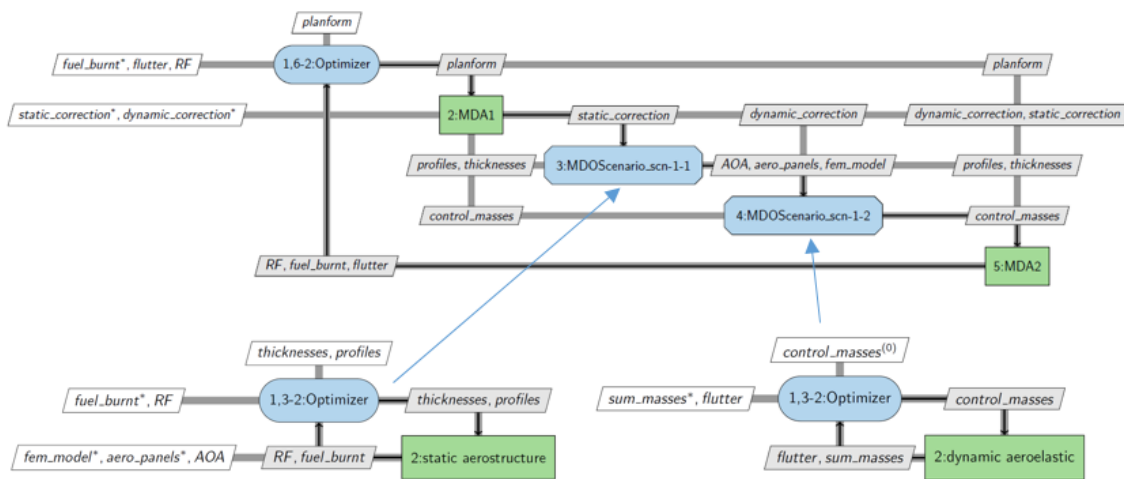


Figure 4: XDSM of the system level MDO process along with the local level scenarios for the static and the dynamic models.

2.4 Modular architectural design

2.4.1 Overview

An initial configuration from a conceptual stage is described in the CPACS format [12]. The link between the CPACS description of the configuration and the process has been done through a dedicated CPACS/GEMSEO interface. While the CPACS is convenient in order to describe the initial and optimized configurations, it is not used to exchange the data which is being passed between the disciplines during the process. For this, a dedicated model handler has been developed that enables to centralize the different parameters and coupling variables which are needed to build the models within the disciplines. Both the CPACS/GEMSEO interface and the model handler enable to build the disciplines for the process through the GEMSEO class denoted as `MDODiscipline`. All those tools are briefly described in next sections.

2.4.2 Model definition in CPACS and interface to GEMSEO

The CPACS description of the model has been considered within the process as an ontology, which means that all the inputs and the outputs of the involved disciplines are defined into the CPACS file. However, the CPACS name space does not natively contain all the inputs and outputs of our required solvers and so it has been extended through an XML Schema Definition (XSD) dedicated to our tools. An example of extended inputs and outputs needed for the SHARPy solver are shown in Figure 5. A dedicated interface

```

<tool uID="sharpy">
  <name>SHARPy</name>
  <version>1.0</version>
  <sharpy xmlns="RHEA">
    <inputs>
      <masses mapType="vector">0;0</masses>
      <massesChordLocation mapType="vector">0.5;0.5</massesChordLocation>
    </inputs>
    <outputs>
      <flutter mapType="vector">0</flutter>
      <totalLumpedMass>0.</totalLumpedMass>
      <cl mapType="vector">0;0;0;0;0;0;0;0;0;0;0;0;0;0;0</cl>
      <slopes mapType="vector">0;0;0;0;0;0;0;0;0;0;0;0;0;0;0</slopes>
    </outputs>
  </sharpy>
</tool>

```

Figure 5: Extended inputs and outputs for the SHARPy solver in CPACS

to GEMSEO, denoted CPACS/GEMSEO, has been developed in order to select any kind of information for the process. More precisely the CPACS file along with the GEMSEO interface has been used for three different purposes:

- to build the design space of the optimizations problems;
- to build the “shell” of the GEMSEO discipline (`MDODiscipline`);
- to generate 3D Computer Aided Designs (CAD) for the 3D high-fidelity CFD computations (offline snapshots).

The figure 6 gives a general overview of the process where the CPACS/GEMSEO tool enables to generate the design space through the class `CPACSDesignSpace` and the process disciplines from `MDODisciplineCPACSBased`, (inherited from GEMSEO class `MDODiscipline`).

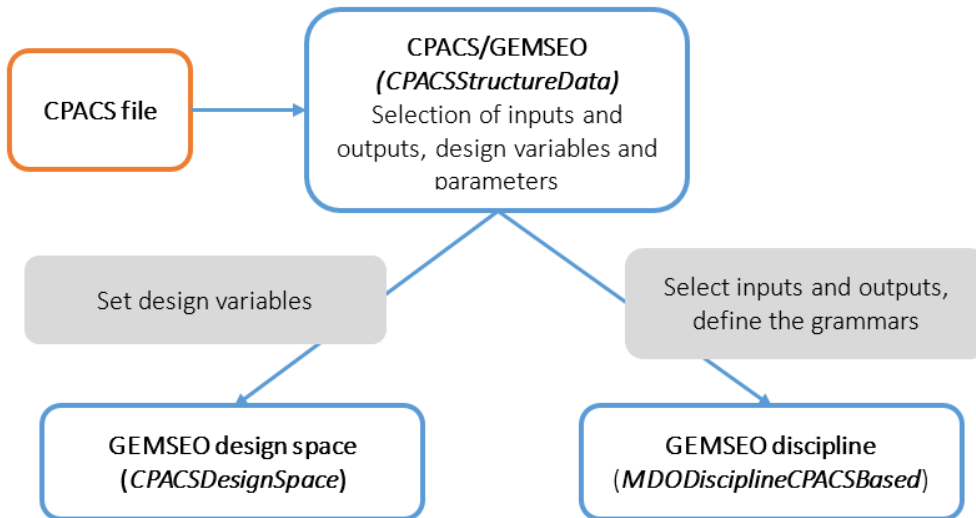


Figure 6: Process schema for the construction of the GEMSEO design space and the process disciplines.

2.4.3 The model handler

The disciplines that stem from CPACS/GEMSEO (`MDODisciplineCPACSBased`) only define an “empty shell” while the inner model for the solvers need to be built. All solver models need to be built from the CPACS data in order to ensure consistency through the process. This model management through the process has thus been delegated to a common shared object which makes the nexus between the solvers and CPACS. This model handler is organized with a component approach where the wing, defined as a part (of the aircraft), is based on components, composed by the different properties needed for the solvers. The figure 7 shows a simplified version of this approach with two components, a wing and a strut. This conforms an abstract structure from which very different solvers can build their inputs and it also plays well with the way models are defined in CPACS.

2.4.4 The disciplines for the process

A discipline involved into the MDO process is represented through the class `MDO-Discipline` in GEMSEO. This class has been derived into an `MDODisciplineCPACSBased` in order to directly build the discipline from the CPACS/GEMSEO toolbox (`CPACS-StructureData`). Once the solver models have been initialized from the model handler, they can be passed to the corresponding disciplines in order to finalize the process set-up. The general approach is depicted in 8.

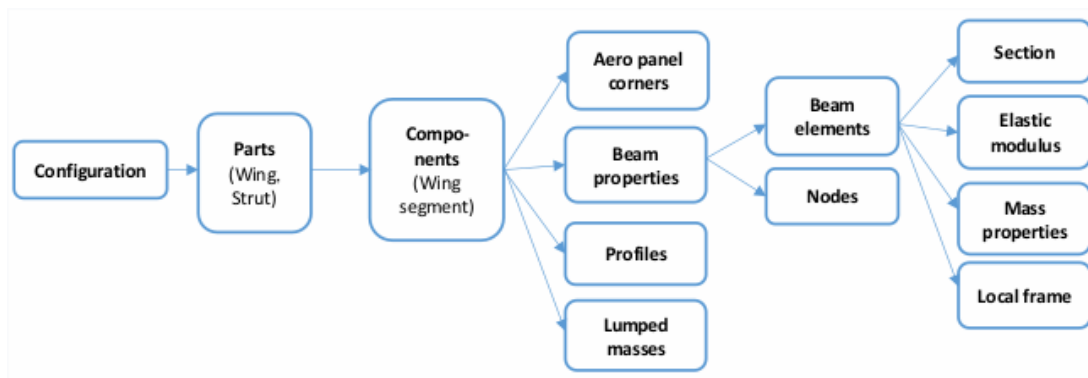


Figure 7: Component-based approach of the model handler

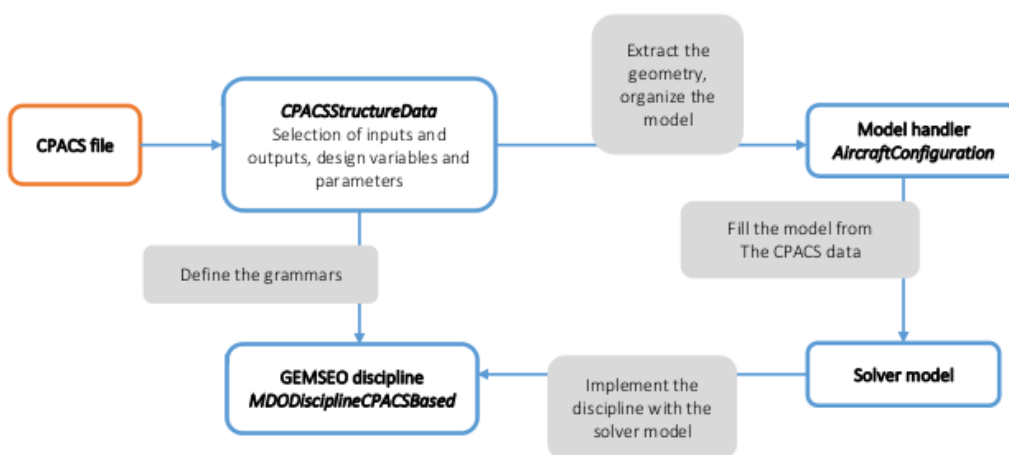


Figure 8: Schema process of the construction of disciplines.

3 RESULTS

We present various analysis performed on SBW designs. The work is divided into two major efforts: firstly we analyzed an already optimized design on which dynamic stability was not taken into account and as a result the strut did flutter; secondly we present the developed tools on a SBW model coming from a conceptual phase.

3.1 Flutter assessment on a SBW designed without stability constraints

Two models from [27, 28] have been combined for the aeroelastic analysis and are overlapped in Fig. 9. The structural model is based on beams to which lifting panels are attached; the high fidelity aerodynamic mesh has been adapted to a mesh for the SU2 suite. For the analysis we will consider one critical point in the structural sizing

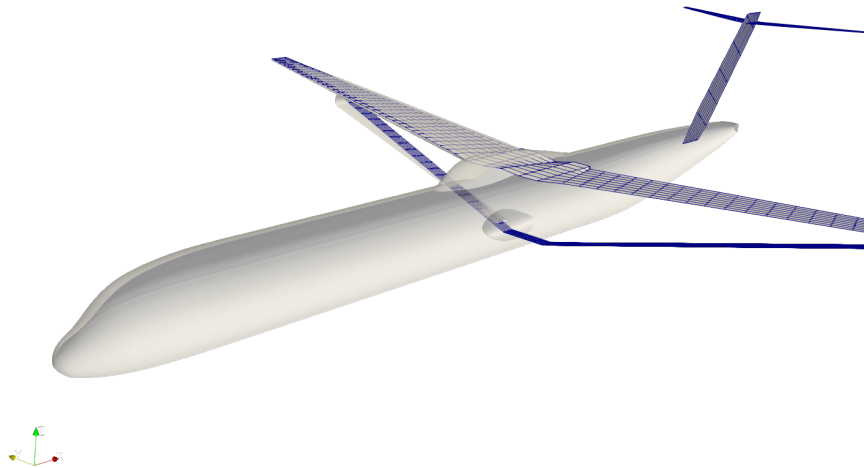


Figure 9: Combined models (CFD mesh and panels representation from structural model)

with flight conditions ($U_\infty = 252.8m/s$ $M_\infty = 0.78$, $Altitude = 7000m$). Since flutter is already encountered at this point, as we will see below, no further exploration was carried across the flight envelope, and the investigation has been instead on the potential design improvements to the aeroelastic stability characteristics. The first ten structural natural frequencies are given in Table 1. Note the existence of symmetric and anti-symmetric modes in the wing with very close frequencies. These are depicted in Fig. 10, where the first five modes are shown.

The panel shapes have been constructed from the beam-model displacements and rotations. They are used in mapping structural deformations onto 3D mesh displacements for the high fidelity analysis: at each panel corner there will be a radial-basis kernel mapping displacements from the panels onto the CFD moving markers (the solid boundaries of wing and strut). With the deformation of wing and strut obtained from interpolation for each modal shape, a mesh deformation scheme within SU2 is launched in order to attain a suitable model for the CFD.

It was found that while convergence of the mesh deformation routines could be achieved without major issues by tweaking the nonlinear steps and the total number of iterations, the resulting mesh would not yield good steady simulations. These problems have been

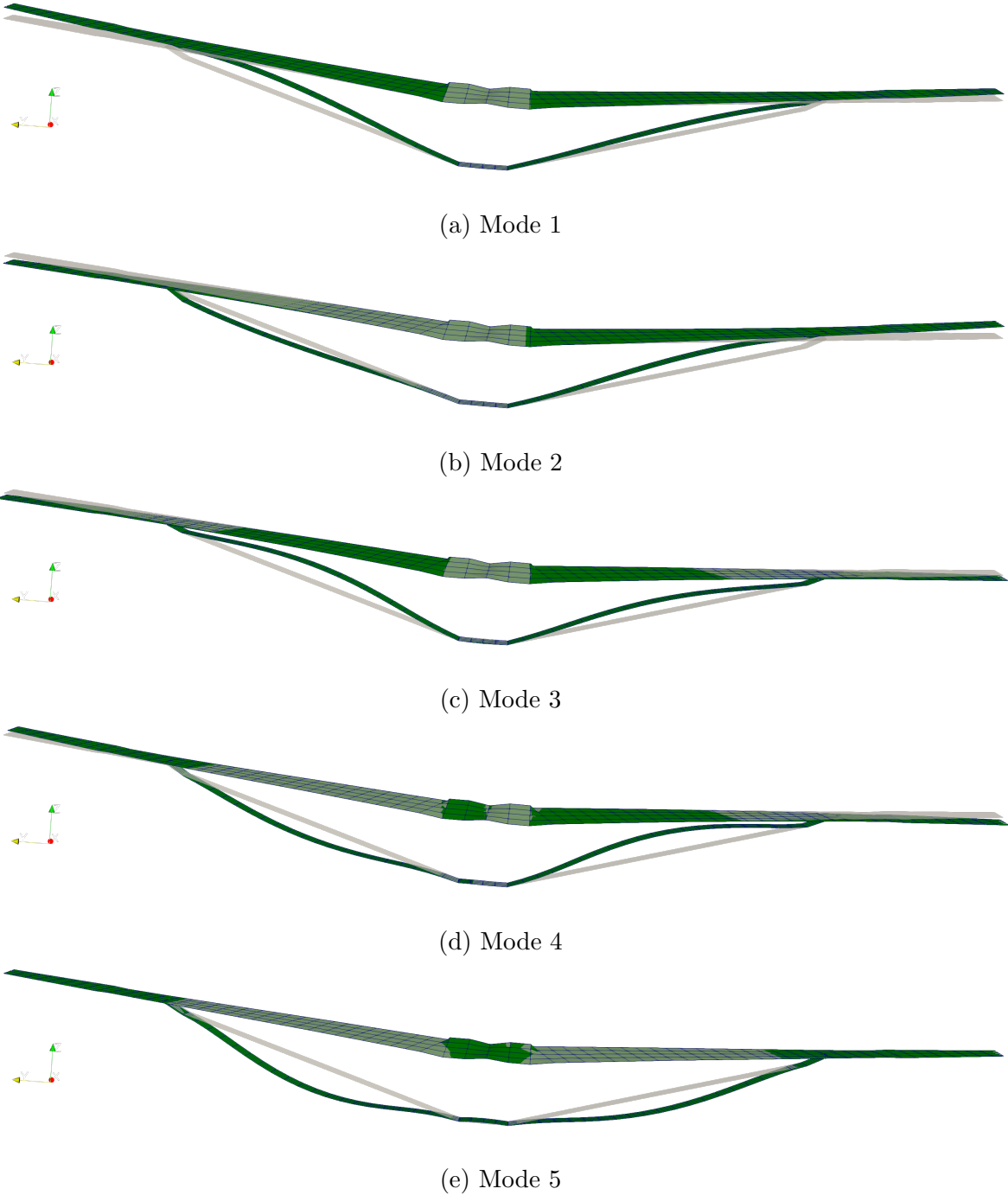


Figure 10: First five structural modes

Table 1: First ten natural frequencies of the wing-fuselage structural model.

Mode	Natural frequency [rads/s]
1	4.79648
2	5.08188
3	10.6141
4	11.4289
5	12.0282
6	12.6861
7	17.4762
8	18.7355
9	23.4287
10	23.5248

previously reported, in [7] for instance, where overset meshes are used to avoid negative volumes. A combination of using an inverse volume mesh deformation scheme together with an extremely low tolerance ($\epsilon \approx 10^{-15}$) and applying a filter at the fuselage connections to bring displacements to virtually zero, finally led to convergent simulations of the steady CFD using an Euler solution. Since the model was provided in Nastran, a DLM-based flutter analysis was carried out first, but with the steady simulations around the modal shapes converged it was expected to move into a solution with higher resolution via the Harmonic Solver recently built into SU2 [29]. Further issues of convergence made us to fall back to the DLM but this time with corrections from the CFD at very low frequencies where convergence could be attained.

3.1.1 Baseline flutter analysis

The flutter analysis on the reference configuration reveals the instability is triggered before the design cruise speed: considering the following flight conditions, $U_\infty = 252.8\text{m/s}$, $M_\infty = 0.78$, $Altitude = 7000\text{m}$, which were set as one corner point of the design envelope, the flutter speed is found at $\approx 170\text{ m/s}$ (33% below cruise speed). The damping and frequency evolution with velocity is results are depicted in Fig. 11. The reduced frequency of flutter can be extracted from the plot and comes at around 0.072. The instability occurs at the interaction between modes 3 and 5. We can also see their anti-symmetric counterparts (modes 4 and 6) becoming unstable. The flutter mechanism is rather unconventional, since the wing bending mode also sets the strut to significant motion, and it interacts with the closest strut-driven mode.

The CFD corrections are introduced via a factor for each modal component which represents the ratio between those values computed by the DLM and the Harmonic Balance. The factor is computed at a frequency close to zero and propagated as a constant to the higher frequencies (considering that flutter occurs at a reduced frequency of only ≈ 0.072 , this approximation should not be too far off). Therefore a set of Generalized Aerodynamic Forces (GAFs) is produced at a selected sampling frequencies, and they are corrected to match the high fidelity data. An in-house P-K solver solves for the flutter speed with these GAFs as inputs. Fig. 12 shows the GAFs after including the corrections. While the GAFs do not significantly vary after adding the CFD solution, the flutter onset has been

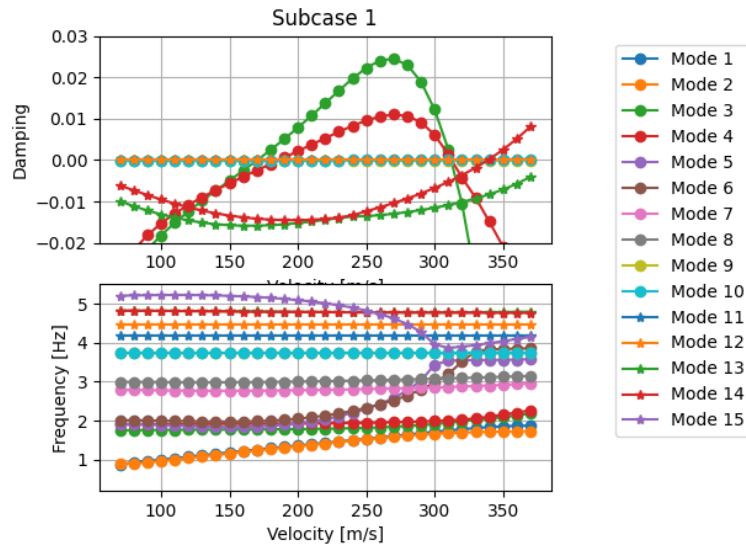


Figure 11: Vg-Vf plots obtained in Nastran for the reference configuration

seen to be very sensitive to modelling details and the additional fidelity brought about by the aerodynamic corrections decrease the flutter speed.

This is seen in the stability plot of Fig. 13. The analysis was carried out with 6 modes to keep resources to a minimum. It is found is that the flutter speed decreases from 173 m/s to 137 m/s, i.e. a 20.9% reduction. This makes even stronger the case about designing this type of aircraft with active stability constraints. It also outlines why flutter margins play a crucial role in the certification process. In the next section we will look at possible solutions to push the stability of the current configuration above design points. The next step in the analysis would be to perform a nonlinear assessment of the flutter speed but since the aircraft is already unstable a parametric analysis under linear assumptions is carried out instead in order to study potential improvements.

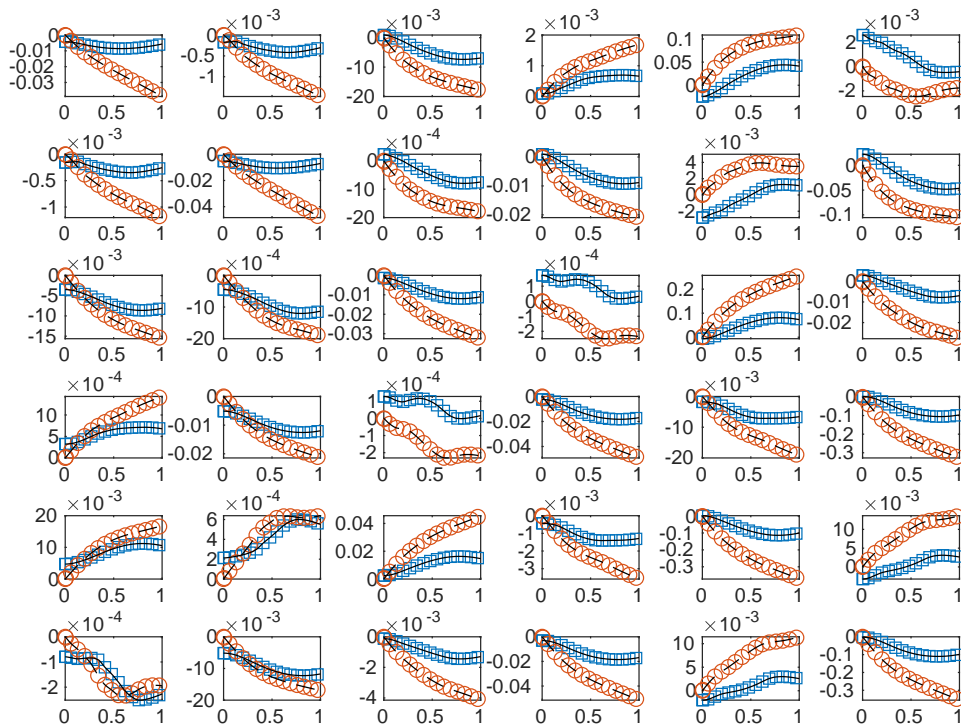


Figure 12: GAFs of the first 6 modes after CFD correction, reduced frequencies from 0 to 1, imaginary (circles) and real (squares) parts.

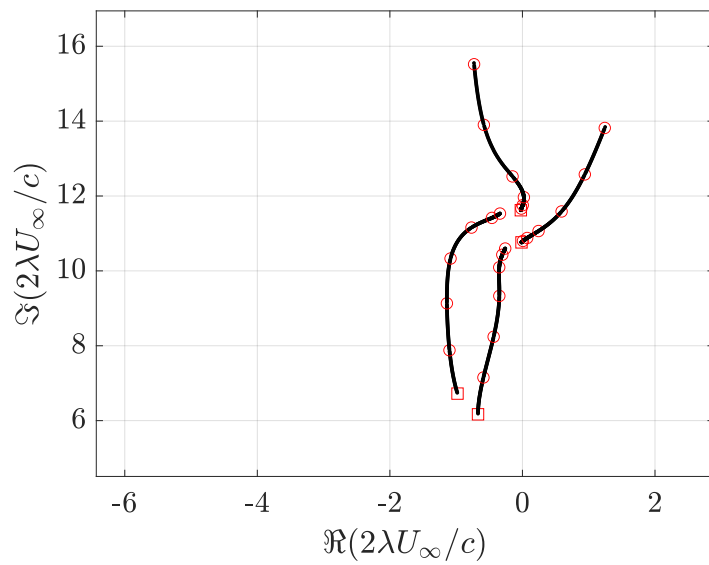


Figure 13: Stability plots for airspeed near the flutter onset after introducing CFD corrections

3.1.2 Parametric analysis for flutter relief

Since the reference design was not designed with flutter constraints and the analysis reveals some issues in the design of the strut, we set up to explore possibilities to enlarge the flutter margins. These are not definitive but can give good insights for future endeavours in SBW configurations. We show the effect of increasing the strut size and of moving strut masses.

Making a bigger strut seemed the first plausible option to improve the stability of the reference configuration. A Factor, α , is applied to the chord dimension along the strut, while the thickness-to-chord ratio of the airfoils is kept constant. The structural model is scaled accordingly, increasing the beam area by α^2 and the moment of inertia by α^4 . In Table 2 we can see that indeed significant improvements can be attained.

Table 2: Flutter assessment when changing the strut chord

flutter	flutter _{mode}	factor
152.665	5	0.85
170.423	3	1
203.342	3	1.1
193.152	10	1.2

As seen in the damping evolution of Fig. 14, a 10% increase in the structural factor α leads to a nearly 20% rise in the flutter. The analysis illustrates that stability and

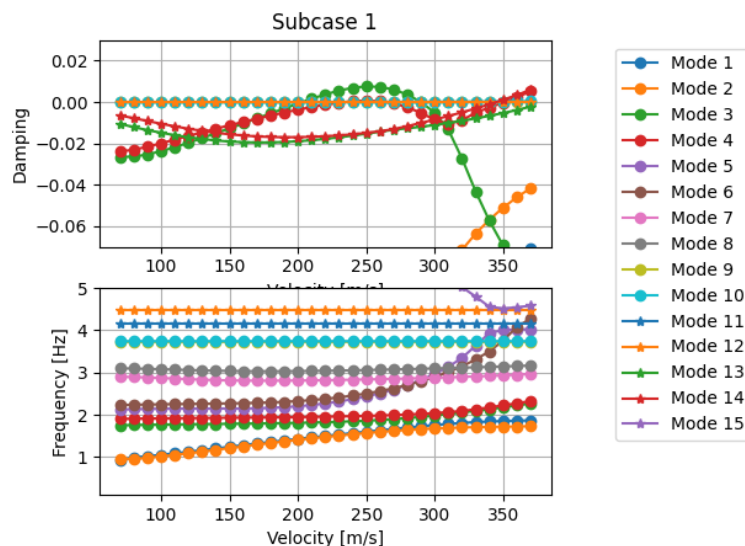


Figure 14: Vg-Vf analysis for the strut extension, $\alpha = 1.1$

strut-size don't correlate monotonically. Thus this can be a good variable to include as part of an optimization problem.

Next we move the lumped masses on the strut together in the chordwise direction. As in classical flutter problems, it proves to be a very effective solution. It should be noted, however, that the ranges in which the masses are moved are likely not feasible due to other physical constraints. Results are shown in Table 3.

Table 3: Flutter assessment when moving strut lumped masses

flutter	flutter _{mode}	factor (% chord)
170.423	3	0
162.767	3	-0.05
156.634	3	-0.1
151.539	3	-0.15
147.136	3	-0.2
180.696	3	0.05
197.31	3	0.1
259.569	14	0.15
213.7	14	0.2

Interestingly, the effect of moving the masses is that mode 3 (and 4, its anti-symmetric) can be made stable, thereby significantly increasing the flutter speed. This is shown in Fig. 15.

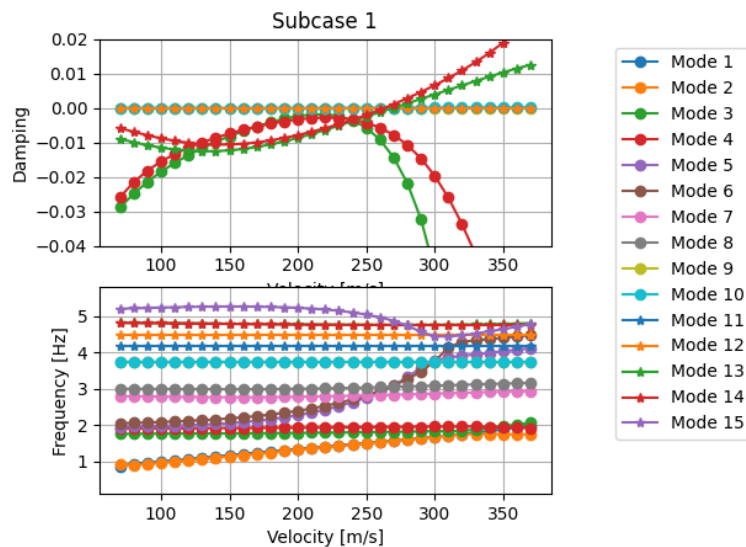


Figure 15: Vg-Vf analysis for moving lumped masses, +0.15% strut chord

Illustrated in Fig. 16, modes 3 and 4 have been damped out, but as modes 13 and 14 (symmetric and anti-symmetric) move down the velocity-axis, the flutter speed is reduced as the masses are further displaced.

These results show the risk of launching the optimization without flutter constraints: the strut turn out to be too small for the stability integrity. We have shown that while redesign to push the stability boundary would be feasible, this would have led to additional costs and also undermine the overall design. In the next section results from a different Strut-Braced Wing design are presented.

3.2 A new SBW design

Having established the need for computational tools that work together in the design phase and account for a broader spectrum of the physics of the problem, we present the progress made in that direction in this section. A CFD campaign was conducted

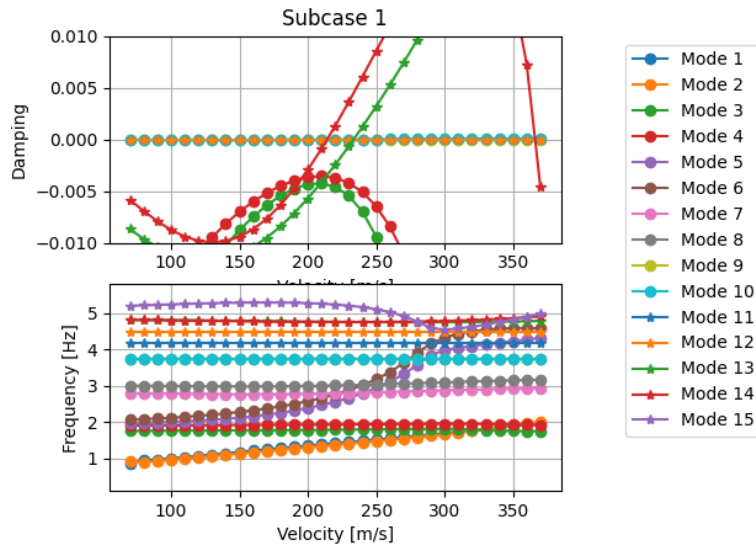


Figure 16: V_g - V_f analysis for moving lumped masses, +0.2% strut chord

around parametric models that are produced as a subset of the design space. A nonlinear flutter analysis on a preliminary unoptimized configuration shows a wing-driven flutter well above the cruise speed.

3.2.1 CFD Campaign and ROM reconstruction

Aerodynamic assessment of a parametric model is carried out first via CFD computations at a set of geometry points given by Design of Experiments (DoEs). After the initial DoE, additional simulations are performed to enrich the sampling of the design space. From the collection of solutions aerodynamic ROMs are constructed from which aeroelastic computations can then be enhanced in an efficient manner. The parameter space is derived from six system-level design variables describing the wing planform, one parameter describing the incidence of the strut, and the freestream angle of attack. The geometric parameter space is shown in Table 4. Design variables describing the airfoil shape of the wing have been omitted with the consideration that minor variations in the airfoil shape would have a marginal effect on the (integral) aerodynamic performance which is the relevant measure for aeroelastic corrections. The incidence angle of the strut is a necessary parameter to attain the requirement that in cruise conditions the strut generates zero-lift.

During the analysis, ROM database(s) can be queried to reconstruct surface solutions corresponding to the target geometry defined at the system level and per iteration. In this context the reconstructions refer to geometry on the wing and strut as well as aerodynamic loads (pressure and shear-stress) on the respective surfaces.

In total 97 distinct wing planforms are considered, including the baseline configuration. For each, a geometry with the minimum and maximum strut incidence is generated, so that a total of 195 geometries are sampled. CFD computations for every geometry are computed at four AoA: -5.0° , 0.0° , 2.5° and 5.0° . Thus the ROMs are constructed from 780 snapshots.

CAD files of the geometries are generated via the GEMSEO/CPACS interface, together

Table 4: Geometric parameters involved in the construction of CFD ROMs

Geometric Parameter	Minimum	Maximum	<i>Baseline</i>
Root chord	2.70 m	3.30 m	<i>2.985 m</i>
Tip chord	0.90 m	1.30 m	<i>1.045 m</i>
Semi-span	28.70 m	33.0 m	<i>31.221 m</i>
Sweep	5.0°	15°	<i>12.5°</i>
Twist at kink	-1.0°	0.0°	<i>-0.5°</i>
Twist at tip	-2.0°	-0.8°	<i>-1.0°</i>
Strut incidence	-3.0°	0.0°	<i>-2.0°</i>

with the Tigl [30] library. After obtaining a mesh for the baseline geometry, the subsequent geometries are meshed in an automated manner. A consistent mesh topology across all snapshots is a requirement for the ROM. Therefore, on the wing and strut surfaces structured meshes are created with consistent number of divisions on both the chordwise and spanwise directions. The rest of the aircraft surface and the volume are meshed with an unstructured algorithm. The surface mesh corresponding to the baseline configuration is illustrated in Figure 17 and in Table 5.

Table 5: Main characteristics of the mesh for the baseline geometry

Number of Elements	Surface elements	Wing surface elements	Strut surface elements
31,078,102	601,258	360,608	69,946

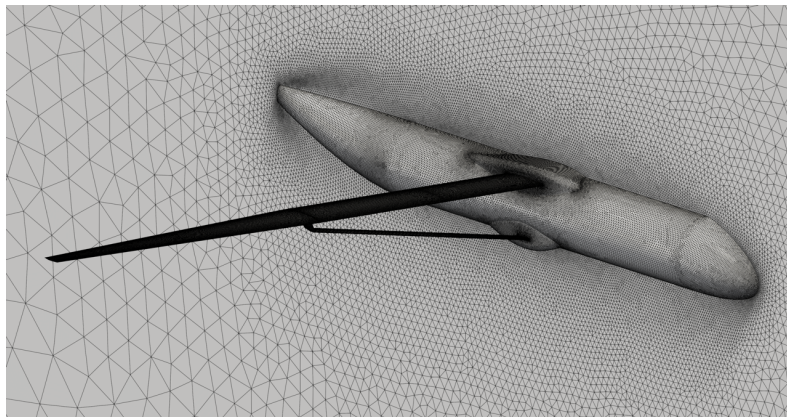


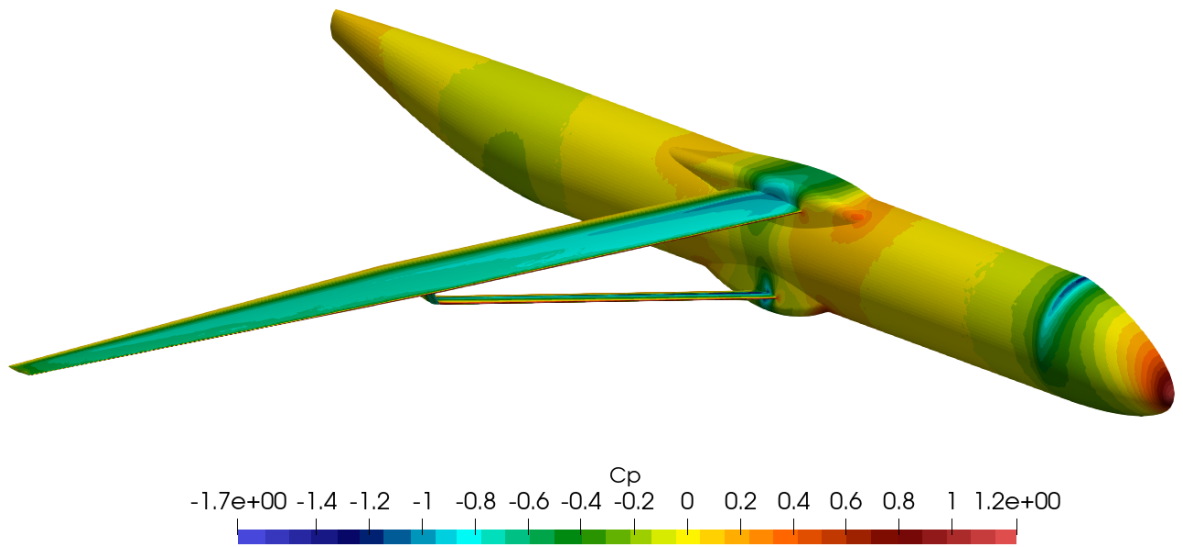
Figure 17: Surface mesh and symmetry plane of baseline configuration.

Steady, Reynolds-averaged Navier-Stokes (RANS) solutions are computed using the CFD solver of the SU2 code. The convective fluxes are solved to second-order accuracy using the Roe scheme with the MUSCL approach. Turbulence is solved to first-order using the Spalar-Allmaras turbulence model. Approximate time to obtain each solution on 144 cores was 8 hours. The fixed freestream conditions are presented in Table 6. Figure 18 shows the C_p and C_f on the aircraft surface, with the main flow features observed for this strut-braced geometry.

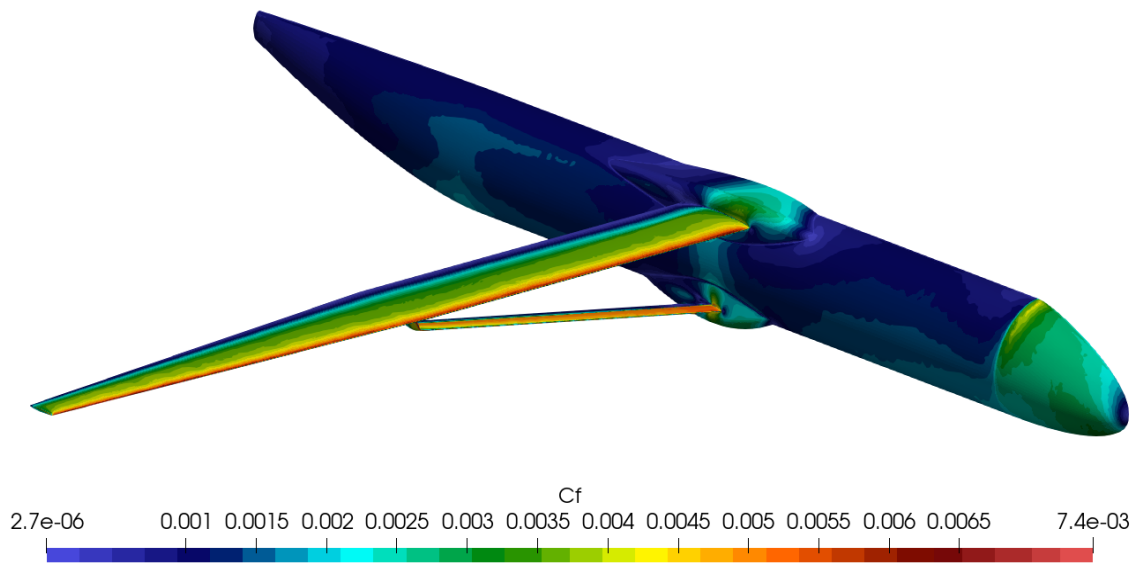
Two ROMs are built, for the wing and strut surfaces respectively. As mentioned before the fields of interest are the geometry and aerodynamic loads, i.e. x , y , and z -coordinates, pressure and shear stress components are reconstructed. As an additional

Table 6: Freestream conditions of the test case

Mach number	Reynolds number	Temperature
0.735	16,659,200	222.7730 K



(a) C_p



(b) C_f

Figure 18: C_p and C_f profiles on CFD solution of the baseline configuration

step, the ROMs are used to identify the cruise AoA for a given wing planform as well as the strut incidence so that the strut contributes zero lift. Figure 19 shows the Pressure and Skin Friction Coefficient contours respectively for the baseline configuration in cruise. It follows from the comparison that most flow features are approximated well but there are clear inaccuracies, most apparent in the reconstruction of shear stress. Figure 20 shows the same comparison on the strut.

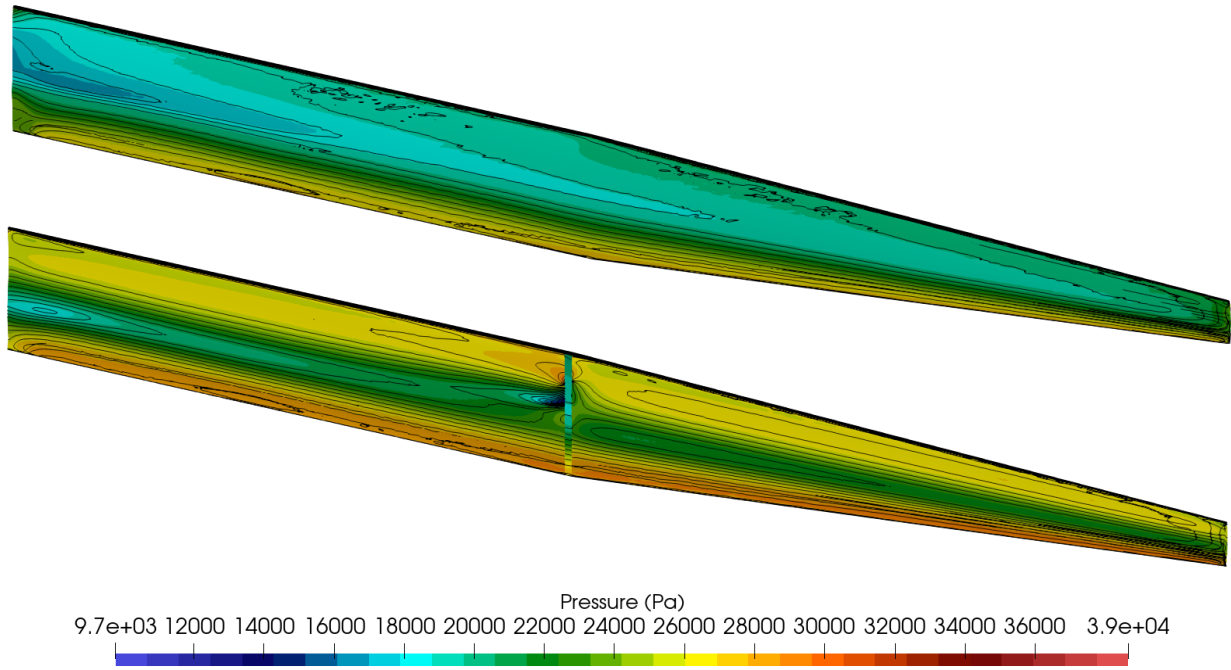
In Table 7 the lift and drag coefficients are compared against reference solutions. The reconstruction of RAZOR shows some discrepancy compared to the prediction, since the strut does generate some lift, as well as the wing generates about 1.77% lower lift than the target. The reference CFD for the same conditions is much closer to the desired cruise condition: the target lift is achieved (with a discrepancy of 0.34%) and the lift generated by the strut is an order of magnitude lower than the reconstruction. In dimensional terms, the strut generates only a marginal 49.5 N lift compared to the approximately 300 kN lift of the wing. The drag coefficients obtained via RAZOR match the reference values well. Note however the very large drag contribution of the strut in comparison to the wing. During the optimisation phase the wave drag of the strut must be reduced substantially by finding good values for the t/c ratio of the strut. In summary, the methodology for finding the cruise condition by leveraging the aerodynamic ROM works well and the models can be used to correct the aerodynamics of the aeroelastic solver.

Table 7: Comparison of coefficients obtained from RAZOR and reference CFD for the baseline configuration in cruise.

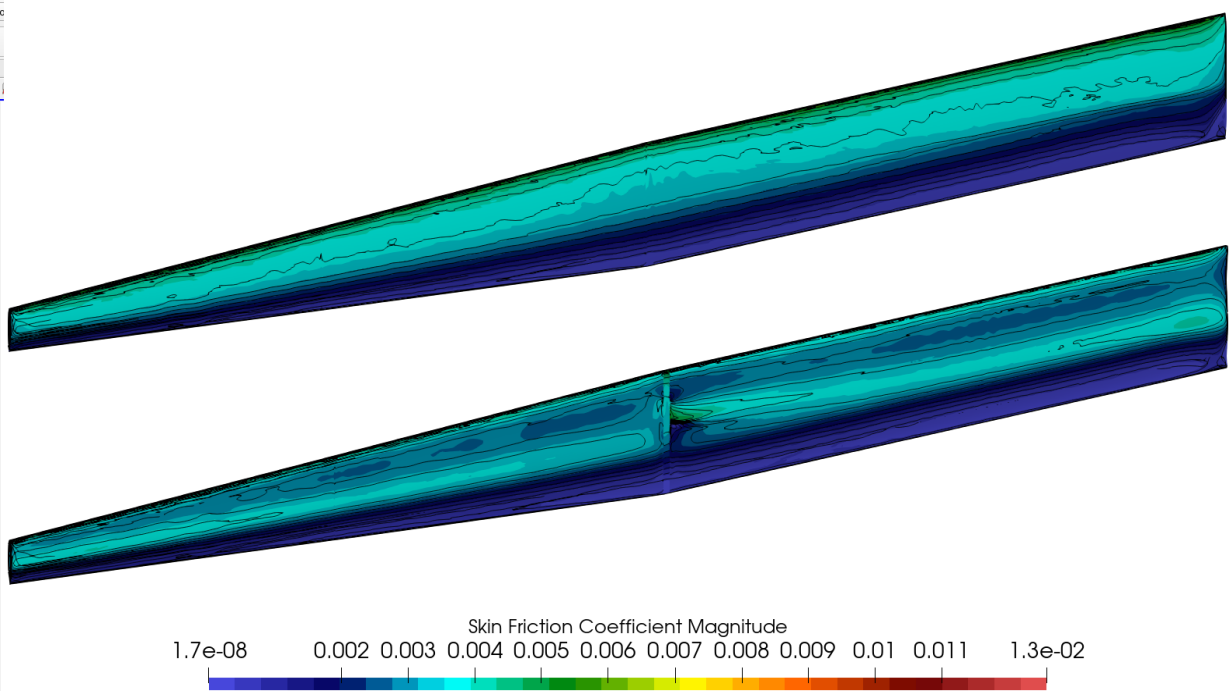
	C_L (RZR)	C_L (CFD)	C_D (RZR)	C_D (CFD)
Wing	0.38016	0.38833	0.011993	0.011909
Strut	0.000410	0.000064	0.008344	0.008030

The solutions are processed according to the aeroelastic correction methodology and the resulting polars are compared. 4 reconstructions and CFD solutions were computed at -5.0° , 0.0° , 2.5° and 5.0° freestream AoA. Plots are shown for sections at the root, midspan and tip locations in Figure 21. The values in general show good agreement, however the 5° case at the wingroot and wingspan shows significant discrepancy.

A. Cea, P. Nagy, N. Roussouly, et al.



(a) C_P



(b) C_f

Figure 19: Comparison of aerodynamic loads on wing obtained from RAZOR reconstruction (black lines) and reference CFD solution (coloured contours) for the baseline configuration in cruise.

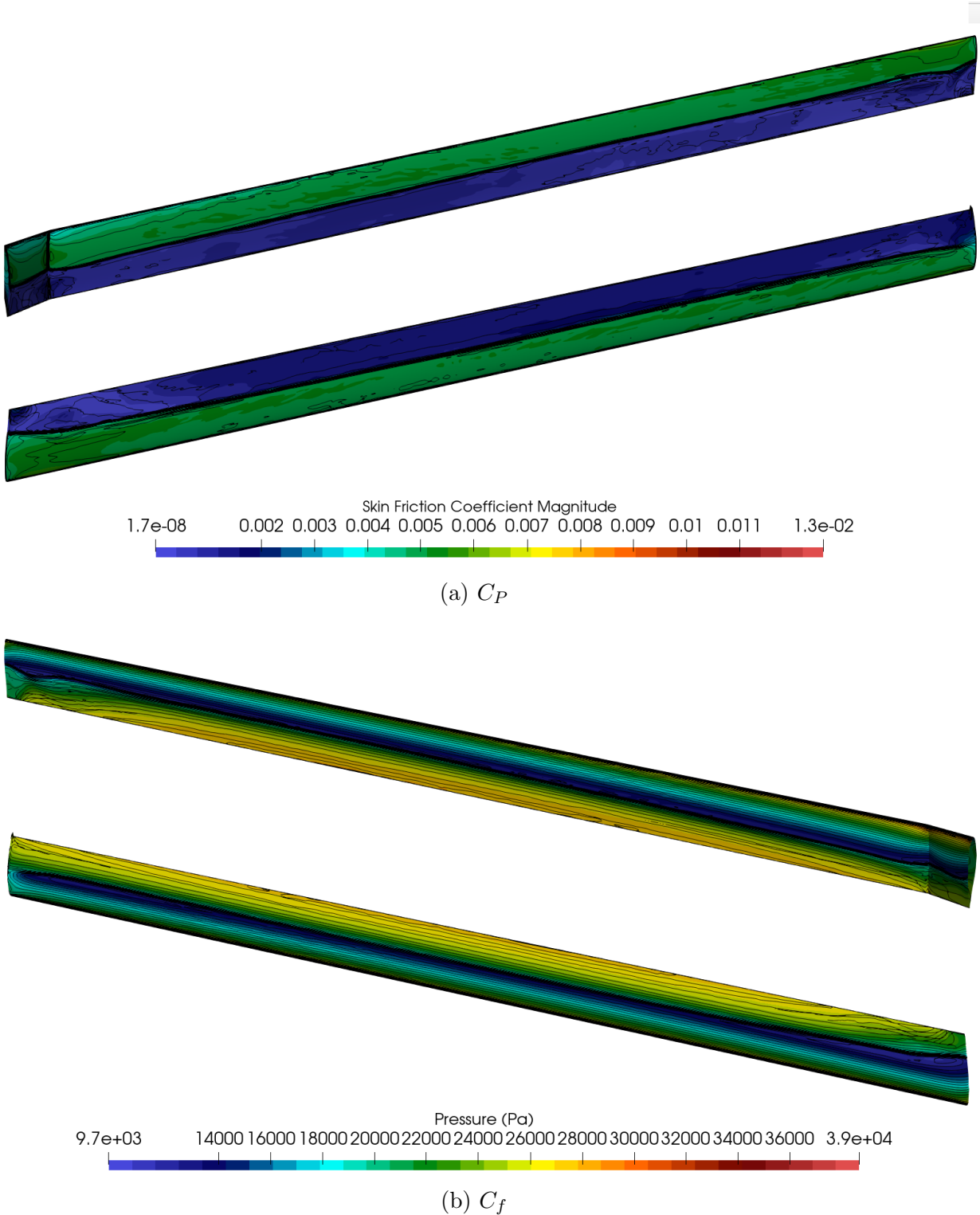


Figure 20: Comparison of aerodynamic loads on strut obtained from RAZOR reconstruction (black lines) and reference CFD solution (coloured contours) for the baseline configuration in cruise.

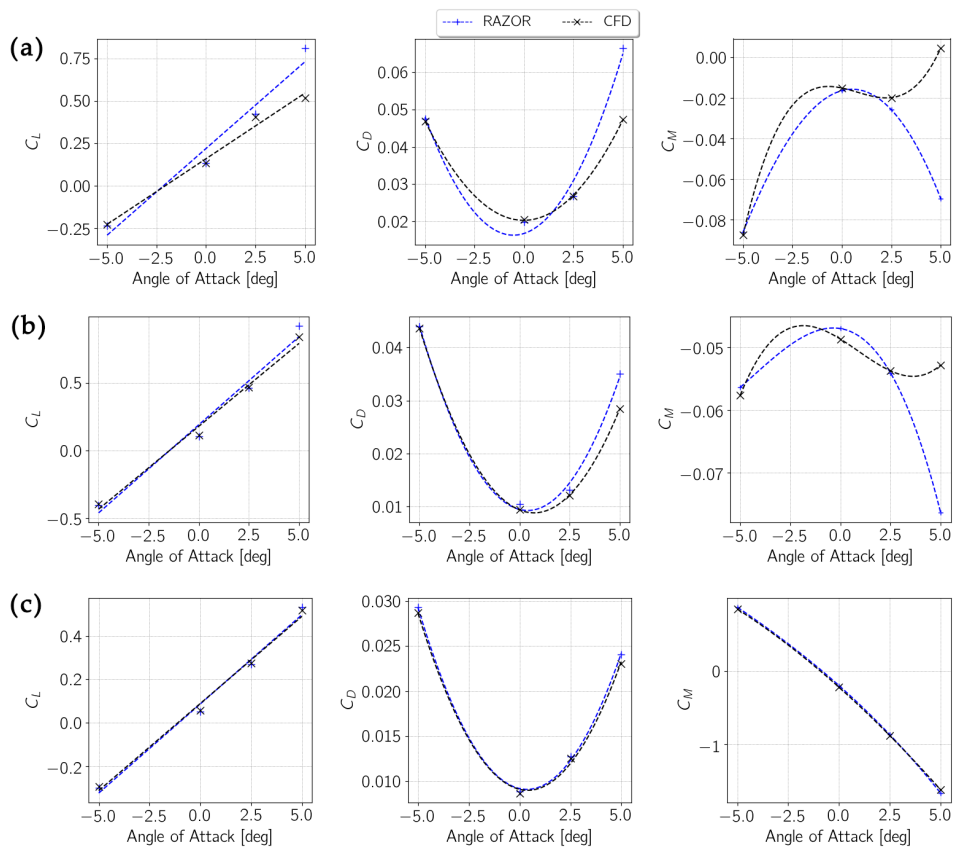


Figure 21: ROMs and CFD polars

3.2.2 Aeroelastic correction process

The general correction process has been presented and validated in [19] where the polars for the whole aircraft rigid model were compared between the pure CFD solution, the ROMs computed by RAZOR and the enhanced SHARPy model. It can be seen that they agree well. The process is summarized in Figure 22. The ROMs in the previous section

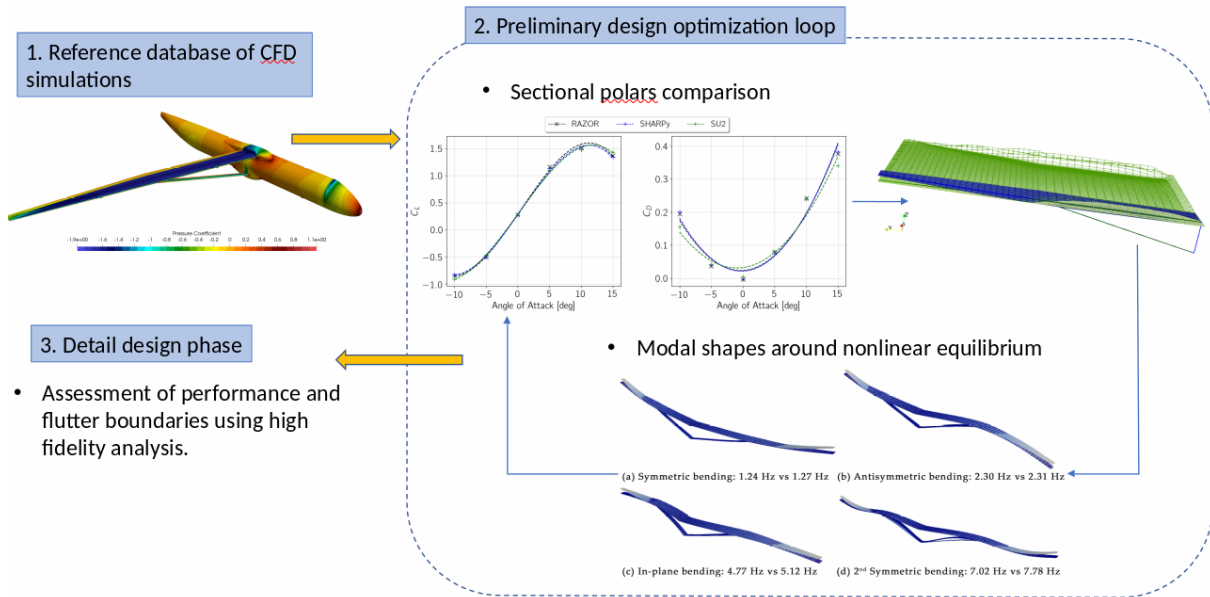


Figure 22: Design process with aeroelastic analysis

are obtained in an offline phase and enter the optimization to enhance the aeroelastic solution, initially on the nonlinear static equilibrium, then on the dynamic state space system produced on the linearized structural model and UVLM aerodynamics. This enhanced LTI system is used to calculate flutter or other dynamic figures of merit. The resulting model of the optimization should then enter a detail phase where the various assumptions and corrections are to be assessed using high fidelity analysis.

3.2.3 Stability preliminary configuration

We set out first to calculate the dynamic stability in a preliminary design that is still to be optimized, to find out that flutter does not occur anywhere close to the cruise speed, which probably indicates an overconservative design. Contrary to linear calculation of the flutter that can be computed directly, the nonlinear flutter entails first the computation of the steady equilibrium for the cruise angle of attack. Two different modelling assumptions are made in which (a) aerodynamic panels of the strut are not considered, and (b) aerodynamic panels in the strut are included, which have a clear effect on the wake behind the junction area. Freestream conditions are those in Sec. 3.2.1.

As seen next, since the flutter mechanism in this model is wing-driven, having aerodynamic on the strut makes a difference but it is not critical in the flutter calculation. The root locus plot for the case with aerodynamics on the strut is shown in Figure 23. The first torsional mode crosses the imaginary axis first, at which point the stability of the wing is compromised. The usual drop in the frequency of the torsional modes is also

observed. The modal shapes and natural frequencies of the wing around the nonlinear

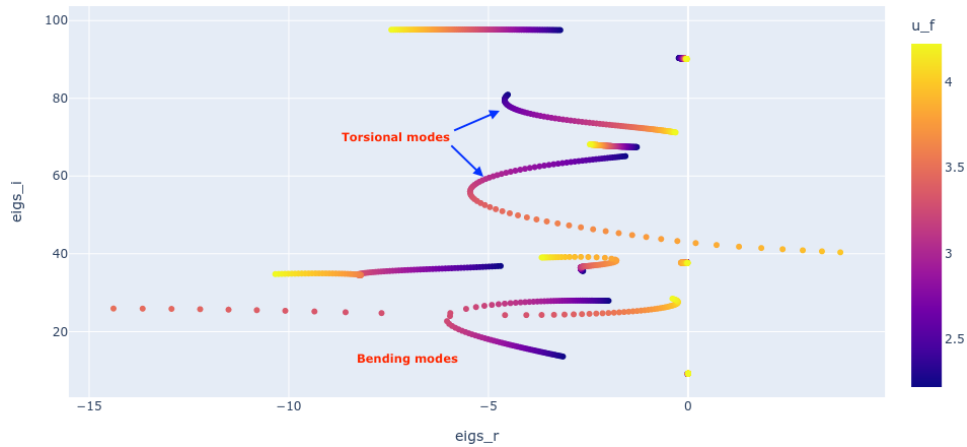


Figure 23: Root locus diagram

equilibrium can be extracted. Figure 24 presents the first out-of-plane (OOP) bending modes, with natural frequencies of 9.4, 27.4, 31.7 rads./s respectively. Similarly, Figure 25 shows the first two torsional modal shapes, whose natural frequencies are 65.2 and 88.6 rads/s. Having pinpointed the modal shapes driving the flutter, an assessment is carried

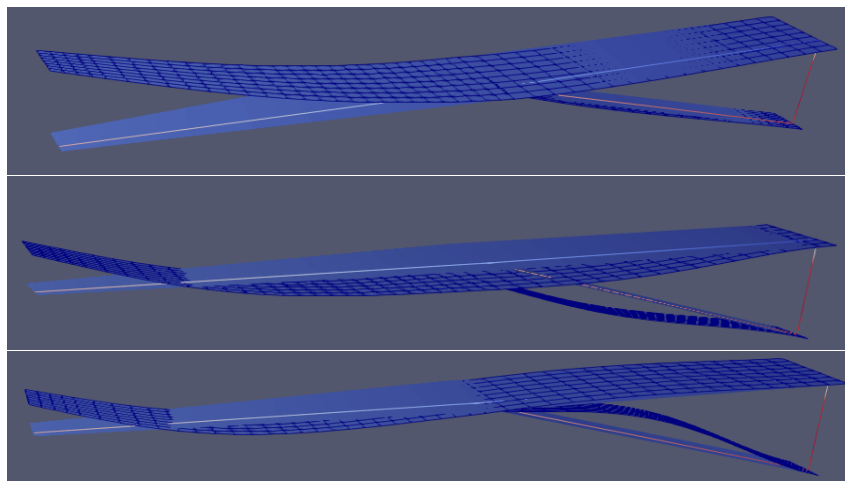


Figure 24: First three OOP bending modes

out with and without aerodynamics on the strut and for a range of Mach numbers. A ratio between the flutter speed and the flying speed is introduced for presenting the results in Figure 26. What the analysis reflects is that the wing is very stiff in torsion. This design needs to be optimized through the MDO process described in section 2.3. The way how the flutter constraint behaves through the local dynamic aeroelastic optimization is shown in the next section.

3.2.4 Optimization of the flutter mechanism

This section shows what happens during a system level iteration of the process described in section 2.3, where the flutter optimization would be executed either sequen-

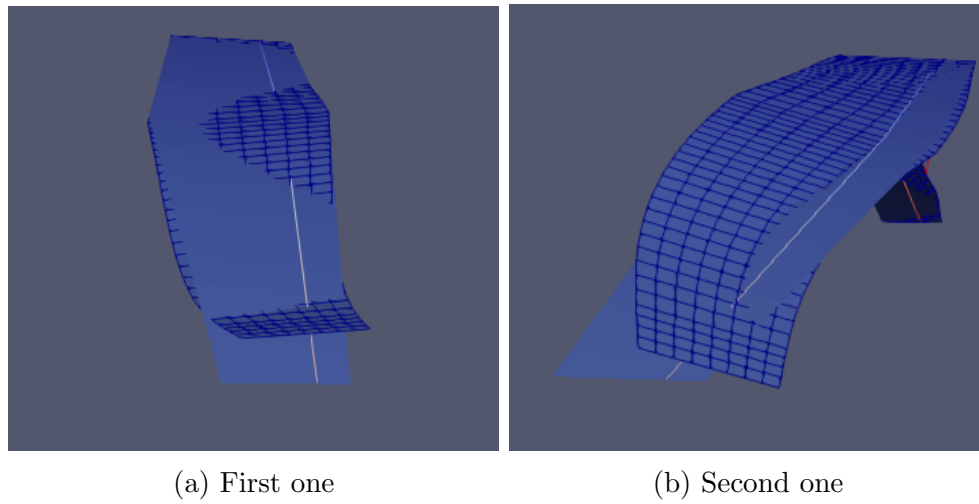


Figure 25: Torsional modes

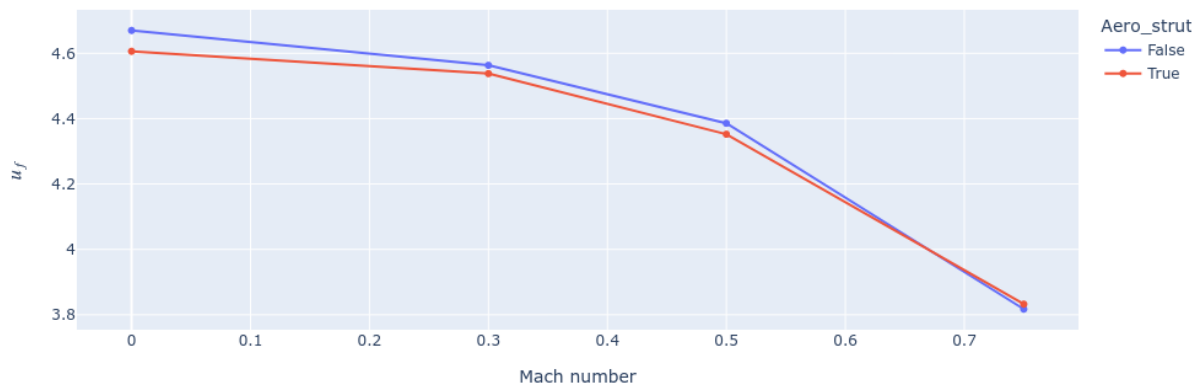


Figure 26: Flutter velocity ratio evolution with Mach number

tially to any static aerostructure optimization, or in parallel to this, from a fixed planform description provided by the system level optimizer.

Conversely to the previous overconservative design (cf. section 3.2.3), we thus start from an assumed optimized design that leads to clear instabilities at the cruise speed within 15% margin. This design is obtained by reducing the structural thicknesses, along with artificially moving secondary masses (engine, tanks, components, etc) slightly backward with respect to the elastic axis of the wing. The conditions are the same than the previous section 3.2.3, and the aerodynamic panels on the strut are taken into account.

Our objective is to vanish the flutter by adding control masses whose values and chordwise locations are optimized. The optimization problem is formulated as

$$m^* = \operatorname{argmin} \left(\frac{\text{lumped_mass}(m)}{\text{lumped_mass}_0} \right) \quad \text{s.t.} \quad \begin{cases} \text{flutter_constraint}(m) \leq 0 \\ m_{low} \leq m \leq m_{up} \end{cases} \quad (1)$$

where $m = [m_{val}^{(i)}, m_{loc}^{(i)}]$ with $i \in (\text{root}, \text{kink}, \text{tip})$ corresponds to mass values and chordwise locations at the root, kink and tip sections of the wing, and `lumped_mass` defines the total lumped masses included into the model (secondary masses plus control masses) with $\text{lumped_mass}_0 = \text{lumped_mass}(m = 0)$. From the vector of design variables m , the total control mass values (resp. locations) are linearly distributed along the wing for each segment root/kink and kink/tip.

The flutter constraint is more precisely formulated as

$$\text{flutter_constraint}(m) = 1 - \frac{\text{flutter_velocity}(m)}{\text{SM} \times \text{cruise_velocity}} \quad (2)$$

where $\text{SM} = 1.15$ defines the safety margin. The value of the flutter constraint at initial design point is 0.055. The optimization problem in equation 1 is solved using the gradient-free algorithm COBYQA [31] which is preferred to a gradient-based approach since the SHARPy solver does not provide any analytical gradient.

The results of the objective and constraint history are presented in figure 28 where the objective is shown through the blue line while the constraint is plotted through the background color (green for feasible and red for non-feasible). It clearly shows that the flutter occurs at the initial point and vanishes throughout iterations. The figure 28 presents the evolution of the design variables, where it can be observed that additional mass is mainly put at kink location, close to the leading edge. This result was well expected since the activation of the flutter was initially induced by moving secondary masses backward. The additional control masses try to balance the instability that may appear with the torsion of the wing. This analysis shows how the flutter can be removed by optimizing control masses within an iteration of the bi-level process described in section 2.3. It must be noted that during the bi-level process, the flutter optimization will be performed only if the flutter occurs, and not at all iterations of the system level scenario.

4 CONCLUSIONS

We have discussed a numerical strategy for integrating flutter constraints in an aircraft design optimization environment. It has required several methodological developments: first, a generic parameterization of the vehicle geometric and structural definition was build to enable automatic generation of models for the different solvers. Second, a bi-level

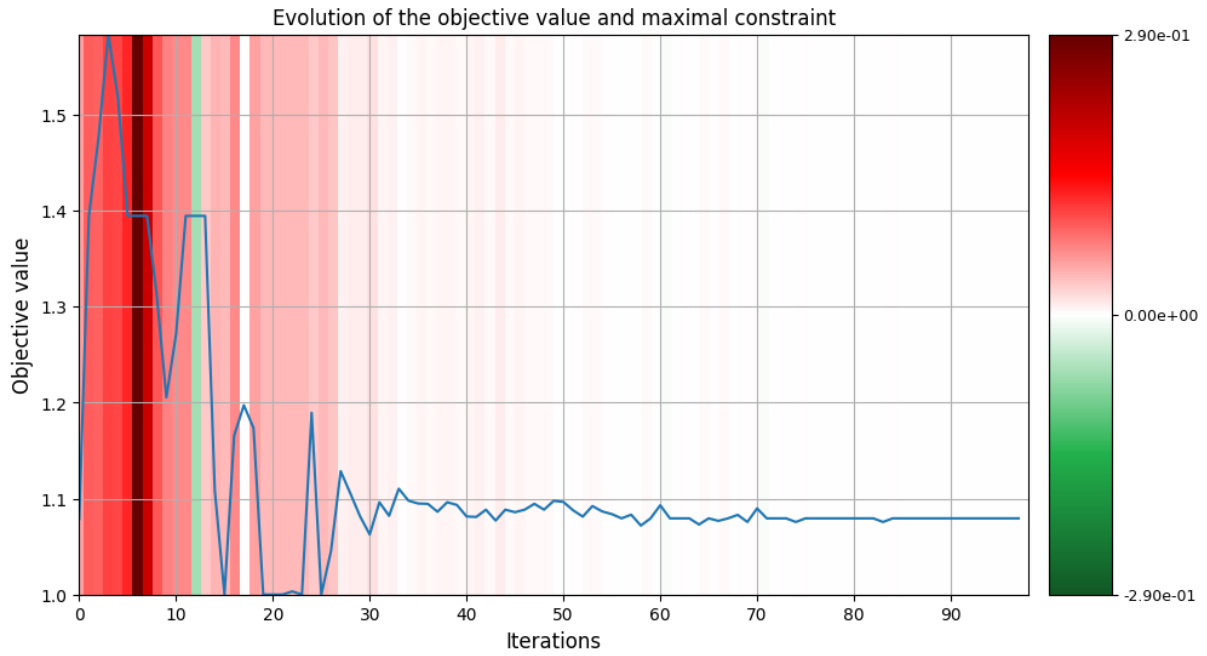


Figure 27: Objective (blue line) and constraint (background color) history of the flutter optimization.

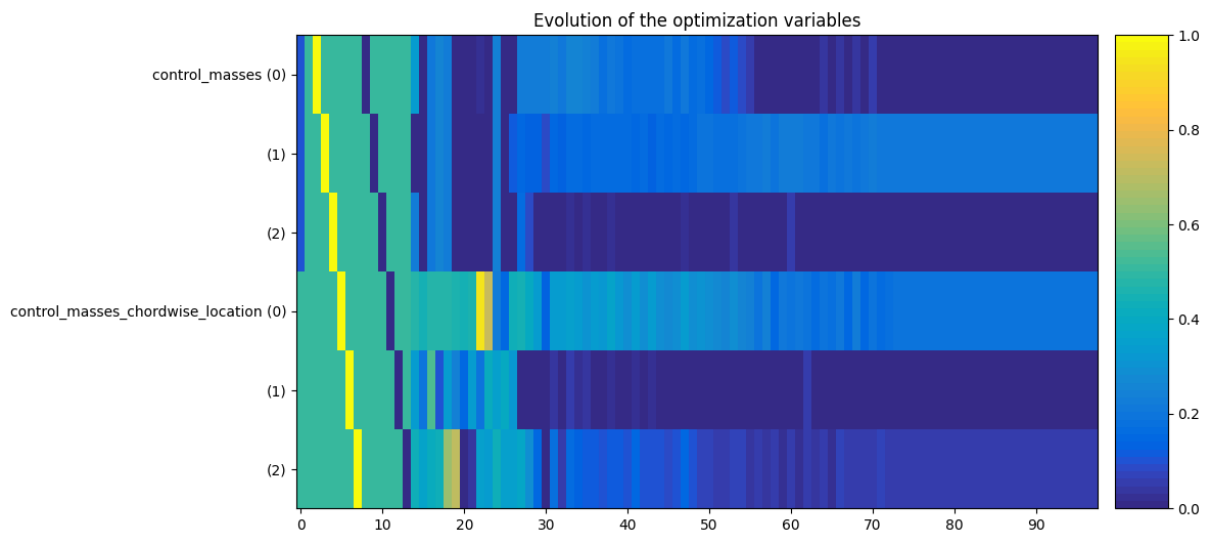


Figure 28: Design variables history of the flutter optimization. Indices (0), (1) and (2) correspond respectively to root, kink and tip position.

optimization strategy has been adopted for the integration of the flutter constraints as an additional optimization loop to any baseline aerostructural optimization software. Finally, for application in the design of transonic wings with very high aspect ratio, sectional polars are reconstructed from a set of parametric reduced-order models obtained from high-fidelity RANS simulations. They are then integrated into the lower fidelity unsteady aerodynamic models. The polar reconstruction process is then exemplified on strut-braced wings, as well as its flutter characteristics, both from parametric studies and preliminary optimization.

ACKNOWLEDGEMENTS

The present work is part of a CleanSky2 project RHEA. RHEA has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 883670. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union. From another Clean Sky 2 project, U-Harward, Arnoult Guillaume and Sergio Ricci provided one of the models studied herein.

REFERENCES

- [1] F. Gern, A. Ko, E. Sulaeman, R. Kapania, W. Mason, B. Grossman, and R. Haftka. Passive load alleviation in the design of a strut-braced wing transonic transport aircraft. In *8th Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, U.S.A., Sept. 2000. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2000-4826.
- [2] J. F. Gundlach, P.-A. Tetrault, F. H. Gern, A. H. Nagshineh-Pour, A. Ko, J. A. Schetz, W. H. Mason, R. K. Kapania, W. H. Mason, B. Grossman, and R. T. Haftka. Conceptual Design Studies of a Strut-Braced Wing Transonic Transport. *Journal of Aircraft*, 37(6):976–983, Nov. 2000. doi:10.2514/2.2724.
- [3] J. Grasmeyer. Multidisciplinary Design Optimization of a transonic strut-braced wing aircraft. In *37th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, U.S.A., 1999. American Institute of Aeronautics and Astronautics. doi:10.2514/6.1999-10.
- [4] M. Sohst, J. Lobo Do Vale, F. Afonso, and A. Suleman. Optimization and comparison of strut-braced and high aspect ratio wing aircraft configurations including flutter analysis with geometric non-linearities. *Aerospace Science and Technology*, 124:107531, May 2022. doi:10.1016/j.ast.2022.107531.
- [5] P. C. Chen and E. Sulaeman. Nonlinear response of aeroservoelastic systems using discrete state-space approach. *AIAA Journal*, 41(9):1658–1666, 2003. doi:10.2514/2.7311.
- [6] J. Fugate, N. Nguyen, and J. Xiong. Aero-structural modeling of the truss-braced wing aircraft using potential method with correction methods for transonic viscous flow and wing-strut interference aerodynamics. *AIAA Aviation 2019 Forum*, (June): 1–27, 2019. doi:10.2514/6.2019-3028.

-
- [7] N. R. Secco and J. R. R. A. Martins. RANS-Based Aerodynamic Shape Optimization of a Strut-Braced Wing with Overset Meshes. *Journal of Aircraft*, 56(1):217–227, Jan. 2019. doi:10.2514/1.C034934.
- [8] J. R. R. A. Martins and J. T. Hwang. Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models. *AIAA Journal*, 51(11):2582–2599, Nov. 2013. doi:10.2514/1.J052184.
- [9] A. Gazaix, F. Gallard, V. Ambert, D. Guénot, M. Hamadi, S. Grihon, P. Sarouille, T. Y. Druot, J. Brézillon, V. Gachelin, et al. Industrial application of an advanced bi-level mdo formulation to aircraft engine pylon optimization. In *AIAA Aviation 2019 Forum*, page 3109, 2019.
- [10] F. Gallard, C. Vanaret, D. Guénot, V. Gachelin, R. Lafage, B. Pauwels, P. J. Barjhoux, and A. Gazaix. GEMS: A Python library for automation of multidisciplinary design optimization process generation. *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2018*, (210049):1–26, 2018. doi:10.2514/6.2018-0657.
- [11] J. R. R. A. Martins, C. Marriage, and N. Tedford. pyMDO: An Object-Oriented Framework for Multidisciplinary Design Optimization. *ACM Transactions on Mathematical Software*, 36(4):1–25, Aug. 2009. doi:10.1145/1555386.1555389.
- [12] M. Alder, E. Moerland, J. Jepsen, and B. Nagel. Recent advances in establishing a common language for aircraft design with cpacs. *Aerospace Europe Conference*, Jan 2020.
- [13] L. Travaglini, S. Ricci, and G. Bindolino. PyPAD: A multidisciplinary framework for preliminary airframe design. *Aircraft Engineering and Aerospace Technology*, 88(5): 649–664, 2016. doi:10.1108/AEAT-02-2015-0061.
- [14] F. Torrigiani, J. N. Walther, R. Bombardieri, R. Cavallaro, and P. D. Ciampa. Flutter sensitivity analysis for wing planform optimization. *International Forum on Aeroelasticity and Structural Dynamics 2019, IFASD 2019*, (June):0–25, 2019.
- [15] T. Klimmek, M. Schulze, M. Abu-Zurayk, C. Ilic, and A. Merle. CPACS-MONA – An independent and in high-fidelity based MDO tasks integrated process for the structural and aeroelastic design of aircraft configurations. 2019.
- [16] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. SU2: An Open-Source Suite for Multiphysics Simulation and Design. *AIAA Journal*, 54(3):828–846, Mar. 2016. doi:10.2514/1.J053813.
- [17] A. Del Carre, A. Muñoz-Simón, N. Goizueta, and R. Palacios. SHARPy: A dynamic aeroelastic simulation toolbox for very flexible aircraft and wind turbines. *Journal of Open Source Software*, 4(44):1885, Dec. 2019. doi:10.21105/joss.01885.
- [18] N. Goizueta, A. Wynn, R. Palacios, A. Drachinsky, and D. E. Raveh. Flutter prediction for a very flexible wing wind tunnel test. *AIAA Scitech 2021 Forum*, (January): 1–17, 2021. doi:10.2514/6.2021-1711.

- [19] P. Nagy, B. Jones, E. Minisci, M. Fossati, A. Cea, R. Palacios, N. Roussouly, and A. Gazaix. Multi-fidelity Nonlinear Aeroelastic Analysis of a Strut-braced Ultra-high Aspect Ratio Wing Configuration. In *AIAA AVIATION 2022 Forum*, Chicago, IL, USA & Virtual, June 2022. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2022-3668.
- [20] E. Albano and W. P. Rodden—. A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. 7(2), 1969. doi:10.2514/3.5086.
- [21] H. Hesse, R. Palacios, and J. Murua. Consistent structural linearization in flexible aircraft dynamics with large rigid-body motion. *AIAA Journal*, 52(3):528–538, 2014. doi:10.2514/1.J052316.
- [22] J. Murua, R. Palacios, and J. M. R. Graham. Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics. *Progress in Aerospace Sciences*, 55:46–72, 2012. doi:10.1016/j.paerosci.2012.06.001.
- [23] A. Cea and R. Palacios. Parametric reduced order models for the aeroelastic design of flexible vehicles. In *AIAA Scitech*. American Institute of Aeronautics and Astronautics, Jan. 2022. doi:10.2514/6.2022-0727.
- [24] L. Sirovich. Turbulence and the dynamics of coherent structures. i - coherent structures. ii - symmetries and transformations. iii - dynamics and scaling. *Quarterly of Applied Mathematics*, 45, 10 1987. doi:10.1090/qam/910463.
- [25] T. Bui-Thanh and M. Damodarany. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. 2003, 06 2003. doi:10.2514/6.2003-4213.
- [26] A. B. Lambe and J. R. R. A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, Aug. 2012. doi:10.1007/s00158-012-0763-y.
- [27] G. G. Carrier, G. Arnoult, N. Fabbiane, J.-S. Schotte, C. David, S. Defoort, E. Benard, and M. Delavenne. Multidisciplinary analysis and design of strut-braced wing concept for medium range aircraft. In *AIAA Scitech*, San Diego, CA & Virtual, Jan. 2022. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2022-0726.
- [28] S. Ricci, N. Paletta, S. Defoort, E. Benard, J. E. Cooper, and P. Barabinot. U-HARWARD: A CS2 EU funded project aiming at the Design of Ultra High Aspect Ratio Wings Aircraft. In *AIAA Scitech*, San Diego, CA & Virtual, Jan. 2022. American Institute of Aeronautics and Astronautics. doi:10.2514/6.2022-0168.
- [29] N. Simiriotis and R. Palacios. A numerical investigation on direct and data-driven flutter prediction methods. *Journal of Fluids and Structures*, 117:103835, Feb. 2023. doi:10.1016/j.jfluidstructs.2023.103835.
- [30] M. Siggel, J. Kleinert, T. Stollenwerk, and R. Maieryl. Tigl - an open source computational geometry library for parametric aircraft design. *CoRR*, abs/1810.10795, 2018. URL <http://arxiv.org/abs/1810.10795>.

- [31] T. M. Ragonneau. Model-based derivative-free optimization methods and software. *arXiv preprint arXiv:2210.12018*, 2022.



INTEGRATION OF A THERMAL MANAGEMENT SYSTEM IN A HYBRID ELECTRIC AIRCRAFT – FUTPRINT50

**Felipe Reyes Barbosa^{1*}, Higor Feltrin Teza², Felipe Issamu Kitadani Odagui²,
Ricardo Gandolfi², Dominik Eisenhut³, Felix Brenner³, Jonas Mangold³, Nicolas
Moebis³, David Bento⁴**

1: Embraer Europe (Airholding S.A.)
Parque Aeronautico de Alverca, 2615-315
felipe.reyes@pt.embraer.com, <https://embraer.com/>

2: Embraer SA
Avenida Brigadeiro Faria Lima, 12227-901, São José dos Campos, São Paulo, Brasil

3: University of Stuttgart
Pfaffenwaldring 31, 70569 Stuttgart, Germany

4: IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, No. 1, 1049-001, Lisboa, Portugal

Abstract *Hybrid-electric aircraft (HEA) are one of the approaches currently under study by the industry to reduce emissions and tackle climate change. The use of batteries and high-power electrical equipment in such aeroplanes poses new challenges that are not usually found in conventional gas-turbine vehicles, such as the necessity of thermal management systems (TMS). In this paper, one of these systems is developed and integrated to the 50 passenger HEA developed in the scope of Futprint50. (maximum 300 words).*

Keywords: Thermal management, hybrid-electric aircraft.

1. INTRODUCTION

According to NASA “Global warming is the long-term heating of Earth’s surface observed since the pre-industrial period (between 1850 and 1900) due to human activities, primarily fossil fuel burning, which increases heat-trapping greenhouse gas levels in Earth’s atmosphere” [1]. It is evident that aviation is a contributor to this phenomenon, as aircraft burn fossil fuel. Up to 3.5% of the anthropogenic climate impact may be caused by aviation [2]. This impact includes, but is not limited to, CO₂ – which is the focus of most research due to its long-term effects. Other combustion products, such as NO_x and water vapour also contribute to global warming.

Given the context above, government, industry (OEMs, airlines, and suppliers) and academia have been directing their efforts to reduce and, eventually, eliminate emissions. The first group’s effort is exemplified by the European Commission’s FlightPath2050, in which the EU’s vision for the aerospace sector through 2050 is reported [3] and by the funding of projects such as Futprint50 – a joint effort between industry and academia to accelerate the entry into service of a hybrid-electric aircraft [4].

The concept aircraft under study, shown in Figure 1, is a 50 PAX, battery-hybrid-electric, based on the ATR42-500’s mission and requirements. Some top-level aircraft requirements (TLARs) are shown in Table 1. These values were obtained from market analyses, which are fully explained in the references [5] [6].



Figure 1: Concept aircraft

Table 1: Selected TLARs [5] [6]

TLAR	FIGURE
Design cruise speed	Mach 0.5-0.55
Design payload	5300 kg
Design range	400 km + reserve
Reserve fuel policy	185 km + 30 min holding
Take-off field length mtom, sl, isa, paved)	≤ 1000 m
Landing field length (sl, isa, paved)	≤ 1000 m
Maximum operating altitude	7629 m (25000 ft)

In addition to the most evident differences of a HEA when compared to a conventional aircraft, such as the use of electric motors and batteries, the adoption of such powertrain creates the need for thermal management: “the ability to manage heat transfer between heat sources and heat sinks to control the temperature of aircraft subsystems/components in order to achieve comfort, safety and efficiency” [7]. In other words, a thermal management system is required to ensure that electronics and batteries are within their temperature operating ranges. In Figure 2 a schematic of the powertrain is shown.

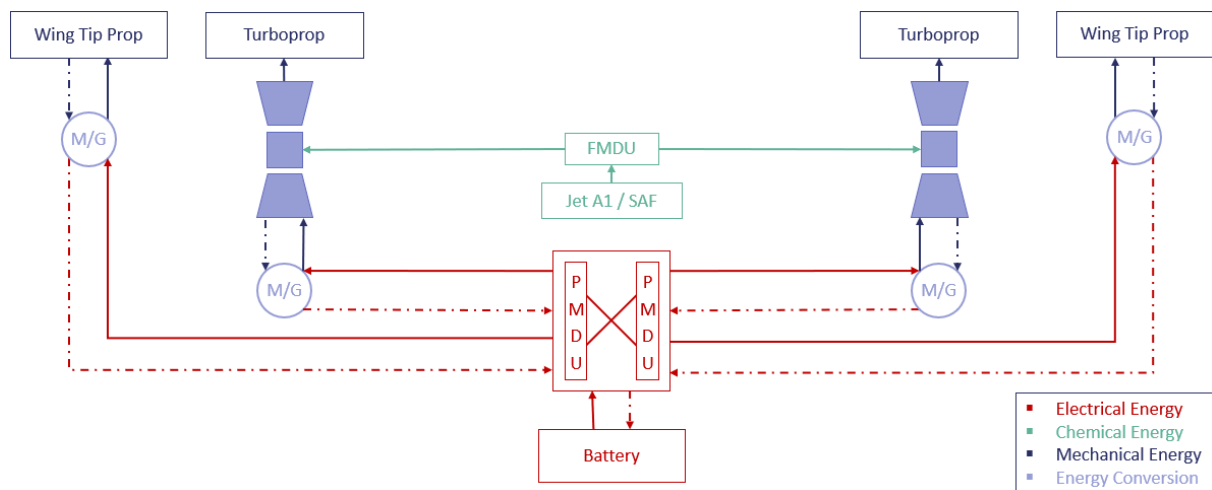


Figure 2: Hybrid electric powertrain

As seen thus far, the conceptual study of an aircraft involves multiple technologies that need to be integrated and have their impact on each other understood – a task that was done in SUAVE (Stanford University Aerospace Vehicle Environment). This is an open-source, multi-fidelity, python-based, conceptual level aircraft design environment developed with the aim of studying new technologies applications to arbitrary aeroplane shapes, i.e. not necessarily conventional “tube and wing” shape [8].

2. METHODOLOGY

2.1 SUAVE Framework

Changes to SUAVE were done in order to adapt it for Futprint’s goal: to design a hybrid-electric aircraft. These are mainly residuals in the iteration processes (for further information reference [9] should be consulted) but the overall sizing process is shown in Figure 3.

The design process begins with definition of TLARs and fixed parameters followed by vehicle definition (aeroplane’s geometry is defined). Then, mass may be estimated by different methods, such as specific power. As the power is still to be calculated, iterations are necessary to adjust the component’s mass. Finally, the mission is solved based on all the previous information and the “synergies and interaction”, that represents the balance of forces and power requirements [9].

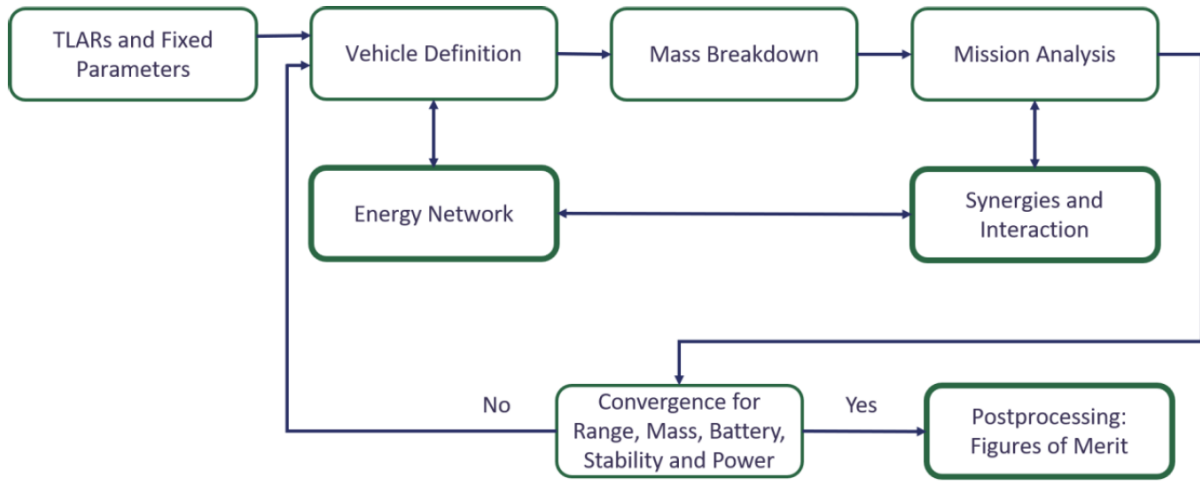


Figure 3: Sizing loop in SUAVE [9]

An important part of SUAVE is the energy network class. This is where all the energy “consumers” and “generators” are mathematically linked at code level. As in Figure 2, all the conventional components are present, but now in conjunction with electrical equipment. The main outputs of this class are the thrust vector and fuel mass flow rate.

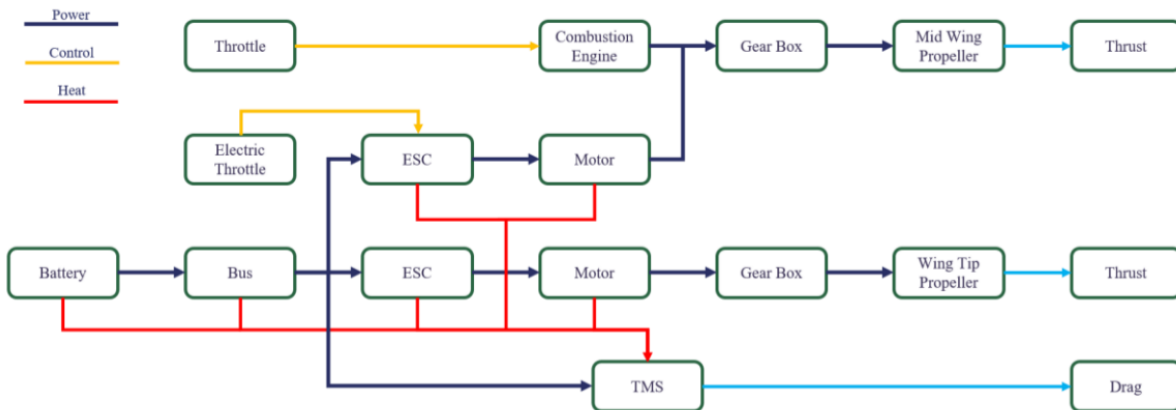


Figure 4: Representation of energy network in SUAVE

2.2 Thermal Management System (TMS)

To better understand the influence of a thermal management system and how it physically integrates to the powertrain, Figure 5 and Table 2 are discussed. The schematic shows the different components that require cooling (blue boxes) and their respective cooling loops (yellow, CL, boxes). The table, on the other hand, shows the different cooling requirements for the components shown in the figure. A careful analysis shows that not only the requirements are significantly different but also that some level of system segregation is required – this is why four subsystems were designed (CL 1 through 4).

First A. Author, Second B. Author and Third C. Author

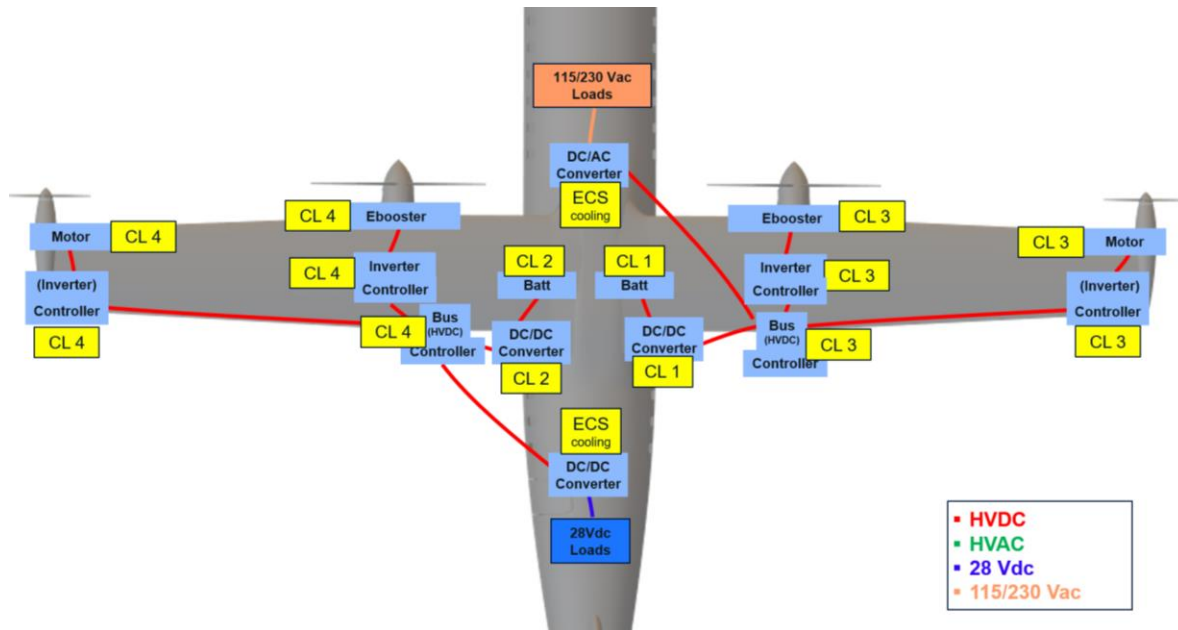


Figure 5: Electrical powertrain and TMS

Table 2: Thermal requirements for electrical components

COMPONENT	OPERATING TEMPERATURE [10]	EFFICIENCY [10]	COOLING METHODS [10]
Batteries	25°C (nominal)	92-96%	Convection, cold plate, dielectric oil immersion [11], tab cooling [12]
Inverters/Converters	-20 to +75°C	97-99%	Convection, cold plate, cryogenic
Electric Motors	-30 to +150°C	95-99%	Liquid, cryogenic

With the information discussed above and considering the mission profile, Table 3 was created. It shows the different cooling loops, the main component they must cool and the thermodynamic cycle chosen to cool it.

Table 3: Relationship between cooling loops, components to be cooled and thermodynamic cycle (see Figure 5)

COOLING LOOP	COMPONENTS TO COOL	CYCLE
CL 1 & CL 2	Batteries	Liquid Cooling (LCS) + Vapour Compression (VCS)

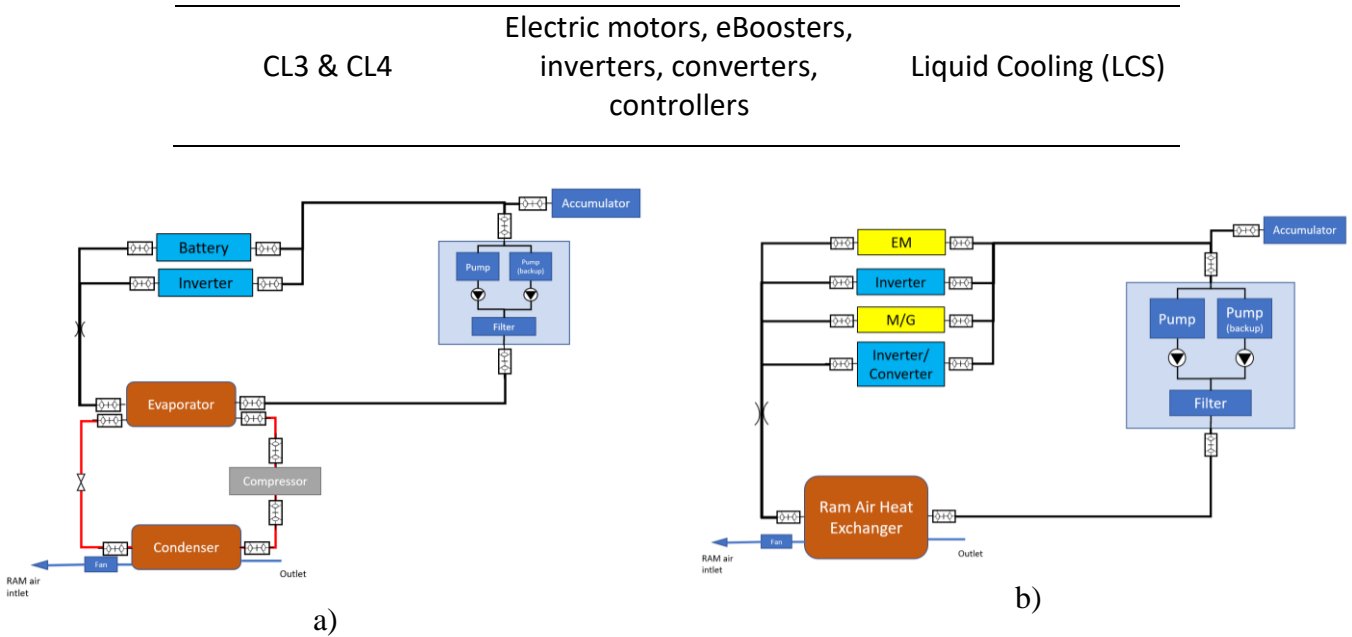


Figure 6: Cooling loops chosen for the aircraft

Once the thermal management system concept was established, detailed modelling was performed with Matlab/Simulink using textbook equations and reference papers. These models were later used to generate datapoints which, then, became part of the surrogate models for SUAVE.

2.3 Simulation parameters

In a hybrid electric aircraft, it is important to define the degree of hybridization (DoH). Several definitions may be found in literature [13] [14], and in this project, it was defined as the percentage of total installed power that comes from electrical engines:

$$DoH = \frac{P_{ele}}{P_{total}} \quad (1)$$

It is also necessary to define the energy management strategy (EMS), i.e. how the energy will be split throughout the flight. A few different concepts were developed and the one selected for this paper is illustrated in Figure 7. While the former is a simple example on the possible variations, the latter is the EMS adopted during this specific case study. Positive values mean that the device is generating or receiving energy, while negative values indicate that energy is being extracted by that component. The values represent percentages with respect to the equipment's nominal power. In other words, -1 for motor turboshaft (red) means that the motor/generator (M/G in Figure 2) connected to the turboshaft is extracting energy up to 100% of its own capacity – not that it is extracting 100% of the shaft energy. Likewise, 1 means that the component is operating at 100% its own capacity.

First A. Author, Second B. Author and Third C. Author

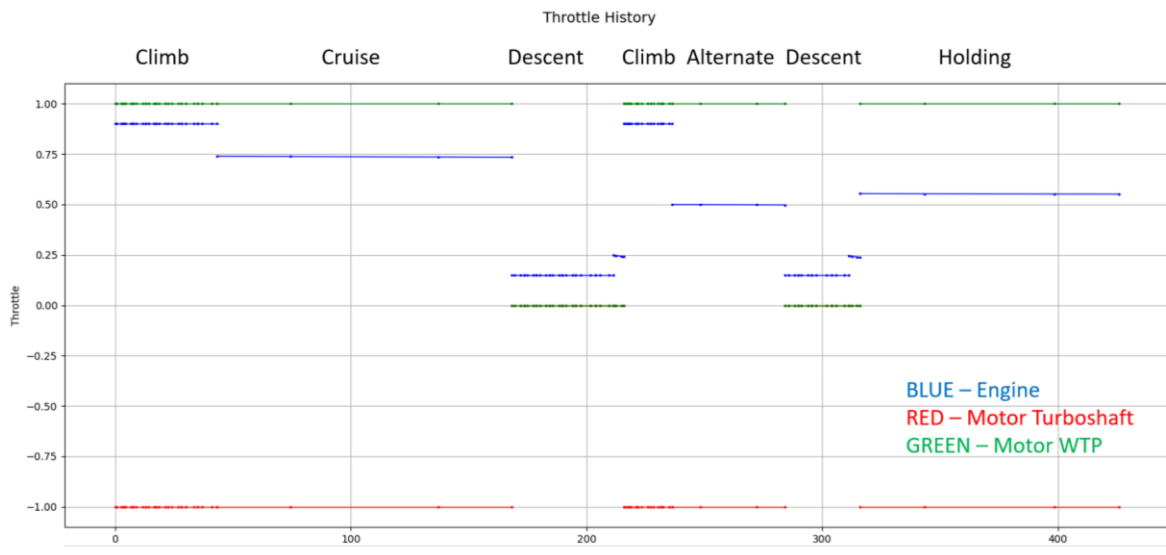


Figure 7: EMS chosen for this case study (EMS 3)

With the definitions established, mission and electric powertrain parameters were set as in Table 4. Several other inputs were defined but omitted here for concision purposes.

Table 4: Selected mission parameters

PARAMETER	VALUE
MISSION	
Distance	400 km
Cruise speed	450 KTAS ⁱ
Cruise altitude	17000 ft
ELECTRICAL POWERTRAIN	
Hybridization of power	20%
Bus voltage	800 V
Battery specific energy	500 Wh/kg

3. RESULTS AND DISCUSSION

3.1 Aircraft and mission

The mission profile generated in SUAVE is shown in figuresFigure 8 and Figure 9 was achieved for an aircraft with the characteristics displayed in Table 5 (SUAVE outputs). It should be noted that the output reproduced from the integration framework are numeric results of iteration processes, based on assumptions adequate for a conceptual study and, therefore, not all figures are significant.

ⁱ True airspeed in knots

Table 5: Selected SUAVE outputs

PARAMETER	VALUE
MASSES	
MTOM	18094 kg
Battery mass	192.37 kg
Electric motors, generators and ESCs	204.44 kg
High power electrical cables	100 kg
VCS (TMS CL1 and CL2) ⁱⁱ	70.26 kg
LCS (TMS CL3 and CL4) ⁱⁱⁱ	44.81 kg
WING GEOMETRY	
Wingspan	26.75 m
Wing wetted area	109.34 m ²
Aspect ratio	13.50

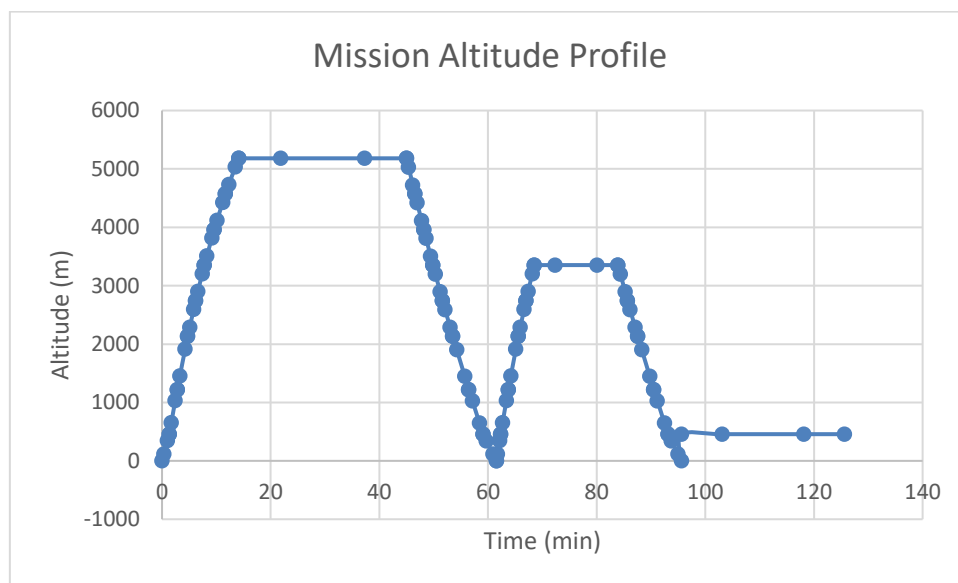


Figure 8: Mission altitude profile

From the results shown, the reader may quickly draw the conclusion that altitude requirement was met but speed was below the value established by the TLARs. Battery mass corresponds to approximately 1% of MTOM, and almost half of the whole electrical powertrain, while the total TMS contribution is about 0.6% of MTOM.

ⁱⁱ Mass value for both VCSs combined.

ⁱⁱⁱ Mass value for both LCSs combined.

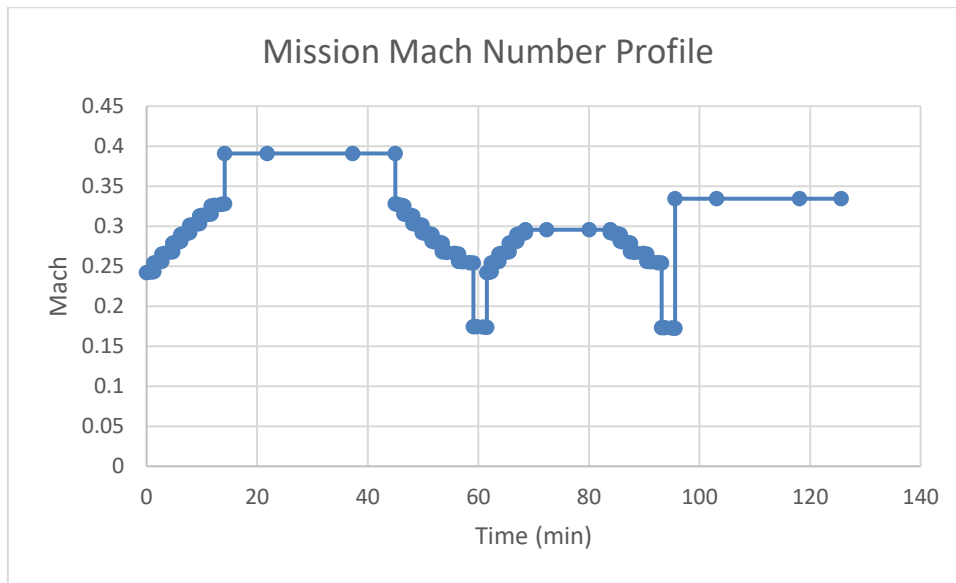


Figure 9: Mission Mach number profile

2.2 Thermal Management

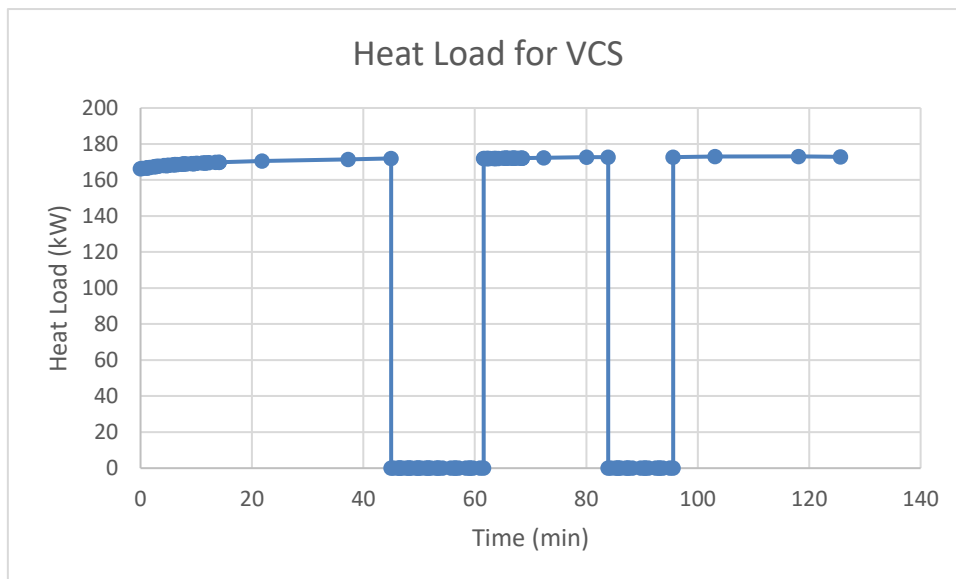


Figure 10: VCS heat load during the mission

With regards to the heat loads for the TMS – shown in figures Figure 10 and Figure 11 –, it follows the trend expected from energy management strategy and mission profile. During take-off and climb, the heat load is high because of the required power. Specifically on the VCS, a slight increase in heat load can be identified, reaching its maximum value at cruise. This is explained by the functioning of a gas turbine engines: as they are volumetric machines, with the increase in altitude and consequent decrease in density less air mass is admitted at the inlet for a given volume, which implicates in a decrease in power (other parameters remaining constant). From a powertrain point of view, this decrease in power is compensated by increase in power from the electrical engines and, therefore, more energy drawn from the batteries, that implies in higher heat

loads. On descent, the electrical motors are in idle and, as such, do not draw energy and do not generate any heat load. The go-around climb, flight to alternate destination and reserve follow the same logic as previously explained in this paragraph.

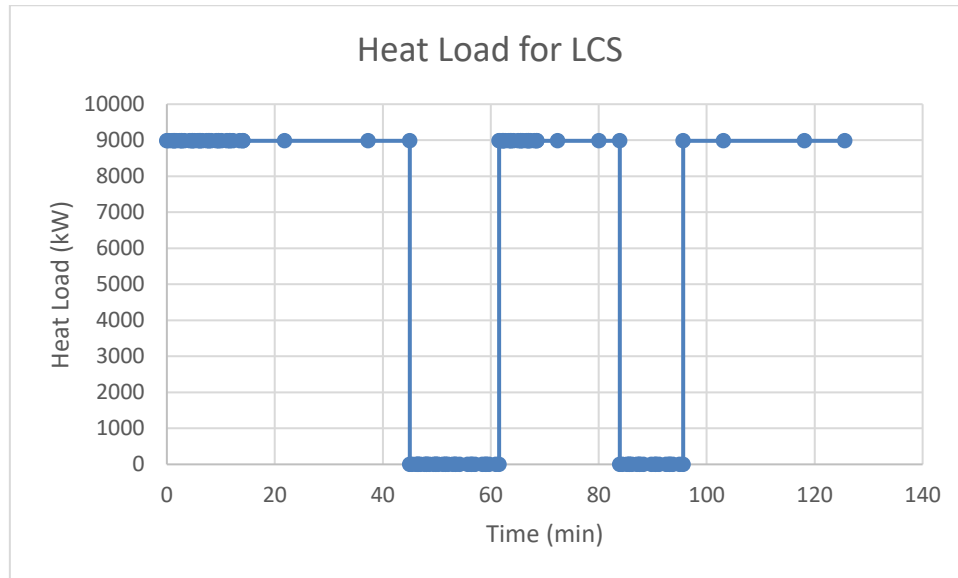


Figure 11: LCS heat load during the mission

In addition to heat load, which is translated into increase in aircraft empty weight and power consumption of the TMS, another important parameter is the required ram air. It should be reminded that the atmosphere is the final heat sink of the thermal cycles shown in introduction and, in these specific systems, the heat is dissipated into it at the ram air heat exchanger (LCS case) or condenser (VCS case). Figures Figure 12 and Figure 13 show the ram air consumption/requirement throughout the mission. The peaks correspond to low altitude (higher temperature) and low speed situations – conditions less favourable to heat exchange when compared to cruise, when lower values may be found. Finally, the reserve section presents high values because it is also close to ground. It should be noted that ram air consumption translates into drag. This is both due to pressure drops in the ducting and heat exchanger and difference in inlet and outlet speeds (change in *momentum*).

A big difference between both cycles is seen – about one order of magnitude. This is explained by the nature of each sub-system: while the VCS can bring the temperature of the working fluid lower than the atmospheric, hence, increasing the temperature difference at the condenser and improving the heat exchange, the LCS cannot do so. Therefore, more air flow is necessary to cool the working fluid (EGW).

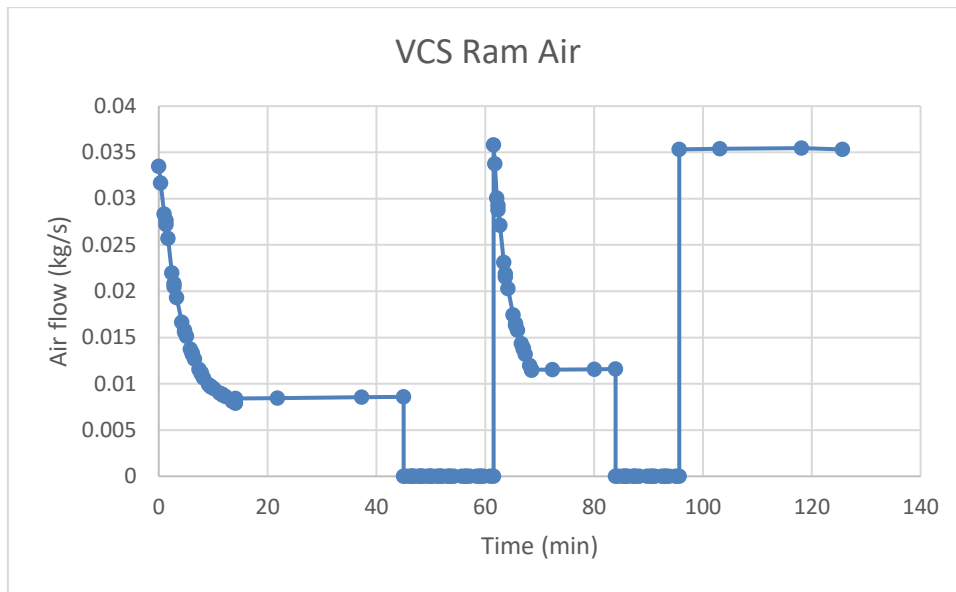


Figure 12: Required ram air for the VCS

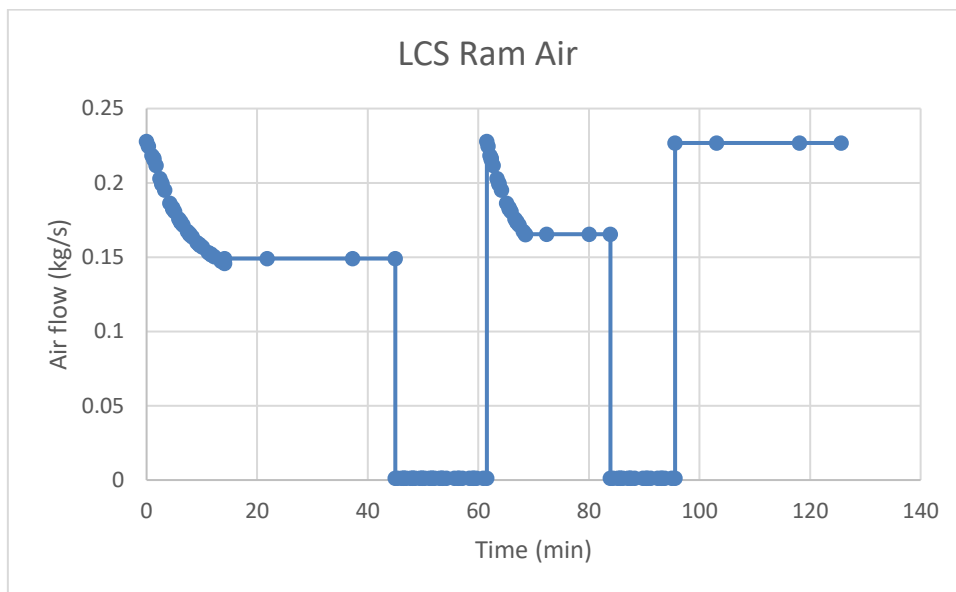


Figure 13: Required ram air for the LCS

2.3 Emissions

Total emissions are shown in Table 6. These parameters may be compared with other aeroplanes generated by SUAVE under different initial conditions. They are also important figures of merit that might indicate the viability of the design from environmental point of view from the very beginning of the design process.

First A. Author, Second B. Author and Third C. Author

Table 6: Emissions of the case study aeroplane

SUBSTANCE	VALUE (kg)
CO ₂	3121.068897
CO	2.606890116
NO _x	12.27984153

4. CONCLUSIONS

This study shows the capabilities developed by Futprint50 in SUAVE and selected results of the project [15] with particular emphasis on thermal management system. The aircraft generated partially matches the TLARs, with cruise speed below the one established but fulfilling range and reserves requirements. In addition, the thermal management system results show consistency with what is expected through the mission profile.

Power consumption could be improved with the use of passive devices, such as skin heat exchangers and phase change materials. They have the potential to diminish the ram air consumption as well, but careful trade-offs should be done to take into account their impact on weight.

Technology improvements could also lead to reductions in the parameters discussed. There are two main approaches to this: improving the thermal components, through increase in specific power, for example, and with the increase in electrical equipment operating temperatures. The latter could lead to bigger temperature differences in heat exchangers, which implies in smaller heat exchangers. It also opens the possibility for synergies, to use the heat absorbed in other equipment and systems.

The data generated and conclusions drawn in this study and other Futprint50 analyses can be used to elaborate roadmaps to inform academia, industry, policy makers and general public on the challenges and how to better direct the ecosystem efforts towards greener aviation.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No 875551.

The authors would like to thank Embraer, University of Stuttgart, Instituto Superior Tecnico and Cranfield University for their support and involvement.

REFERENCES

- [1] NASA, "Global Warming vs. Climate Change," NASA, [Online]. Available: <https://climate.nasa.gov/global-warming-vs-climate-change/>. [Accessed 29 June 2023].
- [2] Air Transport Action Group (ATAG), "Waypoint 2050, 2nd ed.," Air Transport Action Group, Geneva, 2021.
- [3] European Commission, "Flightpath 2050: Europe's Vision for Aviation," Publications Office of the European Union, Luxembourg, 2021.
- [4] Futprint50, "Project Overview," Futprint50, Online.
- [5] R. d. Reis, F. Odaguil, E. Windels, Y. Teeuwen, J. v. d. Pols, P. Laskaridis, D. Bergmann, D. Eisenhut and N. Moebs, "Deliverable 2.1: Requirements and Reference

- Aircraft," Futprint50, Online, 2021.
- [6] D. Eisenhut, N. Moebs, E. Windels, D. Bergmann, I. Geiß, R. Reis and A. Strohmayer, "Aircraft Requirements for Sustainable Regional Aviation," *Aerospace*, 26 February 2021.
- [7] W. Affonso Jr, R. Gandolfi, R. J. N. d. Reis, C. R. Ilario, N. Rodio, T. Kipouros, P. Laskaridis, A. Chekin, Y. Ravikovich, N. Ivanov, L. Ponyaev and D. Holobtsev, "Thermal Management challenges for HEA – FUTPRINT 50," in *10th EASN International Conference on Innovation in Aviation & Space to the Satisfaction of the European Citizens (10th EASN 2020)*, 2nd-5th September 2020, Online, 2020.
- [8] T. Lukaczyk, A. D. Wendorff, E. Botero, T. MacDonald, T. Momose, A. Variyar, J. Vegh, M. Colonno, T. D. Economon, J. J. Alonso, T. H. Orta and C. I. d. Silva, "SUAVE: An Open-Source Environment for Multi-Fidelity Conceptual Vehicle Design," American Institute of Aeronautics and Astronautics.
- [9] J. Mangold, D. Eisenhut, F. Brenner, N. Moebs and A. Strohmayer, "Preliminary hybrid-electric aircraft design with advancements on the open-source tool SUAVE," in *12th EASN International Conference on "Innovation in Aviation & Space for opening New Horizons"*, Barcelona, 2022.
- [10] A. van Heerden, D. M. Judt, S. Jafari, C. P. Lawson, T. Nikolaidis and D. Bosak, "Aircraft thermal management: Practices, technology, system architectures, future challenges, and opportunities," *Progress in Aerospace Sciences*, no. 128, 2022.
- [11] C. Roe, X. Feng, G. White, R. Li, H. Wang, X. Rui, C. Li, F. Zhang, V. Null, M. Parkes, Y. Patel, Y. Wang, H. Wang, M. Ouyang, G. Offer and B. Wu, "Immersion cooling for lithium-ion batteries – A review," *Journal of Power Sources*, vol. 525, 2022.
- [12] I. A. Hunt, Y. Zhao, Y. Patel and G. J. Offer, "Surface Cooling Causes Accelerated Degradation Compared to Tab Cooling for Lithium-Ion Pouch Cells," *Journal of The Electrochemical Society*, vol. 163, no. 9, 2016.
- [13] D. Obertino, "Design and Simulation Study of a Hybrid-Electric Propulsion System for a VTOL Tilt-Rotor UAV, MSc Thesis," Instituto Superior Tecnico, Universidade de Lisboa, Lisboa, 2021.
- [14] R. Larkens, "A coupled propulsion and thermal management system for hybrid electric aircraft design, MSc Thesis," TU Delft, Delft, 2020.
- [15] Futprint50, "Results," Futprint50, [Online]. Available: <https://futprint50.eu/results>. [Accessed 29 June 2023].
- [16] Air Transport Action Group (ATAG), "Waypoint 2050, 2nd ed.," Air Transport Action Group, Geneva, 2021.



MULTI-OBJECTIVE BAYESIAN OPTIMIZATION WITH MIXED-CATEGORICAL DESIGN VARIABLES FOR EXPENSIVE-TO-EVALUATE AERONAUTICAL APPLICATIONS

N. Bartoli^{1*}, T. Lefebvre¹, R. Lafage¹, P. Saves^{1,2}, Y. Diouane³, J. Morlier²,
J. H. Bussemaker⁴, G. Donelli⁴, J. M. Gomes de Mello⁵, M. Mandorino⁶, P.
Della Vecchia⁶

1: ONERA/DTIS,
Université de Toulouse,
31400 Toulouse, France
{nathalie.bartoli,thierry.lefebvre,remi.lafage,paul.saves}@onera.fr

2: ICA, ISAE-SUPAERO, INSA, CNRS, MINES ALBI, UPS
Université de Toulouse,
31400 Toulouse, France
joseph.morlier@isae-supero.fr

3: Polytechnique Montréal
Montreal, QC, Canada
youssef.diouane@polymtl.ca

4: German Aerospace Center (DLR)
Institute of System Architectures in Aeronautics
Hein-Saß-Weg 22, 21129 Hamburg, Germany
{jasper.bussemaker,giuseppa.donelli}@dlr.de

5: EMBRAER S.A.
Av. Brigadeiro Faria Lima, 2170, São José dos Campos, Brazil
joao.mello@embraer.com.br

6: UNINA
University di Napoli Federico II
Naples, 80125, Italy
{massimo.mandorino,pierluigi.dellavecchia}@unina.it

Abstract. *This work aims at developing new methodologies to optimize computational costly complex systems (e.g., aeronautical engineering systems). The proposed surrogate-based method (often called Bayesian optimization) uses adaptive sampling to promote a trade-off between exploration and exploitation. Our in-house implementation, called SEGOMOE, handles a high number of design variables (continuous, discrete or categorical) and nonlinearities by combining mixtures of experts for the objective and/or the constraints. Additionally, the method handles multi-objective optimization settings, as it allows the construction of accurate Pareto fronts with a minimal number of function evaluations. Different infill criteria have been implemented to handle multiple objectives with or without constraints. The effectiveness of the proposed method was tested on practical aeronautical applications within the context of the European Project AGILE 4.0 and*

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

demonstrated favorable results. A first example concerns a retrofitting problem where a comparison between two optimizers have been made. A second example introduces hierarchical variables to deal with architecture system in order to design an aircraft family. The third example increases drastically the number of categorical variables as it combines aircraft design, supply chain and manufacturing process. In this article, we show, on three different realistic problems, various aspects of our optimization codes thanks to the diversity of the treated aircraft problems.

Keywords: Bayesian optimization, multi-objective problems, mixed-categorical variables, aircraft design, aeronautical applications

1 INTRODUCTION

For the last few decades, the aeronautical industry has been essentially driven by incremental improvements based on aircraft design optimization [1]. The use of computer science allowed to reuse well-proven design processes and to benefit from preexisting results to reduce the time dedicated to development. Nonetheless, the forthcoming challenges awaiting aircraft design call for large-scale improvements to meet future expectations in terms of overall environmental impact, noise reduction and cost-effectiveness [2]. These large-scale improvements, considered through novel aircraft configurations, involve an increased proximity between Aerodynamics, Propulsion and Structural mechanics, thus requiring to explore coupled and uncharted physics. As is the case for many industrial complex systems, some powerful optimization algorithms for aircraft design are needed. The H2020 European project AGILE 4.0 (2019-2023)¹ [3], led by DLR, has identified several challenging application cases linked to Multidisciplinary Design Optimization (MDO) methods [4] and Model Based Systems Engineering (MBSE) technologies [5] that will require innovative and sustainable aircraft configurations in the years to come [6]. The main objective is to reduce the environmental impact in terms of fuel consumption, waste and emissions associated with all the aeronautical system activities and operations. This purpose requires extending aeronautical research to cover the entire aircraft lifecycle, from design and production to waste disposal after the system's end-of-life. Consequently, the challenge is to incorporate these new requirements into the early design phase, enabling strategic decision-making processes to optimize the entire aircraft lifecycle [7]. To do this, collaborative multidisciplinary aircraft design and optimization should be carried out involving not-only the aircraft design domain (typically accounted during the conceptual aircraft design) but also other industrial domains, such as manufacturing, supply chain, maintenance or certification [8]. From the different application cases of the project, some quantities of interest have been identified from the Multidisciplinary Design Analysis (MDA) and optimization problems have been deduced involving from 2 to 5 objective functions and several nonlinear constraints that are given by non derivative and expensive-to-evaluate black box functions. Moreover, when evaluating new architecture systems, it is important to take into account categorical variables such as on-board system architecture (conventional, hybrid, electric for instance) or property materials (aluminum, titanium), which do not have a defined order [9]. Different approaches have been investigated to deal

¹<https://www.agile4.eu/>

with mixed integer variables, multiple objectives and constraints handling requirements. These optimizations return a set of feasible configurations considered as optimal trade-offs. This set of solutions is the so-called *Pareto front* that consists in the set of solutions that cannot be improved in one objective without worsening another [10].

Surrogate-Based Optimization (SBO) is a powerful technique for optimizing complex systems requiring a significant amount of computational resources. For multi-objective optimization problems, it involves building a surrogate model, such as a machine learning model, to approximate the behavior of multiple objectives simultaneously, allowing for efficient exploration of the search space and identification of the Pareto front. The surrogate models built offline are then used to predict the objective function values at different points in the search space, enabling the optimization algorithm to focus its computational resources on the most promising regions of the search space [11].

To avoid this offline cost involving a large number of function evaluations, some adaptive strategies have been introduced. Nowadays, Bayesian Optimization (BO) [12] is well known as a powerful tool in scientific research for optimizing complex, expensive-to-evaluate functions with unknown properties. It is particularly useful when only a limited number of evaluations is available which is often the case with scientific experiments. When multiple objectives are involved, multi-objective techniques must be used to find the set of optimal solutions that maximize or minimize these multiple objectives simultaneously, rather than focusing on a single objective. Multi-objective optimization using Bayesian algorithms is a potent strategy for optimizing intricate systems that possess multiple, conflicting objectives. Bayesian optimization techniques allow for efficient exploration of the search space by harnessing probabilistic models to forecast the behavior of the objectives being optimized. This facilitates a trade-off between exploration and exploitation and allows the inclusion of prior knowledge on the objectives.

In summary, multi-objective optimization with Bayesian algorithms is a valuable tool for scientific research, providing an efficient way to identify the Pareto front and make informed decisions based on multiple, conflicting objectives. Here some illustrations are done on three application cases of the AGILE 4.0 project with different characteristics. On the first application case, about retrofitting concepts (engine, winglet, on-board systems), the different impacts on aircraft performance are characterized (e.g., Direct Operating Cost (DOC), emissions) to select the best retrofitting strategy. The second application is about designing an aircraft family composed of three different aircrafts with different commonality choices (wing, engine, empennage). Again, the different solutions are characterized by different impacts on aircraft performance (e.g. mass, aerodynamic efficiency) and different commonality choices. To finish with, the third application concerns Horizontal Tail Plane (HTP) production and search for an equilibrium between airplane performance, manufacturing costs and the efficiency of the supply chain.

The outline of the paper is as follows. In Section 2, the multi-objective optimization problem is presented and two proposed frameworks are then described. A first SBO algorithm JPAD Optimizer is introduced in Section 2.2 and a second BO framework SEGOMOE is given in Section 2.3 where the Continuous Relaxation (CR) as well as the use of the Partial Least Squares (PLS) technique are detailed. Some details concerning the remote access to the frameworks are also provided. Then Section 3 presents three MDO application cases where multiple physics and domains are involved. For each of the application cases, multidisciplinary analysis, optimization problem as well as the obtained

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

results are detailed. Conclusions and perspectives are finally drawn in Section 4.

2 MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS

To begin with, we start formulating the multi-objective paradigm over which the paper will be based. Then, we introduce our two frameworks used to solve such multi-objective optimization problems in an industrial context.

2.1 Multi-objective formalism

Consider the multi-objective constrained optimization problem

$$\left\{ \begin{array}{l} \min_{\mathbf{x} \in \Omega \times S \times \mathbb{F}^l} \mathbf{f}(\mathbf{x}) := [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x})] \\ \text{s.t. } g_1(\mathbf{x}) \leq 0 \\ \quad \vdots \\ g_m(\mathbf{x}) \leq 0, \end{array} \right. \quad (1)$$

where $\Omega \subset \mathbb{R}^d$ represents the bounded continuous design set for the d continuous variables, $S \subset \mathbb{Z}^\ell$ represents the bounded integer set where L_1, \dots, L_ℓ are the numbers of levels of the ℓ quantitative integer variables on which we can define an order relation, and $\mathbb{F}^l := \{1, \dots, L_1\} \times \{1, \dots, L_2\} \times \dots \times \{1, \dots, L_l\}$ is the design space for l categorical qualitative variables with their respective L_1, \dots, L_l levels [9]. The objectives functions are referred as $f_i : \mathbb{R}^d \rightarrow \mathbb{R} \quad \forall i = 1, \dots, n$ where n is the number objectives. The constraints are named $g_j : \mathbb{R}^d \rightarrow \mathbb{R} \quad \forall j = 1, \dots, m$ where m is the number of constraints (inequality or equality). Due to potentially conflicting objectives, the solution of the optimization is not unique but a range of solutions is proposed. The trade-off between these solutions is characterized by the notion of dominance: a solution \mathbf{x} is said to dominate another solution \mathbf{x}' and denoted by $\mathbf{x} \preceq \mathbf{x}'$ if

$$f_i(\mathbf{x}) \leq f_i(\mathbf{x}') \quad \forall i = 1, \dots, n.$$

The set of solutions representing the optimal trade-off is referred to the Pareto set (PS) and the corresponding image of PS in the objective space is known as the Pareto front (PF)

$$\text{PF} := \{\mathbf{f}(\mathbf{x}) | \mathbf{x} \in \text{PS}\}.$$

Figure 1 illustrates a Pareto front (red points) relative to two objectives. An approximation to this PF is given by the scattered green dots and the associated dominated hypervolume given by the green area. Both the SBO and BO frameworks to solve problem (1) are described in the following.

2.2 Surrogate-Based Optimization with JPAD

In this work, one of the optimization tools used is the JPAD Optimizer, which is based on the Multi-Objective Evolutionary Algorithms (MOEA) framework. It should be noted that the JPAD Optimizer is an independent implementation within the JPAD library, as described in [13, 14]. The MOEA framework is a Java library that serves as an open-source platform for the development and experimentation of various optimization techniques. The latter offers a range of pre-implemented algorithms that can be readily

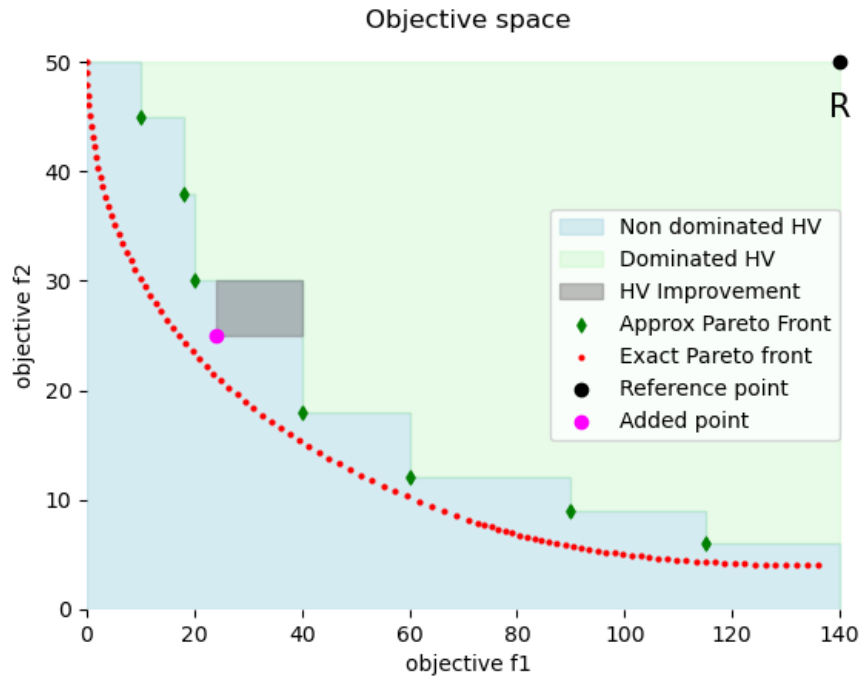


Figure 1: Hypervolume Improvement: the hypervolume indicator of the non-dominated set (green points) corresponds to the area dominated by it, up to R (reference point in black). The gray rectangle is the hypervolume improvement brought by the new added point in magenta.

employed, such as genetic algorithms and particle swarm optimization, among others. For this particular study, the ϵ -NSGAI algorithm has been used. The ϵ -NSGAI algorithm extends the NSGA-II algorithm by incorporating an ϵ -dominance archive and randomized restarts, thereby enhancing the search process and enabling the discovery of a diverse set of Pareto optimal solutions. A comprehensive explanation of this algorithm can be found in [15] with in-depth details and comparisons. The JPAD Optimizer requires a comprehensive factorial database to conduct the analysis effectively. This database enables the solver to construct a specific Response Surface Model (RSM) through spline interpolation. The RSM serves as a surrogate model, capturing the relationship between the input variables and the response variable. By utilizing the RSM as a surrogate model, the optimization algorithm can efficiently explore the design space and identify optimal solutions. However, it is important to note that direct inclusion of categorical variables in the optimization is not feasible. To address this limitation, one potential solution is to perform the optimization iteratively, considering each categorical variable separately for each run. Still, using the JPAD Optimizer in conjunction with the MOEA framework provides a robust and versatile approach to the optimization process. The selection of the ϵ -NSGAI algorithm within the MOEA framework allowed for efficient and effective exploration of the design space, leading to the identification of a diverse range of Pareto optimal solutions. The chosen references provide additional information on the JPAD library, the MOEA framework, and the ϵ -NSGAI algorithm, serving as valuable resources for readers interested in a more detailed understanding of the employed methodologies.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

2.3 Bayesian Optimization with SEGOMOE

The second proposed optimization process is based on a sequential enrichment approach, typically the Efficient Global Optimization algorithm [11] or Super EGO [16], an evolution of EGO to handle constraints. Bayesian Optimization (BO) is based on Gaussian processes (GP) [17] (also denoted by Kriging [18]) and here the idea is to use some adaptive mixture of Kriging based models to tackle high dimension problems. The mixture of experts (MOE) [19, 20] is known to approximate complex functions with heterogeneous behaviour by combining local surrogate models in a global one. In order to consider high-dimensional functions to approximate objective functions and constraints, we used adapted local Kriging-based models [21, 22]. Some recent developments have been made to consider highly non linear constraints [23], mixed integer variables [24] and multi-objective applications [10]. The general framework called Super Efficient Global Optimization coupled with Mixture Of Experts (SEGOMOE) has been proposed by ONERA and ISAE-SUPAERO. The performance of SEGOMOE is validated and proven on different analytical and industrial test cases [25–27].

Here we consider multiple objectives and constraints with mixed integer variables. Some continuous, integer and categorical variables are involved within the different application cases, so the GP surrogate models have to be adapted to deal with. We proposed to combine continuous relaxation to replace categorical variables by continuous ones and Partial Least Squares techniques to handle the increase of dimensionality. The continuous relaxation introduced by [28] uses a one-hot encoding strategy [29] and transforms integer and categorical inputs into continuous ones, so some classic continuous correlation kernels can still be used to build the Kriging models. The design space $\Omega \times S \times \mathbb{F}^l$ is relaxed to a continuous space Ω' constructed on the following way [24]:

- $\forall i \in \{1, \dots, \ell\}$, the integer variable z_i is relaxed within its bounds and treated as continuous.
- $\forall j \in \{1, \dots, l\}$, we use a relaxed one-hot encoding [29] for the categorical variable c_j (and its L_j associated levels) and add L_j new continuous dimensions into Ω' .

Therefore, we get, after relaxation, a new design space $\Omega' \subseteq \mathbb{R}^{d'}$ where $d' = d + \ell + \sum_{j=1}^l L_j > d + \ell + l$. As the number of variables may increase drastically (from d to d'), one new added variable associated to each categorical level, PLS technique is combined with Kriging, denoted as KPLS and KPLS-K models [21, 22], to reduce the number of hyperparameters. During experiments, the chosen number of principal components within KPLS models does not exceed 4 or 5. The code implementation of the proposed GP has been released in the toolbox SMT v1.1² [30].

Once the initial Design of Experiments is given in the relaxed space Ω' , Kriging based surrogate models are built for the objective and constraint functions. Each costly function $f_i(\mathbf{x})$ is approximated by a GP characterized by its mean $\hat{y}_i(\mathbf{x}) : \mathbb{R}^{d'} \rightarrow \mathbb{R}$ and its standard deviation $s_i(\mathbf{x}) : \mathbb{R}^{d'} \rightarrow \mathbb{R}$

$$\hat{f}_i(\mathbf{x}) \sim \mathcal{N}(\hat{y}_i(\mathbf{x}), s_i^2(\mathbf{x})) \quad i = 1, \dots, n$$

²<https://smt.readthedocs.io/en/latest/>

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

For the multi-objective, we assume that the n components of \mathbf{f} are independant to define $\hat{\mathbf{f}}$ as the surrogate model associated to each component

$$\hat{\mathbf{f}}(x) \sim \mathcal{N}(\hat{\mathbf{y}}(\mathbf{x}), \Sigma(\mathbf{x}))$$

where $\hat{\mathbf{y}}(\mathbf{x}) : \mathbb{R}^{d'} \rightarrow \mathbb{R}^n$ is the GP prediction vector given by $[\hat{y}_1(\mathbf{x}), \dots, \hat{y}_n(\mathbf{x})]$ and $\Sigma(\mathbf{x})$ is a diagonal matrix whose diagonal vector is given by $s_i^2(\mathbf{x}), \forall i = 1, \dots, n$. So the initial problem defined by Eq.(1) is replaced by an infill problem defined as follows

$$\begin{cases} \max_{x \in \mathbb{R}^{d'}} & \alpha_{\mathbf{f}}^{\text{reg}}(\mathbf{x}) \\ \text{s.t.} & \hat{g}_1(\mathbf{x}) \leq 0 \\ & \vdots \\ & \hat{g}_m(\mathbf{x}) \leq 0, \end{cases} \quad (2)$$

where $\hat{g}_i(\mathbf{x})$ corresponds to the mean prediction of the GP constraint models and the regularized acquisition function [10] is defined by

$$\alpha_{\mathbf{f}}^{\text{reg}}(\mathbf{x}) := \gamma \alpha_{\mathbf{f}}(\mathbf{x}) - \psi(\mu_{\mathbf{f}}(\mathbf{x})) \quad (3)$$

where $\alpha_{\mathbf{f}}(\mathbf{x})$ is a standard scalar acquisition (EHVI, PI, MPI, ...) depending on $\hat{\mathbf{y}}(\mathbf{x})$ and $\Sigma(\mathbf{x})$, and γ is a constant parameter. The function $\psi : \mathbb{R}^{d'} \rightarrow \mathbb{R}$ is a scalarization operator. Different choices exist for the function ψ and in [10] two options were investigated. Namely, for a given $\hat{\mathbf{y}}(\mathbf{x}) \in \mathbb{R}^{d'}$, we consider

$$\begin{aligned} (\text{reg} = \max) & : \psi(\hat{\mathbf{y}}(\mathbf{x})) := \max_{i \leq d'} \hat{y}_i(\mathbf{x}) \\ (\text{reg} = \text{sum}) & : \psi(\hat{\mathbf{y}}(\mathbf{x})) := \sum_{i=1}^{d'} \hat{y}_i(\mathbf{x}) \end{aligned}$$

The performance of this regularized acquisition function (3) has been tested on analytical examples involving continuous variables and apply with success to an aircraft design test case in [10].

Concerning the different BO criteria relative to the hypervolume Improvement, an illustration is proposed in Fig.1. The Expected Hyper-Volume Improvement (EHVI [31, 32]), the Probability of Improvement (PI [33]), and the Minimum of Probability of Improvement (MPI [34]) are some multi-objective extensions of the well known Expected Improvement (EI [11]). As seen in Fig.1, with respect to the predictions in the Gaussian random field, the idea is to measure how much hypervolume improvement (grey area) could be achieved by evaluating the new point (magenta point), considering the uncertainty of the prediction. These criteria differ depending how they favor well-spread solutions: for instance for EHVI the hypervolume increase is small when adding a new value close to an already observed one in the objective space. To solve problem Eq.(2), any optimization algorithms capable of considering non linear constraints based either on derivative free optimizer such as COBYLA (Constrained Optimization BY Linear Approximation [35]) or based on gradient method such as SLSQP (Sequential Least Squares Programming [36]) or SNOPT (Sparse Nonlinear OPTimizer [37]) can be used together with a multistart strategy. This adaptive process is repeated until the total budget is reached. The feasible

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

points of the final database represent the known Pareto optimal points. Nevertheless, as the set of known points has been enriched sequentially to increase the hypervolume, the final database can be used to build GP models for objectives and constraints as a post-processing step. These final GP models can be coupled to an evolutionary algorithm to deal with the multi-objective constrained problem and retrieve the approximated PF. The well known NSGA-II algorithm (Non-dominated Sorting Genetic Algorithm II [38]) of the pymoo [39] toolbox³ is used for that purpose and ultimately provides the predicted PF based on GP with almost no additional computational cost.

In the end, the proposed strategy provides two outputs: the PF database and the predicted PF. Comparing the proximity between the two PF is a good criterion to know if some additional enrichment points are needed or if the accuracy is sufficient. Algorithm 1 details the SEGOMOE optimization procedure.

Algorithm 1: SEGOMOE for constrained multi-objective and mixed-integer problems.

Inputs: Initial DoE \mathcal{D}_0 and set $t = 0$;

while the stopping criterion is not satisfied **do**

1. Relax continuously integer and categorical input variables to a real bounded space Ω' of dimension $d' = d + \ell + \sum_{j=1}^l L_j$. Namely, we continuously relax the mixed categorical DoE to a continuous DoE \mathcal{D}_t using the relaxation procedure;
2. Build the GP model for each objective function $f_i(\mathbf{x})$ and each constraint $g_j(\mathbf{x})$ related to the continuous DoE with PLS to reduce the number of the hyperparameters and compute an estimation of the search space Ω_f ;
3. Build the acquisition function $\alpha_f^{\text{reg}}(\mathbf{x})$;
4. Maximize the acquisition function within the feasible domain Ω_f :

$$\mathbf{x}_t := \arg \max_{\mathbf{x} \in \Omega_f} \alpha_f^{\text{reg}}(\mathbf{x})$$

5. Add \mathbf{x}_t , $\mathbf{f}(\mathbf{x}_t)$ and $g_1(\mathbf{x}_t), \dots, g_m(\mathbf{x}_t)$ to the DoE \mathcal{D}_{t+1} . Increment t ;

end

Post-process: Use the final database to build GP models (for $f_i(\mathbf{x})$ and $g_j(\mathbf{x})$) and apply NSGA-II algorithm to construct the PF ;

Outputs: The PF database and the predicted PF;

2.4 Optimizers access

In AGILE 4.0 project, a Process Integration and Design Optimization (PIDO) component, also known as RCE [40], is required to execute disciplinary workflows. This executable workflow integrates the different disciplinary tools, which are treated as black-boxes: only inputs and outputs are relevant for the component and the codes are not exposed. The disciplinary tools are installed locally or in servers. Different types of MDO problems can be executed in the component, as basic multidisciplinary analysis, DOEs

³<https://pymoo.org/>

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

or full optimizations. Another key aspect of the collaborative workflow derives from the collaborative design process. Since the different disciplinary tools can be hosted in different locations, executable workflows are typically distributed simulation workflows across multiple networks in multiple companies. In AGILE 4.0, Brics tool [41, 42] is used to provide the mechanism for interconnecting PIDO environments and hence multi-partner and distributed collaborative simulation workflows. By providing data encryption services, Brics supports protection of the data to be exchanged to and from the central data server in which the inputs/outputs (I/O) file is transferred between the disciplinary tools. In the project, I/O file relies on Common Parametric Aircraft Configuration Schema (CPACS) [43]. CPACS is an open-source, XML-based common language for the exchange of product data. It allows storage of parametric definitions of aircraft geometries as well as analysis results of the individual design disciplines. Therefore, any legacy engineering tool in AGILE 4.0 context is equipped with a wrapper that enables the tool to act on product data and produce analysis results in CPACS format.

As all the application cases have to address multi-objective optimization problems with mixed variables, the optimization algorithms currently available through RCE were not entirely suitable (direct application of genetic algorithms would result in an excessively large number of evaluations). Only a few partners, such as ONERA, NLR, or UNINA, were able to provide gradient-free algorithms capable of solving these types of optimization problems. Therefore three different technical approaches have been applied to connect optimizers to the workflows:

- Local optimizer: the first approach consisted in using an adequate optimization algorithm located on the integrator’s site. This means that the integrator of the workflow needs to have also an expertise in optimization. In AGILE 4.0, only UNINA could apply this approach with JPAD algorithm as illustrated in Section 3.1.
- Remote optimizer through Brics call: the second approach consisted in considering an MDA workflow as a whole that can be called remotely like any design competence from an RCE specific workflow located in the optimizer owner’s premises. This approach was successfully applied in multiple application cases and was retained for the application case presented in Section 3.3.
- Remote access through WhatsOpt: the third approach applied the optimization process not on the design workflow but directly on the architectural choices that is connected to the RCE workflow. In that case, the optimizer is made accessible “as a service” through WhatsOpt [44], a web application supporting MDO collaborative activities. This approach was successfully demonstrated in the application case illustrated in Section 3.2.

3 APPLICATIONS

3.1 Airframe upgrade design

This first application case is part of the Upgrade driven stream and aims at accounting retrofitting options in the aircraft MDO workflow. The AGILE 4.0 MDO framework is used to design various retrofitting concepts, including engine replacement, OBS electrification, and winglet installation. Each solution has different impacts on aircraft performance, such as fuel consumption, direct operating cost (DOC), air and noise emissions, as well

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

as different retrofitting costs. The objective is to be able to select the optimal retrofitting strategy by trading DOC and emissions against retrofitting costs.

For this specific application case, a classical Multidisciplinary Feasible (MDF) approach is being utilized. This type of architecture involves an optimizer driving an MDA workflow, where a complete design is executed in each iteration. The MDA components comprising the workflow can be observed in Figure 2. The workflow begins with an engine module, which provides key engine characteristics based on the given engine ByPass Ratio (BPR) input. This module relies on GASTurb⁴ computation. The aerodynamics branch calculates values for both low-speed and high-speed conditions. To account for high-fidelity results in high-speed conditions obtained through CFD analysis, a RSM has been developed. The tool also considers engine geometry and position as inputs. The OBS discipline, named ASTRID [45], is responsible for sizing the overall on-board system (OBS). It determines the masses and the hydraulic, pneumatic, and electric power required by each system during different phases of the mission profile. It also calculates the power-off-takes (secondary power computation) to consider their impact on engine fuel flow. In this study, four different OBS architecture electrification levels are considered, ranging from CONV (conventional) to AEA (all electric aircraft) with two intermediate architectures, namely MEA1 and MAE2 (more electric aircraft 1 and 2). A more detailed description of these architectures can be found in [46]. The performance tool uses a simulation-based approach to compute ground and in-flight performance. It calculates overall mission profile, performance, fuel consumption, flight time, and gaseous emissions. The structural competence relies on a surrogate model based on a high-fidelity structural analysis DOE. This tool optimizes the wing mass by adjusting lamination parameters and thicknesses while ensuring aeroelastic stability, angle-of-attack, strength, and buckling constraints are met. The last three tools described are executed in a converger loop, with iterations performed until convergence is achieved on the maximum take-off weight. The noise competence calculates noise emissions at the certification points in accordance with FAR36 and ICAO Annex 16 regulations. It also provides the noise margin with respect to the certification limits. The cost competence evaluates both recurring and non-recurring costs, including aircraft price and direct operating costs. It also estimates the development, operation, and equipment costs associated with a retrofitting activity. Additionally, it computes cost savings (a component of direct operating costs) resulting from reductions in fuel consumption, maintenance costs, and emission taxes. The cost methodology is explained in [47]. Further details about the MDA workflow presented here can be found in [48].

This application case includes the integration of costly evaluations required to compute the wing structural weight and the high-fidelity aerodynamic effect of the relative nacelle/wing location. To mitigate the cost and computational time, for each design competence, a RSM was built in an off-line process with SMT [49]. A database computed by PROTEUS [50], a tool developed by TUD, is used to calculate the wing's structural mass. The RSM utilizes a database of 56 points (to achieve more than 10 points per dimension) obtained through a Latin Hypercube Sampling (LHS) approach. Additionally, an RSM has been developed based on CFD analysis provided by CFSE to consider the aerodynamic effects of both the pylon and the installed engine on the drag coefficient. The RSM for this purpose utilizes a database of 90 points obtained through an LHS approach. For this RSM, the initial database was smaller (around 10 points per dimension)

⁴<https://www.gasturb.de/> GasTurb

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

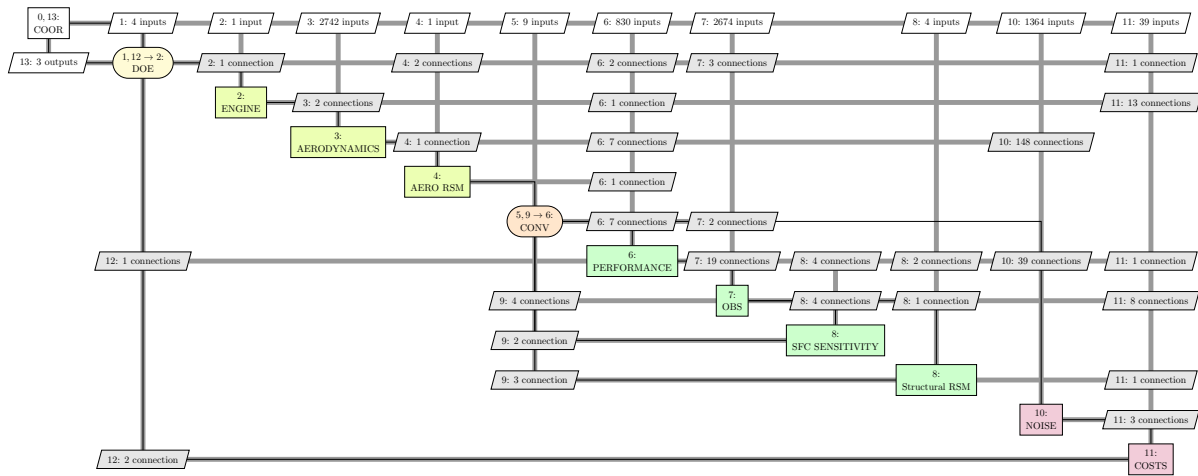


Figure 2: XDSM DOE MDA for Airframe upgrade design.

but had to be enriched to increase the accuracy of drag prediction. These RSMs have been incorporated as design competences within this workflow.

Regarding the choice of optimization solver, two different approaches were tested during the frame of the project. The first optimization strategy relies on the JPAD optimizer provided by UNINA and described in Section 2.2. In this approach, a DOE (full factorial composed of 108 points) is built using the workflow illustrated in Figure 2. In a second step, using JPAD Optimizer, four different optimizations are performed, each corresponding to one of the categorical variables (OBS choice) considered in the optimization problem. For each of this optimization and each quantity of interest, a specific RSM using spline interpolation is built based on the obtained database using the JPAD optimizer. Subsequently, the optimization algorithm is applied, using the RSM. The generation of the DOE dataset occurs online, while the offline optimization process takes place once all the data are available for processing. The second optimization approach takes advantage of SEGOMOE, from ONERA (see Section 2.3) that can handle directly categorical variables. With this multi-objective BO solver, a remote call to the MDA workflow from the ONERA location was necessary at each evaluation as explained in Section 2.4. The optimization workflow executed using SEGOMOE is illustrated in Figure 3 starting with an initial DOE of 13 points (selected among the 108 full factorial points) and running 68 iterations.

The optimization problem addressed by both of the methodologies presented is defined in Table 1. Four main objective functions have been established. The first objective is the maximum take-off weight (MTOW) of the aircraft, which is computed through the iterative convergence process described earlier. The second objective is the cumulative emission index (CEI), which quantifies the level of emissions in terms of air and noise. This index is calculated as a weighted sum of three emissions, expressed in non-dimensional units. The third objective is the difference between costs and savings. This objective captures the disparity between two economic factors. The first factor encompasses the capital costs incurred by the manufacturer to retrofit the aircraft, including the expenses associated with development, operations, and equipment required for the retrofitting activity. The second factor concerns the operational savings, which include the reductions in fuel consumption, maintenance costs, and emission taxes achieved compared to the

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

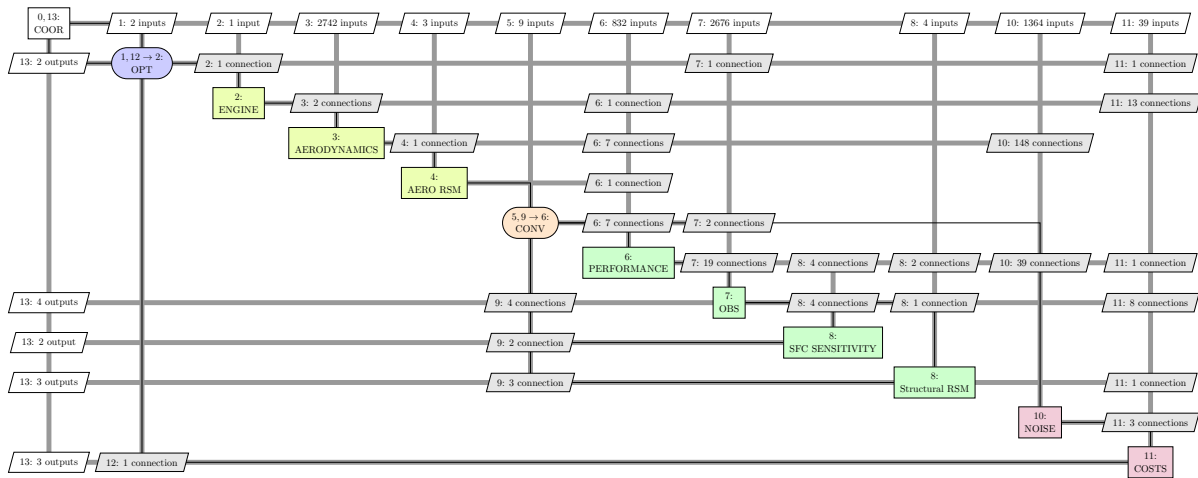


Figure 3: XDSM optimization for airframe upgrade design.

baseline aircraft during a typical mission. The fourth objective is the maximum specific air range (SAR) achievable in cruise conditions and weight, representing the aircraft’s flight efficiency and fuel consumption. Four design variables are considered. One variable is categorical, representing four different OBS architectures that range from conventional to all-electric configurations. The other three variables are continuous, including the engine BPR and the engine X and Z locations with respect to the wing attachment points. Figure 4 represents the system under analysis, illustrating the components that can be retrofitted and are affected by the design variables considered in the problem. Lastly, four constraints have been selected. The retrofit solution must maintain at least the same maximum take-off weight and the same take-off and landing distances as the baseline configuration. Additionally, a reduction of 6 dB in cumulative noise emissions must be achieved compared to the original aircraft, leading to a maximum cumulative noise equal to 263 dB.



(a) Engines and OBS are highlighted. (b) CFD simulation of nacelle location effects.

Figure 4: Reference regional jet aircraft.

The following are the results of the optimization process explained earlier. Figure 5 displays the DOE points and the PF obtained using both approaches. The workflow depicted in Figure 2 generated a total of 108 points in the DOE, represented by orange dots in Figure 5. On the same figure are highlighted, as red crosses, the points satisfying the optimization constraints for both DOE (45 feasible points out of 108 full factorial

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

Table 1: Definition of the airframe upgrade design problem.

	Function/variable	Nature	Quantity	Range
Minimize	Maximum Take-Off Weight	cont	1	
Minimize	Cumulative Emission Index	cont	1	
Minimize	Cost - savings	cont	1	
Maximize	Max Cruise Specific Air Range	cont	1	
	Total objectives		4	
with respect to	OBS architecture	cat	4 levels	[CONV, MEA1, MEA2, AEA]
	Total categorical variables		1	
	Engine Bypass Ratio	cont	1	[9, 15] (–)
	Engine <i>X</i>	cont	1	[–0.98, –0.80] (<i>m</i>)
	Engine <i>Z</i>	cont	1	[–0.39, –0.21] (<i>m</i>)
	Total continuous variables		3	
	Total relaxed variables		7	
subject to	Maximum Take-Off Weight ≤ 39058.5 <i>kg</i>		1	
	Take-Off Field Length ≤ 1500 <i>m</i>		1	
	Landing Field Length ≤ 1400 <i>m</i>		1	
	Cumulative Noise ≤ 263 <i>dB</i>		1	
	Total constraints		4	

DOE points) and JPAD optimization.

Regarding the results obtained with SEGOMOE (the workflow shown in Figure 3), green crosses represent all the evaluated points fulfilling the optimization constraints (among the 81 points computed with this approach). The global PF, indicated by the 15 blue circles, was derived from combining both databases obtained by JPAD optimizer and SEGOMOE. This front is composed by 4 points from JPAD optimizer and 11 points from SEGOMOE optimizer. Among the points on the PF, the best solution depends on the designer’s preferences. For instance, the point with the minimum difference between costs and savings (the lower one) is always an optimum choice in terms of economic profit. However, it may not be the optimal solution concerning other variables such as CEI and SAR. In fact, several solutions that offer significant benefits in terms of emissions, SAR, and MTOW may be disregarded in the pursuit of achieving lower costs. Similarly, there are points that offer maximum benefits in terms of other variables but may not be desirable due to their higher costs. When examining opposite solutions on the PF, the selection of engine BPR and OBS architecture can vary depending on the performance objective to be maximized: for example, a BPR of 9.0 with a conventional OBS architecture (CONV) or a BPR of 15.0 with an advanced electric aircraft (AEA) OBS architecture. Opting for lower BPR and a moderate level of electrification (representing state-of-the-art technologies) helps reduce retrofitting costs while achieving moderate performance improvements. Conversely, increasing the level of retrofitting by adopting advanced engine technology and full OBS electrification (beyond state-of-the-art) can significantly enhance overall performance metrics like SAR and CEI, but at the cost of higher retrofitting expenses. For instance, considering a higher BPR and advanced electrification, emissions are reduced (CEI decreases from 1 to 0.86), MTOW slightly decreases (around -3.1% compared to the baseline), and SAR increases (around +25% compared to the baseline). As a result, the costs minus savings increase to 0.44 million euros per year, indicating a loss for an airline operating the aircraft under the considered scenarios.

In this application, two approaches were compared, both providing solutions located on the PF. For this case, SBO with JPAD appears to be a bit more costly - in terms of

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

number of function evaluations - than BO with SEGOMOE and ends up with a smaller number of feasible solutions. Nevertheless, this approach is less sensitive to a change in optimization problem (in terms of objectives and constraints) as it relies on an initial database whereas SEGOMOE enrichment process is tailored for a unique optimization.

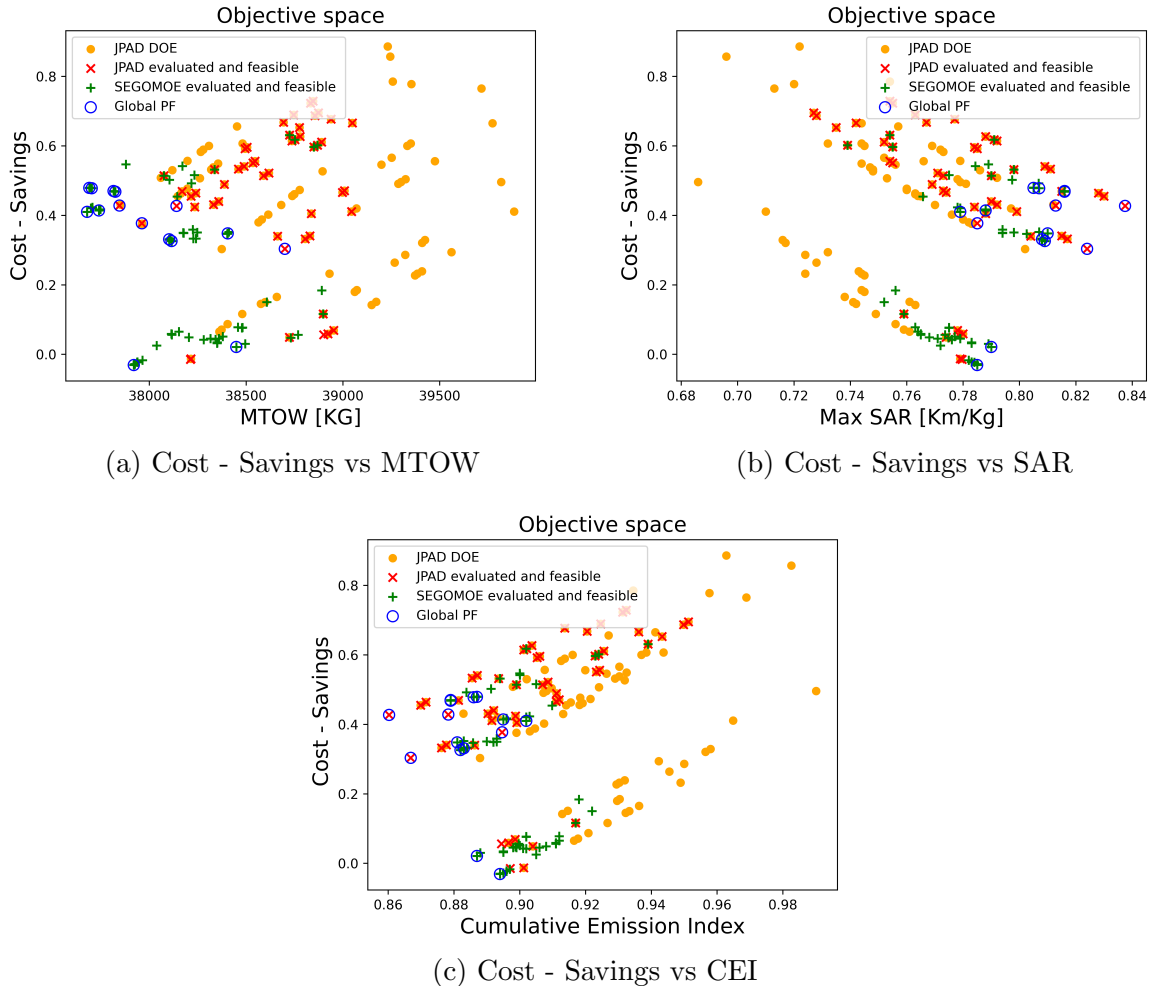


Figure 5: Different PF using four objectives: 108 DOE points for JPAD (orange dot), 15 points (blue circle) on the final PF combining JPAD (red cross) and SEGOMOE (green cross) databases.

3.2 Design of an aircraft family

This application case is part of the upgrade-driven stream and aims at designing a family of three business jet aircraft considering commonality options in the MDO workflow [5]. The commonality options enable sharing one or more aircraft components between the aircraft in the family: by sharing components, design, certification, production, and maintenance costs are reduced. However, operating costs might be increased due to the use of components not designed for the typical flight conditions.

Shareable components include wings, empennage (horizontal and vertical tail), engines, on-board systems (OBS), and landing gears, shown in Figure 6. Next to the commonality choices there are three design variables per wing: leading edge sweep, rear span loca-

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

tion (determining flap size), and thickness-to-chord ratio. These three design variables per wing are inactive if the associated wing is a shared wing. Each aircraft family is optimized for two objectives: direct operating costs (DOC), representing the impact on performance, and Original Equipment Manufacturer (OEM) non-recurring costs, representing the potential benefit of commonality. Table 2 presents the optimization problem in more details.

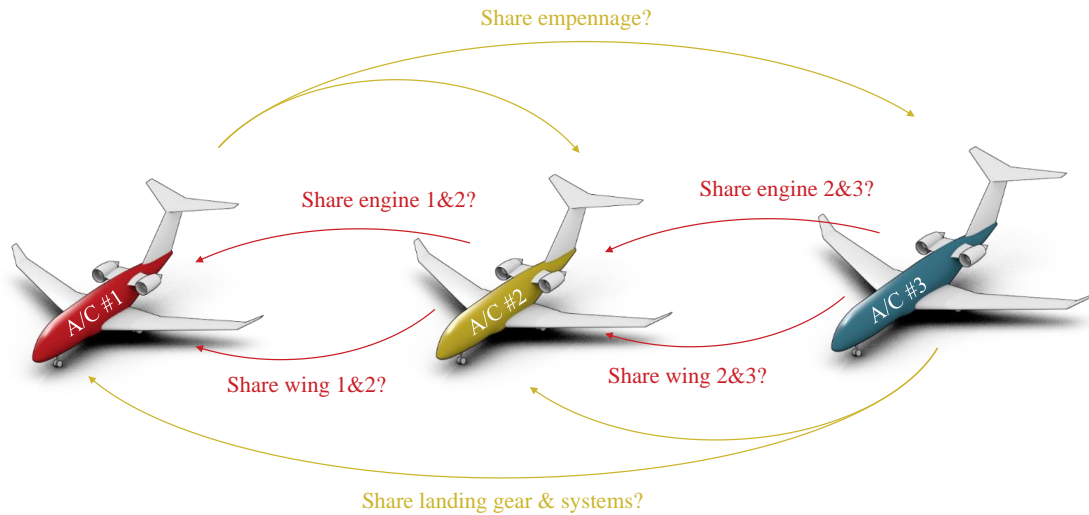


Figure 6: Visualization of the aircraft family including commonality sharing decisions.

The design space is modeled using the Architecture Design Space Graph (ADSG) method [51] implemented in the ADORE tool [52], both developed during the project. MultiLinQ is used to couple the generate architecture instances to the central data schema used in the MDO workflow [5]. The MDO workflow consists of two levels: an aircraft-level workflow (shown in Figure 7) applying commonality decisions and sizing one aircraft at a time, and a family-level workflow converging the three aircraft-level workflows and performing cost calculation on the family-level. For a more detailed overview of the design space model, coupling to the MDO workflow and implementation of the MDO workflow for the family design application case, the reader is referred to [3, 5].

For this application case SEGOMOE is used as optimization algorithm due to the need to handle hierarchical, mixed-discrete design variables, and the need to minimize the number of function evaluations as one family evaluation can take up to 2 hours. Hierarchical design variables were handled using the imputation method, where inactive variables are replaced by the mean value to prevent duplicate design vectors [27]. SEGOMOE was accessed through an ask-tell API implemented in WhatsOpt running on a server at ONERA's premises as described in Section 2.4. For a more detailed description of how SEGOMOE was coupled to ADORE and the MDO workflow, the reader is referred to [3].

First, a 50-point DOE was executed to create the initial database of design points for SEGOMOE and to verify the correct behavior of the MDO workflow. Then, SEGOMOE was used to generate an addition of 18 infill points to explore the design space and extend the Pareto front. Figure 8 shows the main Pareto front, with infill points shown in red. Several example families are shown with colors indicating the originating family member (colors are defined in Figure 6). As can be seen, when no components are shared

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

Table 2: Definition of the aircraft family design problem.

	Function/variable	Nature	Quantity	Range
Minimize	Direct Operating Costs	cont	1	
Minimize	OEM Non-Recurring Costs	cont	1	
	Total objectives:		2	
with respect to	engine commonality 1&2	cat	2 levels	sharing yes/no
	engine commonality 2&3	cat	2 levels	sharing yes/no
	wing commonality 1&2	cat	2 levels	sharing yes/no
	wing commonality 2&3	cat	2 levels	sharing yes/no
	landing gear commonality 1&2	cat	2 levels	sharing yes/no
	landing gear commonality 2&3	cat	2 levels	sharing yes/no
	OBS commonality 1&2	cat	2 levels	sharing yes/no
	OBS commonality 2&3	cat	2 levels	sharing yes/no
	empennage commonality 1&3	cat	2 levels	sharing yes/no
	empennage commonality 3&2	cat	2 levels	sharing yes/no
	Total categorical variables:		10	
	Leading edge sweep*	cont	3	[30.0, 42.0] (deg)
	Rear span location*	cont	3	[0.72, 0.82] (%chord)
	Wing thickness/cord ratio*	cont	3	[0.06, 0.11] (-)
	* only active if the corresponding wing is not shared			
	Total continuous variables:		9	
	Total relaxed variables:		29	
subject to	Balanced Field Length ≤ 1524 m		1	
	Landing Field Length ≤ 762 m		1	
	Total constraints:		2	

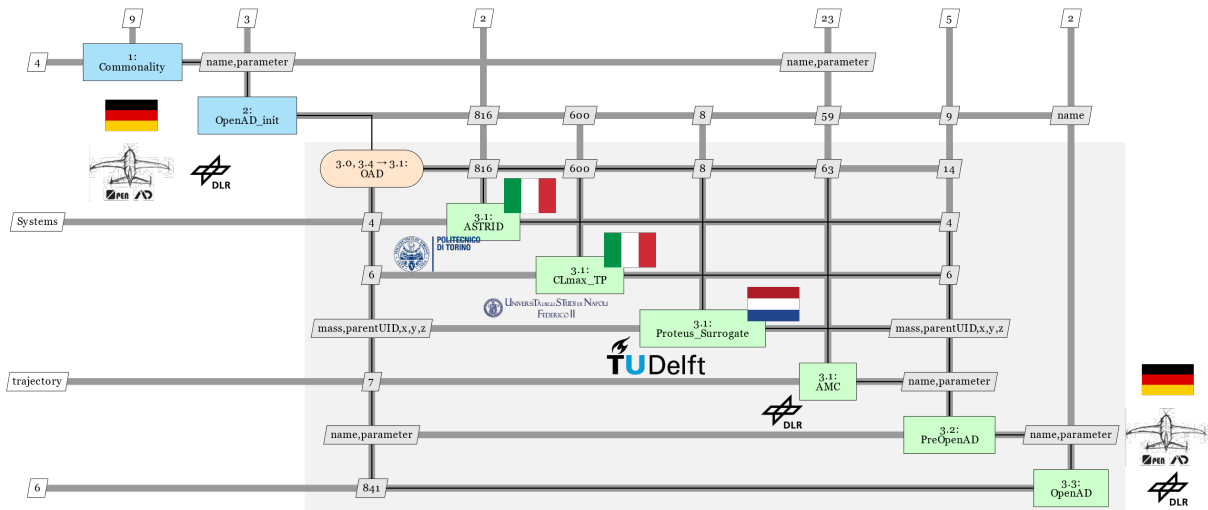


Figure 7: MDA for family aircraft design.

(family #1) the OEM non-recurring costs are high whereas the operating costs are low, because all aircraft components are used at the operating points they are designed for. Introducing more component sharing reduces OEM NRC: family #62 achieves the lowest operating costs at a reduced NRC. Family #54 represents the opposite extreme: the lowest NRC coupled with the highest operating costs, as achieved by a high number of shared

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

components. Family #30 represents a good compromise, at a reduced NRC compared to family #62, with only a moderate increase in DOC.

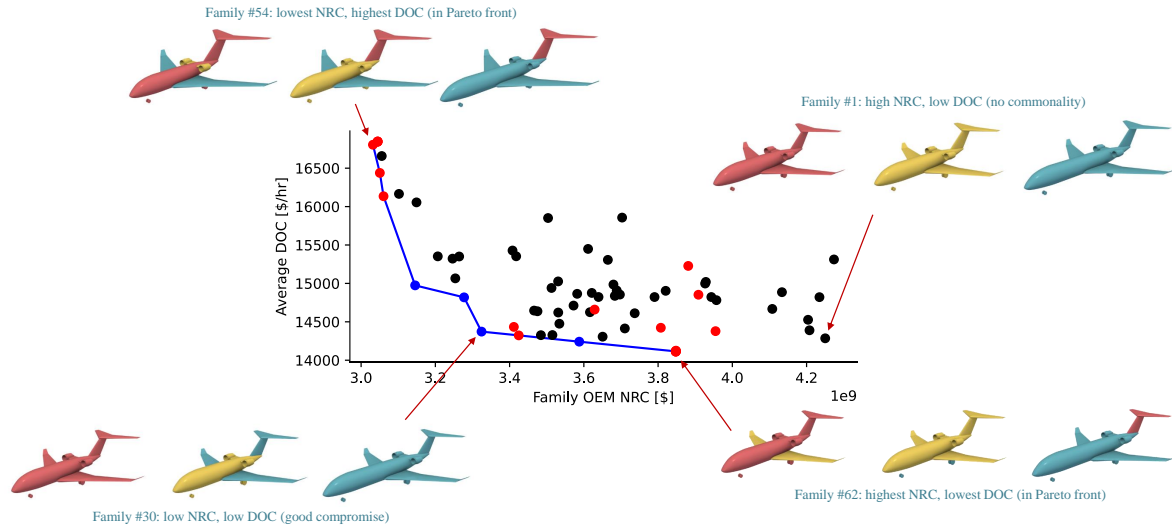


Figure 8: Results of the aircraft family design, showing the Pareto front for minimization of Original Equipment Manufacturer Non-Recurring Cost and Direct Operating Cost and several families. The initial DOE is given by the 50 black dots. Infill points generated by SEGOMOE are shown in red. Colors correspond to the originating aircraft as defined in Figure 6.

3.3 Aircraft design with supply chain, performance and manufacturing process

This application case is part of the Production stream and aims at accounting the supply chain aspects in the aircraft MDO workflow, in addition with design and manufacturing aspects. The application focuses on the design of a horizontal tail plane (HTP) made by different materials and produced by different supply chains, combination of enterprises. The different solutions are characterized by different impact on aircraft performance (e.g. mass, aerodynamic efficiency), different manufacturing processes and different combinations of supply chains.

The objective is to have the ability to select the best HTP to produce by trading performance vs manufacturing costs vs supply chain performance. In this study, HTP structure breakdown includes skin, stringers, spars and ribs that impact the three domains:

- the manufacturing domain deals with the choices of materials, manufacturing and assembly processes feasible for the aircraft components. For each combination of materials, a Technology Factor (TF), a dimensionless number ranging from 0 to 1, is produced.
- the design domain deals with the estimation of the aircraft performance. The Technology Factor is considered in the evaluation as it quantifies the impact that materials, manufacturing and assembly processes, have on the mass and drag of the HTP and, consequently, on the fuel consumption of the vehicle in cruise. The estimation of the fuel consumption is the main output of the design domain.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

- the supply chain domain estimates the production performance by characterizing the enterprises involved in the aircraft production and assembly. Here, each component of the HTP is characterized by a Production Quantity (PQ) that indicates the percentage of this component that each enterprise has to produce. The cost, time, quality and risk are the main outputs of this domain.

For more details regarding this study, the reader can refer to [6, 8, 53]. Figure 9 proposes a sketch of the interaction between the three domains to build the MDA.

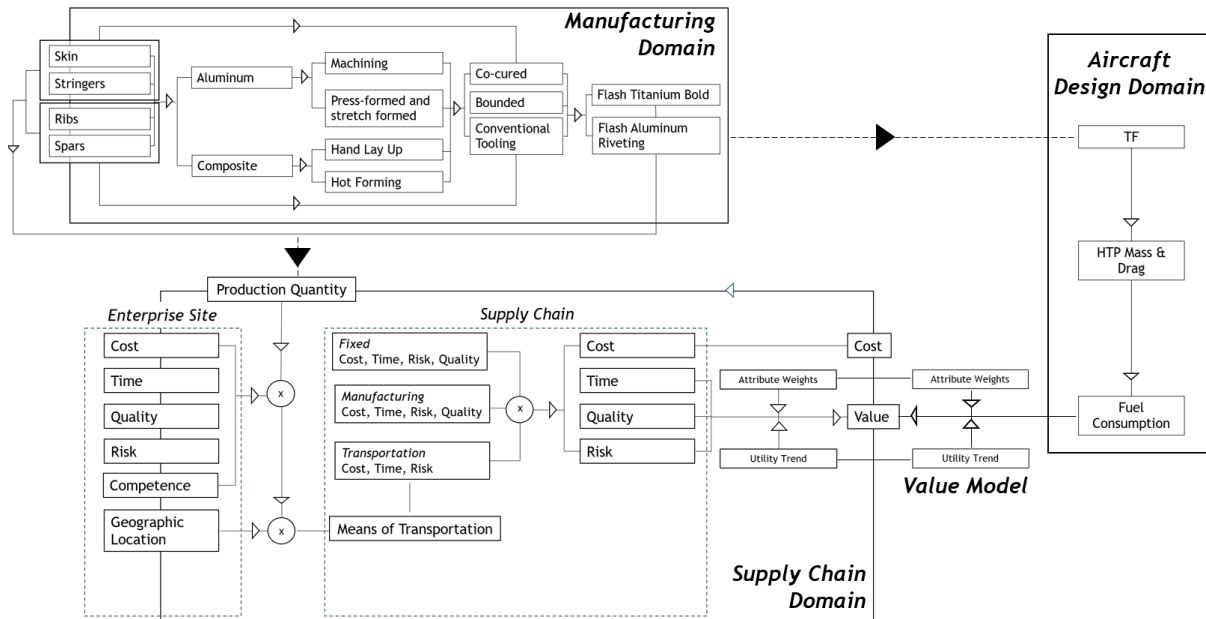


Figure 9: MDA including Manufacturing, Supply Chain and Aircraft Design Variables.

In a first step, the focus was on the supply chain aspect in order to define the formalization, the settings of the optimizer, and the settings for remote calls. During this step, the initial problems are used to calculate the full enumeration of solutions in order to compare the performance of the optimizer with the true Pareto Front of the database. Several increasingly complex problems were tackled, starting with a reduced number of components (for instance only skins and stringers), limited PQ numbers, and limited production sites. Eventually an optimization problem considering all the HTP components and all possible supply chain sites was successfully tackled.

In a second step, the activities focus on integrating the manufacturing options in the overall optimization problem. It requires to add more variables, namely the choice of materials and their associated manufacturing processes for each HTP component. In addition, two constraints have now to be considered:

- the material incompatibility, that aims at discarding all the unfeasible combinations of material and manufacturing processes for the HTP components (for instance, an HTP can not be made using both skin in Aluminium - Machining and ribs in Thermoset - Hand Lay Up);
- the minimal supply chain site competence that checks that no combination of skin, spar, stringer and ribs (depending of their material and manufacturing process) can be produced on a site with a competence of 0.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

Table 3 presents the optimization problem in more details. Five objectives are considered, Cost, Risk, Time and Quality that are related to the supply chain domain and Fuel Consumption that is connected to the overall aircraft design and manufacturing domain. The variables represent all the possible choices of the HTP components characteristics and are treated as categorical variables in the optimization problem. Two types of variables are considered for each HTP component:

- the choice of production and assembly site. Here the PQ have been frozen to 1, but all the possible production sites are available. Each component can therefore be produced among 21 different sites, either belonging to the OEM (Original Equipment Manufacturer) or to a supplier or sub-suppliers.
- the choice of materials and their associated manufacturing processes. Depending of the component, several combinations of material and manufacturing process can be selected (up to 6 for skin). For instance, choices could be Aluminium - Machining, Thermoset - Hand Lay Up or Thermoset - Infusion Process.

Finally, the two constraints described earlier are included.

Table 3: Definition of the complete problem.

	Function/variable	Nature	Quantity	Range
Minimize	Cost	cont	1	
Minimize	Risk	cont	1	
Minimize	Time	cont	1	
Minimize	Fuel Burn	cont	1	
Maximize	Quality	cont	1	
	Total objectives		5	
with respect to	skin prod. location	cat	21 levels	geographic sites
	spar production location	cat	21 levels	geographic sites
	stringer production location	cat	21 levels	geographic sites
	rib production location	cat	21 levels	geographic sites
	skin Material & Manuf. process	cat	6 levels	Alu-Machining,...
	spar Material & Manuf. process	cat	5 levels	Alu-Machining,...
	stringer Material & Manuf. process	cat	4 levels	Alu-Machining,...
	rib Material & Manuf. process	cat	5 levels	Alu-Machining,...
	Total categorical variables		8	
	Total relaxed variables		104	
subject to	Material incompatibility		1	
	Supply chain sites		1	
	Total constraints		2	

Regarding optimization process, a 300-point DOE was executed to create the initial database of design points for SEGOMOE and to verify the correct behavior of the MDO workflow. Then, SEGOMOE was used to generate 450 additional infill points to explore the design space and extend the Pareto front. The main results of the optimization process are depicted in Figure 10 and Figure 11. Figure 10 presents the PF database obtained at the end of the optimization run with 450 iterations. The initial DOE points are represented by orange dots and can provide information concerning the workflow

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

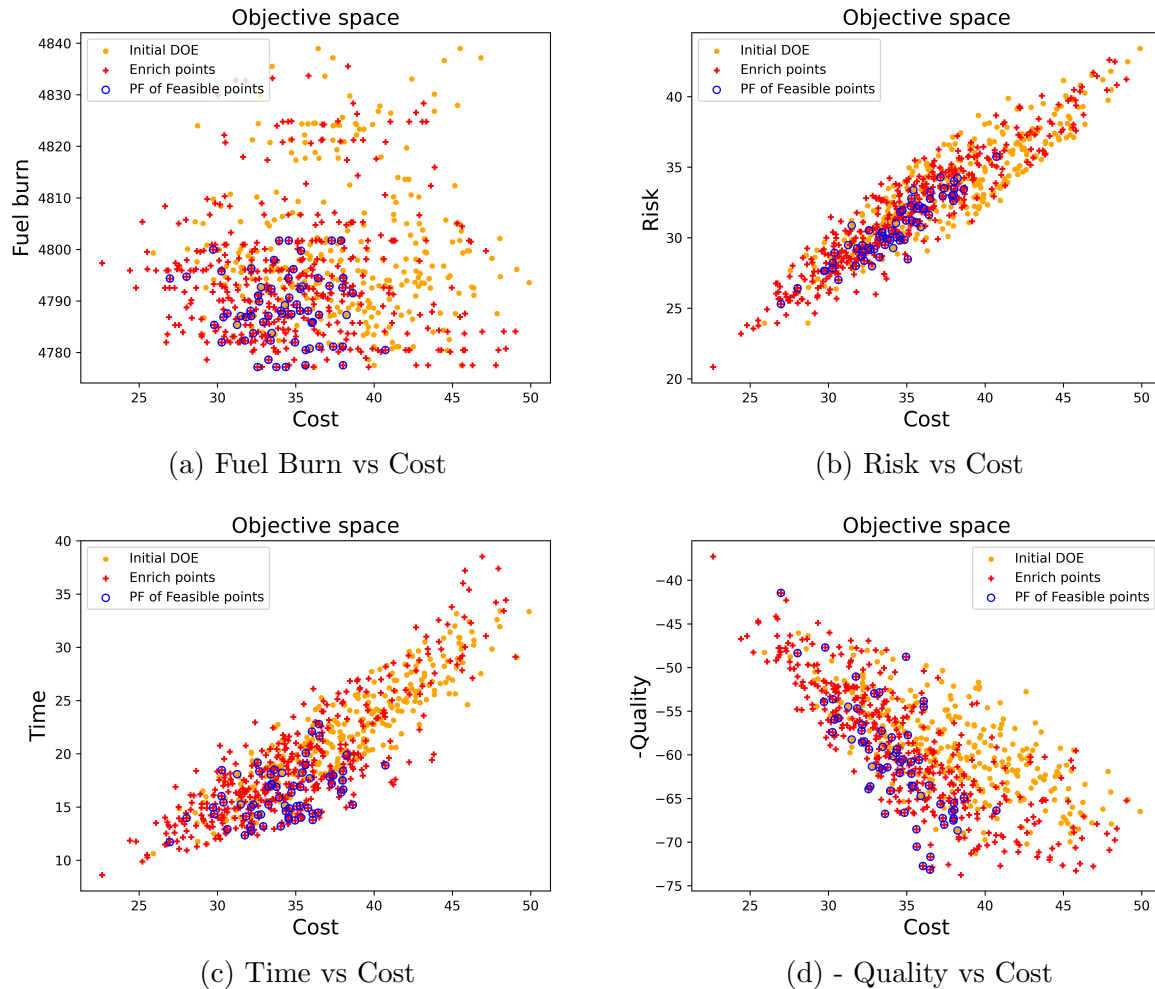


Figure 10: Database for five objectives: initial DOE of 300 points (orange dots), 450 enrichment points (red crosses) and, from the complete database (750 points) 65 feasible points on the PF are identified by the blue circles.

behavior. The enrichment process of SEGOMOE added the 450 red crosses points that extended the database in direction of the Pareto Front (as it can be seen, for instance, on the -Quality vs Cost Figure 10d). As the optimization is considering two constraints, not all the database points are feasible and, at the end of the process, 65 points are identified on the PF (all highlighted with blue circles).

As indicated in Algorithm 1, a post-processing step is then performed applying NSGA-II algorithm to construct the predicted PF (using the GP models built upon the database). Figure 11 illustrates the results of this step with the PF database still depicted by blue circles. The predicted PF is represented by green diamonds, consisting of 33 candidates. It can be observed that both databases appear intermingled, indicating that the PF database already exhibits promising performance. Nevertheless, as an additional step, all the points belonging to the predicted PF have been evaluated with the workflow and the outputs are depicted in black diamonds. Two pieces of information can be derived from this evaluated PF. Firstly, the evaluated candidates closely align with the predicted values (with the exception of a few), indicating the accuracy of the GP models, despite being built upon 750 points. Secondly, the PF database appears to maintain its good

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

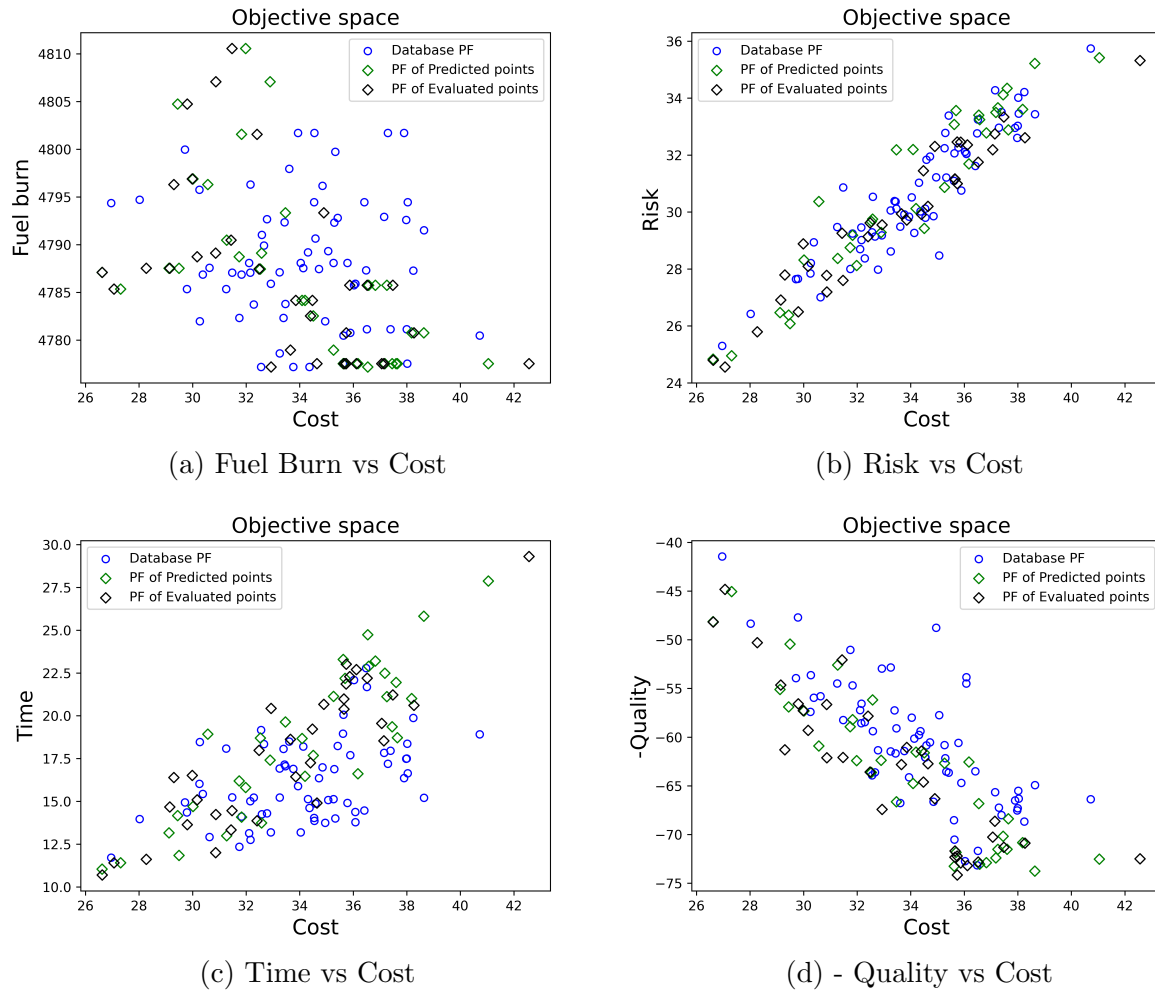


Figure 11: Comparison of PF (predicted and evaluated) with five objectives: 65 points from the PF database (blue circles), 33 points from the predicted PF (green diamonds) or evaluated with the workflow (black diamonds). The predicted and evaluated points are quite close to each other and complementary to the PF database.

performance, characterized by its proximity to the evaluated PF. To verify this, both the PF database and evaluated PF were sorted, revealing that 52 points (out of 65) from the PF database and 26 points (out of 33) from the evaluated PF contribute to the final combined PF. As expected, the post-processing step of SEGOMOE successfully populates the PF identified by the algorithm, thereby enriching the available choices for the application stakeholders.

4 CONCLUSIONS AND PERSPECTIVES

Within the frame of AGILE 4.0 project, multi-objective problems involving mixed integer variables have been successfully solved using SBO and BO optimizers based on two associated frameworks JPAD Optimizer and SEGOMOE. In terms of function evaluations, the number of calls to the expensive black boxes (objectives and constraints) is still small compared to the number of design variables involved. SEGOMOE has been successfully applied to three realistic problems involving 2 to 5 objective functions with mixed integer

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

variables. The use of continuous relaxation strategy to handle the categorical variables has been coupled with PLS to reduce the increase of dimension allowing to consider up to 104 choices.

To apply BO to even more complex multidisciplinary processes, several developments need to be addressed. Future work will focus on extending the handling of hierarchical and mixed discrete variables while improving computational efficiency for larger databases. Specifically, with regards to mixed integer variables, the intention is to integrate recent advancements done within the SMT toolbox concerning mixed correlation kernels [9] or hierarchical kernels [49] for Gaussian processes into BO and validate their effectiveness. These studies will be conducted within the context of the Horizon Europe COLOSSUS project ⁵. The COLOSSUS project aims to develop a system-of-systems design methodology that facilitates the integrated optimization of aircraft, operations, and business models. This methodology will be applied to intermodal transport and wildfire-fighting scenarios.

ACKNOWLEDGEMENTS

This work is part of the activities of ONERA - ISAE - ENAC joint research group. The research presented in this paper has been performed within the framework of the AGILE 4.0 project (Towards Cyber-physical Collaborative Aircraft Development) and has received funding from the European Union Horizon 2020 Programme under grant agreement n°815122. Some methodology developments have been performed in the framework of the COLOSSUS project (Collaborative System of Systems Exploration of Aviation Products, Services and Business Models) and has received funding from the European Union Horizon Programme under grant agreement n°101097120. The authors are also grateful to the partners of the AGILE 4.0 consortium for their contribution and feedback.

REFERENCES

- [1] A. Chan, A. F. Pires, and T. Polacsek. Trying to elicit and assign goals to the right actors. In *Conceptual Modeling: 41st International Conference, ER 2022*, 2022.
- [2] M. Fioriti, P. D. Vecchia, and G. Donelli. Effect of progressive integration of on-board systems design discipline in an mda framework for aircraft design with different level of systems electrification. *Aerospace*, 9(3), 2022.
- [3] J. Bussemaker, L. Boggero, and B. Nagel. The AGILE4.0 project: MBSE to support cyber-physical collaborative aircraft development. In *33rd Annual INCOSE International Symposium*, Honolulu, HI, USA, 2023. To be published.
- [4] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013. doi:10.2514/1.J051895.
- [5] J. H. Bussemaker, P. D. Ciampa, J. Singh, M. Fioriti, C. C. D. L. Hoz, Z. Wang, D. Peeters, P. Hansmann, P. D. Vecchia, and M. Mandorino. Collaborative design of a business jet family using the AGILE 4.0 MBSE environment. In *AIAA AVIATION 2022 Forum*, 2022. doi:10.2514/6.2022-3934.

⁵<https://colossus-sos-project.eu/>

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

- [6] G. Donelli, J. M. Mello, F. I. Odaguil, T. van der Laan, T. Lefebvre, N. Bartoli, L. Boggera, and B. Nagel. Value-driven systems engineering approach addressing manufacturing, supply-chain and aircraft design in the decision-making process. In *INCOSE 33rd Annual International Symposium 2023*, 2023.
- [7] A. Chan, A. Fernandes Pires, T. Polacsek, and S. Roussel. The aircraft and its manufacturing system: From early requirements to global design. In *Advanced Information Systems Engineering: 34th International Conference, CAiSE 2022, Leuven, Belgium, June 6–10, 2022, Proceedings*, pages 164–179. Springer, 2022.
- [8] U. Merola, G. Donelli, T. Lefebvre, N. Bartoli, J. M. Mello, F. I. Odaguil, T. van der Laan, and B. Nagel. Value-driven optimization campaign addressing manufacturing, supply chain and overall aircraft design domains in the early development stage. In *INCOSE 33rd Annual International Symposium 2023*, 2023.
- [9] P. Saves, Y. Diouane, N. Bartoli, T. Lefebvre, and J. Morlier. A mixed-categorical correlation kernel for Gaussian process. *arXiv preprint*, 2022. doi:<https://doi.org/10.48550/arXiv.2211.08262>.
- [10] R. Grapin, Y. Diouane, J. Morlier, N. Bartoli, T. Lefebvre, P. Saves, and J. Bussemaker. Regularized infill criteria for multi-objective bayesian optimization with application to aircraft design. In *AIAA AVIATION 2022*, 2022.
- [11] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.
- [12] R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.
- [13] F. Nicolosi, A. De Marco, L. Attanasio, and P. D. Vecchia. Development of a java-based framework for aircraft preliminary design and optimization. *Journal of Aerospace Information Systems*, 13(6):234–242, 2016.
- [14] V. Trifari, M. Ruocco, V. Cusati, F. Nicolosi, and A. De Marco. Java framework for parametric aircraft design–ground performance. *Aircraft Engineering and Aerospace Technology*, 89(4):599–608, 2017.
- [15] J. B. Kollat and P. M. Reed. Comparison of multi-objective evolutionary algorithms for long-term monitoring design. In *Impacts of Global Climate Change*, pages 1–11. 2005.
- [16] M. Sasena. *Flexibility and efficiency enhancements for constrained global design optimization with Kriging approximations*. PhD thesis, University of Michigan, 2002.
- [17] C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, 2006.
- [18] D. G. Krige. A statistical approach to some basic mine valuation problems on the witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6):119–139, 1951.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

- [19] D. Bettebghor, N. Bartoli, S. Grihon, J. Morlier, and M. Samuelides. Surrogate modeling approximation using a mixture of experts based on EM joint estimation. *Structural and Multidisciplinary Optimization*, 43(2):243–259, 2011. ISSN 1615-147X. URL <http://dx.doi.org/10.1007/s00158-010-0554-2>. 10.1007/s00158-010-0554-2.
- [20] R. P. Liem, C. A. Mader, and J. R. R. A. Martins. Surrogate models and mixtures of experts in aerodynamic performance prediction for mission analysis. *Aerospace Science and Technology*, 43:126–151, 2015. doi:10.1016/j.ast.2015.02.019.
- [21] M. A. Bouhleb, N. Bartoli, A. Otsmane, and J. Morlier. Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952, 2016. ISSN 1615-1488. doi:10.1007/s00158-015-1395-9. URL <http://dx.doi.org/10.1007/s00158-015-1395-9>.
- [22] M. A. Bouhleb, N. Bartoli, A. Otsmane, and J. Morlier. An improved approach for estimating the hyperparameters of the kriging model for high-dimensional problems through the partial least squares method. *Mathematical Problems in Engineering*, 2016, 2016.
- [23] R. Priem, H. Gagnon, I. Chittick, S. Dufresne, Y. Diouane, and N. Bartoli. An efficient application of bayesian optimization to an industrial mdo framework for aircraft design. In *AIAA AVIATION 2020 FORUM*, page 3152, 2020.
- [24] P. Saves, N. Bartoli, Y. Diouane, T. Lefebvre, J. Morlier, C. David, E. Nguyen Van, and S. Defoort. Multidisciplinary design optimization with mixed categorical variables for aircraft design. In *AIAA SCITECH 2022 Forum*, page 0082, 2022.
- [25] N. Bartoli, T. Lefebvre, S. Dubreuil, R. Olivanti, R. Priem, N. Bons, J. R. Martins, and J. Morlier. Adaptive modeling strategy for constrained global optimization with application to aerodynamic wing design. *Aerospace Science and Technology*, 90:85–102, 2019. doi:<https://doi.org/10.1016/j.ast.2019.03.041>.
- [26] T. Lefebvre, N. Bartoli, S. Dubreuil, M. Panzeri, R. Lombardi, P. D. Vecchia, L. Stingo, F. Nicolosi, K. Anisimov, A. Savelyev, A. Mirzoyan, and A. Isyanov. Enhancing optimization capabilities using the agile collaborative mdo framework with application to wing and nacelle design. *Progress in Aerospace Sciences*, 2020. doi:10.1016/j.paerosci.2020.100649.
- [27] J. H. Bussemaker, N. Bartoli, T. Lefebvre, P. D. Ciampa, and B. Nagel. Effectiveness of surrogate-based optimization algorithms for system architecture optimization. In *AIAA AVIATION 2021 FORUM*, page 3095, 2021. doi:10.2514/6.2021-3095.
- [28] E. C. Garrido-Merchán and D. Hernández-Lobato. Dealing with categorical and integer-valued variables in bayesian optimization with gaussian processes. *Neurocomputing*, 380:20–35, 2020.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

- [29] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. Karro, and D. Sculley. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1487–1495, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3097983.3098043. URL <https://doi.org/10.1145/3097983.3098043>.
- [30] M. A. Bouhleb, J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins. A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, 135, 2019. doi:10.1016/j.advengsoft.2019.03.005.
- [31] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation*, 7(2):117–132, 2003.
- [32] M. T. Emmerich, K. C. Giannakoglou, and B. Naujoks. Single-and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, 2006.
- [33] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21:345–383, 2001.
- [34] A. A. Rahat, R. M. Everson, and J. E. Fieldsend. Alternative infill strategies for expensive multi-objective optimisation. In *Proceedings of the genetic and evolutionary computation conference*, pages 873–880, 2017.
- [35] M. J. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in optimization and numerical analysis*, pages 51–67. Springer, 1994.
- [36] D. Kraft et al. *A software package for sequential quadratic programming*. DFVLR Obersfaffehofen, Germany, 1988.
- [37] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [39] J. Blank and K. Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.
- [40] B. Boden, J. Flink, N. Först, R. Mischke, K. Schaffert, A. Weinert, A. Wohlan, and A. Schreiber. Rce: An integration environment for engineering and science. *softwarex* 15, 100759 (2021), 2021.
- [41] E. Baalbergen, E. Moerland, W. Lammen, and P. Ciampa. Methods to support efficient collaboration for competitive aircraft design. In *6th CEAS Aerospace Europe Conference*, 2017.

N. Bartoli, T. Lefebvre, R. Lafage, P. Saves, Y. Diouane, J. Morlier, J.H. Bussemaker, G. Donelli, Joao Marcos Gomes de Mello, M. Mandorino and P. Della Vecchia

- [42] E. Baalbergen, J. Vankan, L. Boggero, J. H. Bussemaker, T. Lefebvre, B. Beijer, A.-L. Bruggeman, and M. Mandorino. Advancing cross-organizational collaboration in aircraft development. In *AIAA AVIATION 2022 Forum*, page 4052, 2022.
- [43] M. Alder, E. Moerland, J. Jepsen, and B. Nagel. Recent advances in establishing a common language for aircraft design with cpacs. 2020.
- [44] R. Lafage, S. Defoort, and T. Lefebvre. Whatsopt: a web application for multidisciplinary design analysis and optimization. In *AIAA Aviation 2019 Forum*, page 2990, 2019.
- [45] S. Chiesa, G. A. Di Meo, M. Fioriti, G. Medici, and N. Viola. Astrid–aircraft on board systems sizing and trade-off analysis in initial design. *Research and Education in Aircraft Design–READ 2012*, 2012.
- [46] P. Della Vecchia, L. Stingo, F. Nicolosi, A. De Marco, G. Cerino, P. D. Ciampa, P. S. Prakasha, M. Fioriti, M. Zhang, A. Mirzoyan, et al. Advanced turboprop multidisciplinary design and optimization within agile project. In *2018 Aviation Technology, Integration, and Operations Conference*, page 3205, 2018.
- [47] P. Della Vecchia, M. Mandorino, V. Cusati, and F. Nicolosi. Retrofitting cost modeling in aircraft design. *Aerospace*, 9(7):349, 2022.
- [48] M. Mandorino, P. Della Vecchia, S. Corcione, F. Nicolosi, V. Trifari, G. Cerino, M. Fioriti, C. Cabaleiro De La Hoz, T. Lefebvre, D. Charbonnier, et al. Multidisciplinary design and optimization of regional jet retrofitting activity. In *AIAA AVIATION 2022 Forum*, page 3933, 2022.
- [49] P. Saves, R. Lafage, N. Bartoli, Y. Diouane, J. Bussemaker, T. Lefebvre, J. T. Hwang, J. Morlier, and J. R. R. A. Martins. SMT 2.0: A surrogate modeling toolbox with a focus on hierarchical and mixed variables gaussian processes. *arXiv preprint*, 2023. doi:<https://doi.org/10.48550/arXiv.2305.13998>.
- [50] N. Werter and R. De Breuker. A novel dynamic aeroelastic framework for aeroelastic tailoring and structural optimisation. *Composite Structures*, 158:369–386, 2016.
- [51] J. H. Bussemaker, P. D. Ciampa, and B. Nagel. System architecture design space exploration: An approach to modeling and optimization. In *AIAA Aviation 2020 Forum*, page 3172, 2020. doi:10.2514/6.2020-3172.
- [52] J. Bussemaker, L. Boggero, and P. D. Ciampa. From system architecting to system design and optimization: A link between mbse and mdao. In *INCOSE International Symposium*, volume 32, pages 343–359. Wiley Online Library, 2022. doi:10.1002/iis2.12935.
- [53] G. Donelli, P. D. Ciampa, T. Lefebvre, N. Bartoli, J. M. Mello, F. I. Odaguil, and T. van der Laan. Value-driven model-based optimization coupling design-manufacturing-supply chain in the early stages of aircraft development: Strategy and preliminary results. In *AIAA AVIATION 2022 Forum*, page 3723, 2022.



AN EXPLORATORY STUDY OF OPEN-SOURCE FRAMEWORKS FOR MDAO

Roberto di Giuseppe^{1,2}, Scott Delbecq¹ and Valérie Budinger¹, Vincent Pauvert²

1: ISAE-SUPAERO, University of Toulouse, Toulouse, France
roberto.di-giuseppe@isae-supero.fr

2: Safran Tech - Safran Group, Blagnac, France

Abstract. *Multidisciplinary Design Analysis and Optimization (MDAO) is a key methodology used by engineers and researchers to explore and optimize engineering systems composed of multiple disciplines and couplings. The techniques of this field are particularly relevant for designing complex systems such as aeronautical systems. Today MDAO is adopted worldwide and several open-source tools are available. Although the underlying features are standard, these tools differ for some typical features, such as numerical methods, visualization tools or available integrated/external optimizers. Choosing among these tools for a given engineering application is not straightforward. This article focuses on three frameworks that arise from different environments and respond to different needs: OpenMDAO, GEMSEO, CoSApp. A study case is first proposed as a reference problem and then analyzed. The three MDAO architectures implemented (MDF, IDF and NVH) show a similar range of performance. Thus the need to test the frameworks on a more complex problem to obtain more discriminating results and specific features is highlighted. After a brief presentation of the three frameworks, several quantitative criteria are computed, such as the lines of code required to define the multidisciplinary optimization problem on the three platforms and the performance of the solving process. In addition, qualitative criteria concerning the reconfiguration of the MDO process and the problem visualization are assessed.*

Keywords: MDAO, design optimization, system analysis, aeronautical systems

1 INTRODUCTION

Nowadays, the growing interest in environmental sustainability pushes academic and industrial entities to search for disruptive technologies that could reduce the carbon footprint of the aviation sector. The electrification of aircraft propulsion systems is one solution investigated. However, the performance of a fully electric aircraft requires a highly integrated design to take advantage of the additional degrees of freedom that offer electrification. Multidisciplinary design analysis and optimization (MDAO) can provide engineers with valuable tools. One essential step in MDAO approach is the formulation of the optimization problem, which requires the designer to write down his intent in a mathematical form by considering the design variables, the objective function and the constraints. In the field of MDAO a large set of architectures exist. Martins recognizes 13 different architectures [1]. The choice of architecture depends strongly on the problem and the models of each discipline, so choosing one of them is not straightforward for an engineer. In the present study, the two well-known architectures MDF and IDF and the new architecture NVH found in the literature are retained and adopted in [2].

The study case used for the test is extracted from [2]. It consists of a design optimization problem targeting the minimization of the motor mass of a high dynamic electro-mechanical actuator. The problem is an interesting option for apprehending and testing several MDO architectures as it consists of a low number of function analyses with a strong coupling with equations.

1.1 State of the art

Different proprietary and open-source platforms are available and the goal of this paper is to test 3 open-source MDAO frameworks on an algebraic problem by evaluating several criteria that describe the main features and performances of a MDAO tool. Various commercial frameworks have been developed by the leading modeling and simulation platforms such as Phoenix Integration's ModelCenter [3], Esteco's modeFRONTIER [4], Noesis Solutions' Optimus [5], Siemens' Heeds [6], Altair's HyperStudy [7], Ansys' optiSLang[8], Dassault Systems Isight [9]. Also, more recent companies have also developed solutions based on their expertise in optimization such as Pseven [10] and VisualDoc [11].

Unlike the proprietary market, open-source frameworks are not as abundant. Beyond the 3 frameworks presented in this paper, RCE is another software known in the MDAO scene. Developed under the leadership of DLR, it is written in JAVA and adopts a component-based approach [12]. One of its main features lies in the GUI. The 3 software used in this work, all coded in Python, are introduced here below:

- OpenMDAO [13]: published in 2015, it has been developed to meet the need for a new MDO framework to address aircraft design challenges at NASA. Its development highlights the strong impact of the derivatives computation method on the performances of the resolution process and focuses on the monolithic MDO architectures for the best possible computational efficiency;
- GEMSEO [14]: published in 2021, GEMSEO derives directly from GEMS software developed as part of the IRT Saint Exupéry MDA-MDO project putting forward the automatic programming of MDO processes, the integration of state-of-the-art algorithms for optimization, design of experiments (DOE), surrogate models and

coupled analyses. GEMSEO fosters the deployment of distributed and multi-level MDO architectures in parallel to monolithic architectures;

- CoSApp [15]: published in 2018, the tool's code is the property of Safran SA and comes from various open-source projects. The framework is designed to be used in industry and help technical departments in system integration to meet product specifications.

2 PRESENTATION OF THE STUDY CASE

The study case is based on an electro-mechanical actuator problem proposed in [2]. The advantage is that it is relatively simple to implement but includes numerical difficulties, making its resolution challenging. The following equations and inequalities represent the problem:

$$T_{em} = J_{mot} \cdot A_{max} \frac{N_{red}}{p} + F_{ema} \frac{p}{N_{red}} \quad (1)$$

$$J_{mot} = J_{mot_{ref}} \cdot \left(\frac{T_{em}}{T_{em_{ref}}} \right)^{\frac{5}{3.5}} \quad (2)$$

$$\Omega_{mot} = \Omega_{mot_{ref}} \cdot \left(\frac{T_{em}}{T_{em_{ref}}} \right)^{-\frac{1}{3.5}} \quad (3)$$

$$\Omega_{mot} \geq V_{max} \cdot \frac{N_{red}}{p} \quad (4)$$

$$M_{mot} = M_{mot_{ref}} \cdot \left(\frac{T_{em}}{T_{em_{ref}}} \right)^{\frac{3}{3.5}} \quad (5)$$

Where T_{em} is the motor electromagnetic torque, J_{mot} is the motor inertia, A_{max} is the maximum actuator acceleration, N_{red} is the reducer gear ratio, p is the screw pitch, V_{max} and F_{ema} are the maximum velocity and external load of the actuator, Ω_{mot} the maximum mechanical speed of the motor and M_{mot} its mass. The motor sizing scaling laws are based on a reference motor where $T_{mot_{ref}}$, $J_{mot_{ref}}$, $\Omega_{mot_{ref}}$, $M_{mot_{ref}}$ are respectively its electromagnetic torque, inertia, maximal mechanical rotational speed and mass. Motor torque Equation (1), motor inertia Equation (2) are two coupled disciplines of the sizing problem whereas motor speed Equation (3), motor speed constraint Equation (4) and motor mass objective Equation (5) are ordinary analysis functions. The motor torque (1) and motor inertia Equation (2) are coupled through motor electromagnetic torque T_{em} and motor inertia J_{mot} . The ball screw efficiency is supposed equal to 1 and independent of the pitch and the axial load.

The design problem consists of minimizing the motor mass M_{mot} , with respect to reducer gear ratio N_{red} , subject to motor speed constraint Equation (4).

The optimization problem formulation is the following:

$$\begin{aligned}
 & \text{minimize} && M_{mot} \\
 & \text{with respect to} && N_{red} \\
 & \text{subject to} && V_{max} \cdot \frac{N_{red}}{p} - \Omega_{mot} \leq 0
 \end{aligned} \tag{6}$$

Its XDSM diagram is shown in Fig.1.

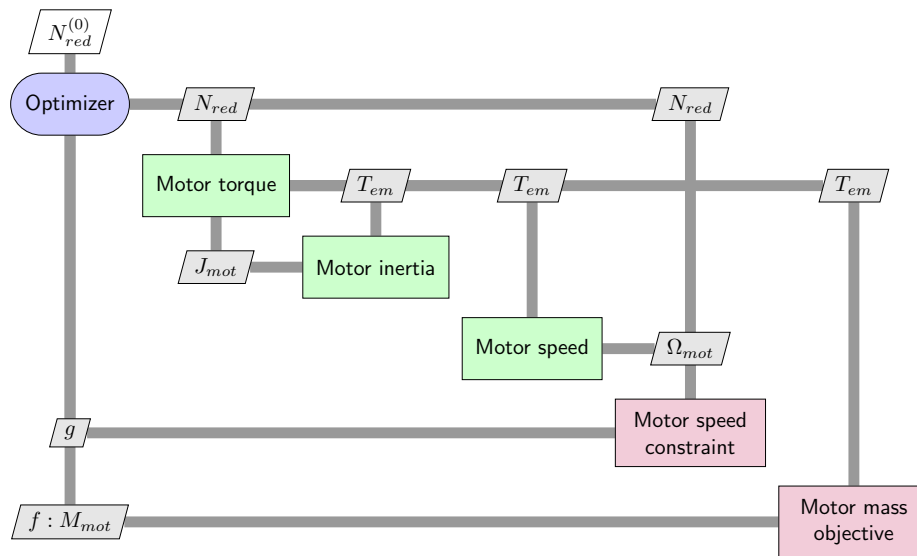


Figure 1: XDSM diagram for the study case [2]

An evaluation of each framework’s performance in resolving this problem is assessed. This evaluation is valid in a specific domain of assumptions. In this work, the main assumptions are here resumed:

- The MDF architecture requires using a numerical solver to resolve the coupled equations of motor torque and motor inertia. In this study, the type of solver is linked to the choice of the optimizer, shown in purple in Fig.1. In the case of gradient-based optimization (Tab.1), a Newton-based solver is used, while for gradient-free optimization a non-linear adapted Gauss-Seidel method is used; In the CoSApp framework, a Newton’s method is used for both cases. The tolerance of the solver is fixed to 10^{-10} meaning that:

$$\text{residual} \leq 10^{-10}$$

The choice of the tolerance and of the MDA solver can have a significant impact on the performance;

- In the case of IDF (Individual Design Feasible), the default tolerances of GEMSEO on equalities and inequalities have been modified to find the exact optimum. For the sake of equity in comparison, these tolerances have been applied to the other platforms and are set to 10^{-10} ;

- The termination tolerance on the optimization algorithm is set to 10^{-8} and the maximum iteration number is equal to 500;
- The packages' versions used are 3.25.0 for OpenMDAO, 0.13.0 for CoSApp and 5.0.0 for GEMSEO.

3 PROPOSITION OF COMPARISON CRITERIA

The study case has been solved in different configurations for each platform by changing the MDAO architecture and the optimization algorithm, as shown in Tab.1. In the case of OpenMDAO and GEMSEO, another test case is analyzed by considering that partial derivatives can be computed analytically or by finite difference approximation.

Table 1: List of architectures and optimizers for the combinations investigated

Architecture	Optimization Algorithm
MDF (Multidisciplinary Feasible)	SLSQP (Gradient-based)
IDF (Individual Design Feasible)	COBYLA (Gradient-free)
NVH (Normalized Variable Hybrid)	

One of the main goals of this work is to investigate meaningful criteria that allow different MDAO frameworks to be tested and compared. A significant distinction can be made between quantitative and qualitative criteria where the former designates some computable parameters and the latter subjective factors that could affect the user experience. The quantitative criteria selected are the following:

- The total number of function evaluations that is the result of a counter increased each time a discipline is executed. As stated in literature [2, 13], this metric can indicate the computational cost of each architecture;
- The number of total derivative evaluations presented only for the analytical gradient-based optimization;
- The execution time of the optimization process is retrievable in OpenMDAO and GEMSEO respectively through the results saved in the driver and in the configuration logger printed by the program. In CoSApp the execution time is obtained by measuring the clock time between the start and the end of the execution command;
- The absolute error that measures the proximity to the exact optimum;
- The list of available optimization algorithms and numerical solvers;
- The number of lines of code needed to define a discipline and the optimization problem.

The qualitative criteria selected are the following:

- Support in the generation of a MDAO process based on established architectures;

- The availability of tools to view and analyze the optimization problem (e.g. diagram XDSM [16] and matrix N2).

3.1 Disciplines and problem architecture

One of the strengths of GEMSEO consists in the support in the definition of an architecture. The user can choose from a list of well-known MDAO architectures and GEMSEO will instantiate the corresponding problem. As can be seen in Tab.2, the definition of a discipline does not require the same number of lines of code in all frameworks. In each tool, a discipline is defined by two Python class methods, one to set it up and the other to compute the results (code in Annexes). In CoSApp, once the input variables are declared in the setup part, the computation function can call them directly. In OpenMDAO and GEMSEO, this is not the case because the inputs are declared but must be applied to other variables in the compute function. OpenMDAO differs from GEMSEO in requiring each variable to be passed in a single line of code. Therefore, CoSApp requires the least lines of code for implementing a discipline.

Regarding optimization problem definition, GEMSEO requires the least number of lines of code in all architectures. There are several reasons for this. In GEMSEO, variables in one discipline are visible in the other disciplines, whereas in the other two frameworks variables have to be connected and this is achieved through the exposure of variables from one component to the higher level in the hierarchical model. In turn, the naming of variables becomes essential. Secondly, OpenMDAO and CoSApp require the definition of the MDA in the MDF architecture, whereas GEMSEO automatically detects it if it is specified when the problem is created, and this can be done in a single line of code. Finally, in CoSApp and OpenMDAO, consistency constraints must be added in the IDF architecture, but this is not the case in GEMSEO, which ensures consistency if the IDF architecture is declared.

Table 2: Count of lines of code

	GEMSEO	OpenMDAO	CoSApp
Discipline	15	20	12
Optimization problem (IDF)	15	41	34
Optimization problem (MDF)	19	52	30
Optimization problem (NVH)	12	34	29

One of the main differences between GEMSEO and OpenMDAO (as CoSApp) is that consistency constraints are embedded in GEMSEO architecture, so the user does not have to formulate them explicitly. This aspect is translated into a gain in implementation time and is directly linked to the qualitative criterion of automatic reconfiguration. Indeed, the effort required on GEMSEO to switch between an IDF and MDF architecture is considerably lower than in the other platforms (see Tab.3).

3.2 Gradient-based optimization

The results presented in Tab.4 and resumed graphically in Fig.2 have been obtained by means of the Sequential Least Squares Programming algorithm (SLSQP), which uses the partial derivatives provided by the user in the Jacobian matrix in each of the disciplines.

Table 3: Steps required to switch from IDF to MDF

	GEMSEO	OpenMDAO and CoSApp
Step 1	Suppression of coupling variables from design space	Suppression of coupling variables from design space
Step 2	Choice of the numerical solver	Suppression of consistency constraints
Step 3		Creation of the MDA
Step 4		Choice of the numerical solver

For both frameworks, the computational time taken to solve the MDF architecture is superior to that required for the IDF and NVH architectures.

In the MDF architecture, GEMSEO computation time is more than 10 times greater than OpenMDAO. Further investigation shows that this result strongly depends on the solver chosen. As a matter of fact, in GEMSEO one can note a difference if a Quasi-Newton method is employed. As for the number of function and derivative evaluations the ratio between the two frameworks is lower (around 2). GEMSEO calls the objective function one more time.

Regarding the IDF architecture, the frameworks show a computation on the order of hundreds of seconds, while the number of functions and derivatives evaluations is greater for GEMSEO with 15 more evaluations. The numbers in round brackets indicate the consistency constraint evaluations that must be explicitly formulated in OpenMDAO, differently from GEMSEO. The number of calls to the objective function differs by 3 evaluations.

Concerning the NVH architecture, the computation time for GEMSEO is more than 1.5 times greater, the analysis functions and derivatives are evaluated 6 times more. GEMSEO evaluates the objective function one more time than GEMSEO. The accuracy of the platforms is equal because the absolute error has the same order of magnitude for all the configurations.

Table 4: Results for full-analytic gradient-based optimization ¹

Architecture Platform	MDF		IDF		NVH	
	GEMSEO	OpenMDAO	GEMSEO	OpenMDAO	GEMSEO	OpenMDAO
Computation time [s]	0.3225	0.0305	0.0529	0.0237	0.0332	0.0172
Function evaluations	101	50	75	60 (24)	48	42
Derivative evaluations	98	45	70	55 (22)	42	36
Calls to objective function	7	6	15	12	8	7
Absolute error	8.99e-08	8.99e-08	8.99e-08	8.99e-08	8.99e-08	9.02e-08

The results shown in Tab.4 are obtained by providing analytical derivatives between the input and output variables of each discipline. Both GEMSEO and OpenMDAO allow defining this in the `_compute_jacobian` and `compute_partials` functions, respectively.

¹CoSApp does not enable the usage of analytical derivatives

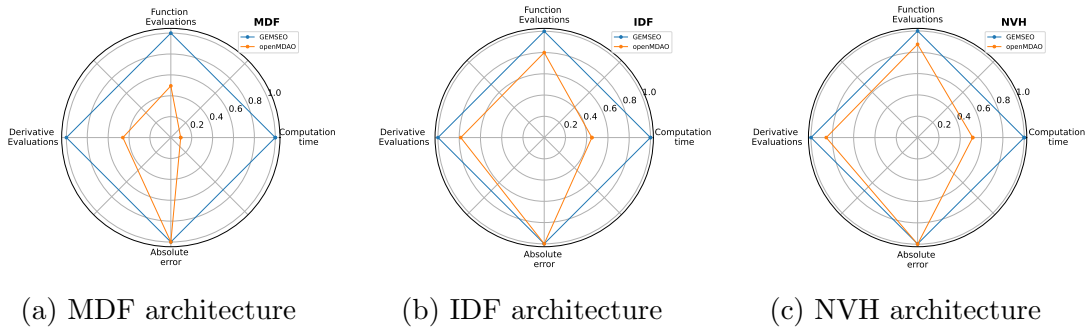


Figure 2: Results for analytical gradient-based optimization (data are normalized to the minimum value for each criterion)

CoSApp does not allow specifying analytical derivatives in the disciplines. For this reason, it is not found in the table. If partial derivatives cannot be obtained or explicit in analytical form, they can be approximated by finite differences or complex steps. In this work, the first method is used and the results are shown in Tab.5. The comparison between the Tab.5 and the Tab.4 illustrates that for all the architectures, both GEMSEO and OpenMDAO take longer to find the optimum and the number of function evaluations increases.

Concerning the MDF architecture, the order of magnitude of the computation time for GEMSEO and CoSApp is of the tenth of a second while for OpenMDAO is the hundredth of a second. As for the number of function evaluations, GEMSEO and CoSApp exceed OpenMDAO with 314 and 630 evaluations respectively. GEMSEO and CoSApp call the objective function 8 and 149 more times. In spite of this, the computing time required by CoSApp is less than that of GEMSEO. This paper cannot explain why as it would require more thorough investigations, but a clue may be found in the derivative calculations.

Regarding the IDF architecture, OpenMDAO's computation time is again on the order of hundreds of a second, while CoSApp and GEMSEO have a computation time on the order of tenths of a second. GEMSEO and CoSApp exceed the OpenMDAO function evaluations with 93 and 662 evaluations respectively. As for the calls to the objective function, GEMSEO and CoSApp require 17 and 131 more evaluations respectively.

In the case of the NVH architecture, all the computation times are in the order of hundreds of a second. Here, OpenMDAO shows the lowest number of function evaluations. GEMSEO and CoSApp exceed it with 58 and 150 evaluations respectively. OpenMDAO takes also the least number of objective function calls. GEMSEO and CoSApp are above it with 8 and 30 more calls respectively.

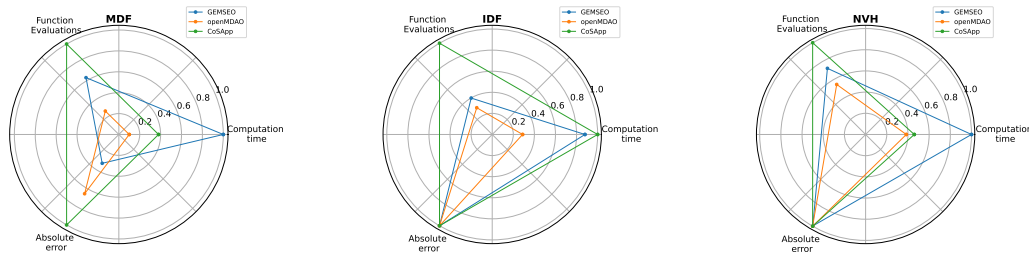
3.3 Gradient-free optimization

The results presented in Tab.6 and resumed graphically in Fig.4 have been obtained by means of the Constrained Optimization BY Linear Approximations algorithm (COBYLA), which constructs successive linear approximations of the objective function and constraints and optimizes these approximations in a trust region at each step.

The comparison between the Tab.5 and Tab.6 illustrates that for the MDF architecture a gradient-free algorithm performs better in terms of computation time.

Table 5: Results for gradient-based optimization with derivatives approximated by finite differences method

Architecture Platform	MDF			IDF			NVH		
	GEMSEO	OpenMDAO	CoSApp	GEMSEO	OpenMDAO	CoSApp	GEMSEO	OpenMDAO	CoSApp
Computation time [s]	0.4022	0.0401	0.1529	0.1082	0.0354	0.1230	0.0651	0.0250	0.0299
Function evaluations	534	220	850	351	258 (68)	880 (352)	238	180	330
Calls to objective function	29	21	170	62	45	176	33	25	55
Absolute error	4.37e-08	8.99e-08	1.37e-07	8.99e-08	8.99e-08	8.99e-08	8.99e-08	9.02e-08	9.02e-08



(a) MDF architecture

(b) IDF architecture

(c) NVH architecture

Figure 3: Results for gradient-based optimization with derivatives approximated by finite differences (data are normalized to the minimum value for each criterion)

Concerning the MDF architecture, the frameworks all show a computation time on the order of hundredths of a second except GEMSEO which is in the order of the tenth of a second. The minimum number of function evaluations of CoSApp is equal to 395. GEMSEO and OpenMDAO exceed it with 14 and 61 evaluations respectively. CoSApp calls the objective functions 63 more times than OpenMDAO and 62 more times than GEMSEO. The absolute errors are similar with the largest ratio between CoSApp and GEMSEO being equal to 1.53.

Regarding the IDF, OpenMDAO’s and CoSApp’s computation times are in the order of magnitude of hundredths of a second while GEMSEO is in the order of tenths of a second. In this case, the number of function evaluations is similar with a minimum of 390 represented by OpenMDAO and CoSApp. GEMSEO takes 143 more function evaluations. As for the objective function, OpenMDAO and CoSApp call it 78 times while GEMSEO calls it 29 more times. The absolute errors are all in the same order of magnitude.

As for the NVH architecture, all the frameworks have a computation time in the order of the hundredth of a second. The minimum of function evaluations is 168 obtained with OpenMDAO and CoSApp at the same time. GEMSEO exceeds it with 120 evaluations. The number of calls to the objective function is equal for OpenMDAO and CoSApp while GEMSEO exceeds it with 20 evaluations. The absolute errors are all in the same order of magnitude.

One should note that the computation time is not one of the most meaningful parameters, because it can include other platform performances not considered here, like the connections between variables or other security checks that are performed to prevent algorithms from diverging. For example, in GEMSEO, the user can activate or deactivate these

checks, and if all the checks are activated in the configuration API command, the computation time can more than double. Furthermore, it should be noted that in GEMSEO the COBYLA algorithm is wrapped from NLOPT library ([17]) whereas in OpenMDAO and CoSApp it is wrapped from SciPy library [18]. This is the reason why the results in terms of function evaluations and objective function calls are the same for OpenMDAO and CoSApp in the IDF and NVH architectures.

Table 6: Results for gradient-free optimization

Architecture Platform	MDF			IDF			NVH		
	GEMSEO	OpenMDAO	CoSApp	GEMSEO	OpenMDAO	CoSApp	GEMSEO	OpenMDAO	CoSApp
Computation time [s]	0.1064	0.0244	0.0621	0.1637	0.0302	0.0641	0.0827	0.0113	0.0191
Function evaluations	409	456	395	533	390 (156)	390 (156)	288	168	168
Calls to objective function	17	16	79	107	78	78	48	28	28
Absolute error	8.95e-08	9.54e-08	1.37e-07	8.99e-08	8.86e-08	8.75e-08	8.99e-08	9.23e-08	8.93e-08

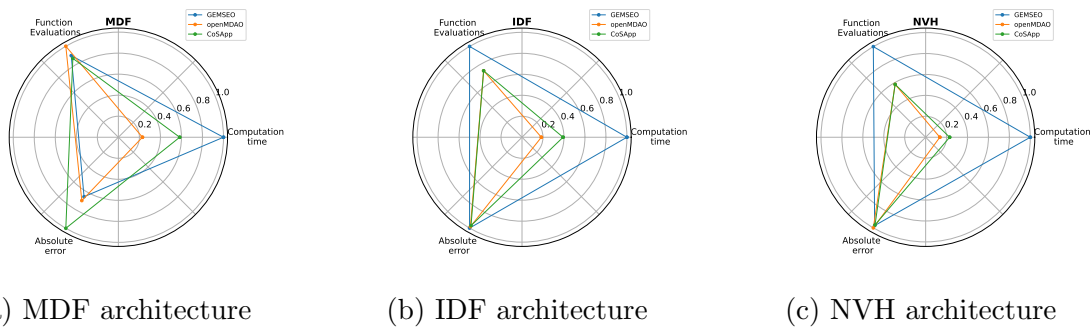


Figure 4: Results for gradient-free optimization (data are normalized to the minimum value for each criterion)

3.4 Optimizers and Solvers

In the approach to MDAO, the choice of the optimizer is an important criterion as it may strongly depend on the problem to solve. So it is necessary to evaluate the possibilities offered by each platform. In Tab.7, the classification collects the different optimizers available for several Python libraries. The first remark concerns the number of algorithms available in GEMSEO that exceeds the other two frameworks. Secondly, it should be noted that CoSApp does not allow the use of a genetic algorithm, which is instead available in various versions in GEMSEO and embedded versions in OpenMDAO.

In MDAO, analysis can be as important as optimization. Thus, it is helpful to compare the different framework solutions for numerical solvers. Numerical solvers can be divided into linear and non-linear types, and within the non-linear class there are some based on Newton’s method and others based on the fixed-point approach. This theoretical distinction is clearly explained in both the OpenMDAO and GEMSEO documentations. When using a Newton’s based non-linear solver, a linear solver is always associated and, unlike CoSApp, both OpenMDAO and GEMSEO allow the user to choose the linear solver. Table 8 shows an overview of linear and non-linear solvers available in the three frameworks. GEMSEO and OpenMDAO share a common base consisting of the Scipy linear solver methods, while for non-linear solvers only GEMSEO allows the reuse of Scipy

algorithms.

Table 7: Algorithms available. Algorithms marked with E are platform embedded

Python package	Scipy	Pyoptparse	Nlopt	pSeven	pymoo	PDFO	Other
OpenMDAO	Nelder-Mead						
	Powell	ALPSO					
	CG	CONMIN					
	BFGS	IPOPT					
	Newton-CG	NLPQLP					
	L-BFGS-B	NSGA-2	-	-	-	-	Differential evolution (E)
	TNC	ParOpt					Simple Genetic Algorithm (E)
	COBYLA	PSQP					
	SLSQP	SLSQP					
	Basinhoppin	SNOPT					
SHGO							
	Dual annealing						
GEMSEO	Differential evolution			PSEVEN			
	L-BFGS-B			PSEVEN_FD			
	Linear Interior Point		NLOPT_BFGS	PSEVEN_MOM	PYMOO_GA		
	Revised simplex		NLOPT_BOBYQA	PSEVEN_NCG	PYMOO_NSQA2	PDFO_BOBYQA	
	Simplex		NLOPT_COBYLA	PSEVEN_NLS	PYMOO_NSQA3	PDFO_COBYLA	SNOPT
			NLOPT_MMA	PSEVEN_POWELL	PYMOO_RNSQA3	PDFO_NEWUOA	MMA
	SLSQP		NLOPT_NEWUOA	PSEVEN_QP	PYMOO_UNSGA3		
	SNOPT		NLOPT_SLSQP	PSEVEN_SQ2P			
	TNC			PSEVEN_SQP			
	SHGO						
	Dual annealing						
CoSApp	Nelder-Mead						
	Powell						
	CG						
	BFGS						
	Newton-CG						
	L-BFGS-B						
	TNC						
	COBYLA						
	SLSQP						
	dogleg						
trust-constr							
trust-ncg							
trust-exact							
trust-krylov							

Table 8: Overview of linear and non-linear solvers

Platform	GEMSEO	OpenMDAO	CoSApp
Linear solvers	Direct method based on LU decomposition of a sparse matrix	Direct solver based on the linear equation solving algorithm of <code>scipy.linalg.solve</code>	
	BIConjugate Gradient method (BICG) ¹	Direct solver based on LU decomposition	
	BIConjugate Gradient STABilized method (BICGSTAB) ¹	Linear Block Gauss-Seidel	
	Generalized Minimal RESidual method (GMRES) ¹	Linear Block Jacobi	
	LGMRES method ¹	Krylov iterative methods	
Non linear solvers	Quasi-Minimal Residual Method (QMR) ¹	BICG method ¹	
		BICGSTAB method ¹	
		Conjugate Gradient method (CG) ¹	
		Conjugate Gradient Squared method (CGS) ¹	
		GMRES method ¹	
Non linear solvers	Newton-Raphson's method		
	Modified Powell's method ²		
	Levenberg-Marquardt's method ²		
	Broyden's methods ²		
	Anderson mixing method ²		
	Method based on scalar Jacobian approximation ²	Newton's method	Newton-Raphson
	Method based on diagonal Broyden Jacobian approximation ²	Broyden's method	Broyden's method
Method based on a tuned diagonal Jacobian approximation ²		Non linear Block Gauss Seidel	
Method based on Krylov approximatio for inverse Jacobian ²		Non linear Block Jacobi	Hybrid Powell's method
	DF-SANE method ²		
	Jacobi's method		
	GaussSeidel's method		

¹Wrapped from `scipy.sparse.linalg`

²Wrapped from `scipy.optimize.root`

3.5 Visualization tools

The XDSM schema can help engineers in the design phase see the variables exchanged between different disciplines and the couplings [16]. GEMSEO allows the user to generate this schema from a `xlsx` file from Excel in which only the output and input variables in each discipline, the type of architecture chosen, the design variables and the constraints must be entered. It is a simple and powerful implementation that can be used in real

time when the authors of individual subsystem models discuss the interfaces between their models. As a result, this enables action to be taken before the programming phase. For example, a discipline expert could set the model to change inputs and outputs to facilitate the solving of problems and avoid couplings that overcharge the optimization. In addition, GEMSEO allows the creation of an XDSM diagram automatically from the problem code. An extensive post-processing toolbox helps the user to analyze the problem results in GEMSEO. As for OpenMDAO and CoSApp, direct post-processing features are not available and the user is invited to store the problem data in a recorder in order to build his own post-processing tools.

Regarding the N2 matrix, all platforms allow its generation, although the user can benefit from differences in appearance and functionality as they are also dynamic. For example, this matrix in GEMSEO does not report the solvers used (Fig.5). Between OpenMDAO and CoSApp, the matrix is identical, although in CoSApp the connections between variables must be traced by the user and are not automatic (Fig.6, Fig.7).

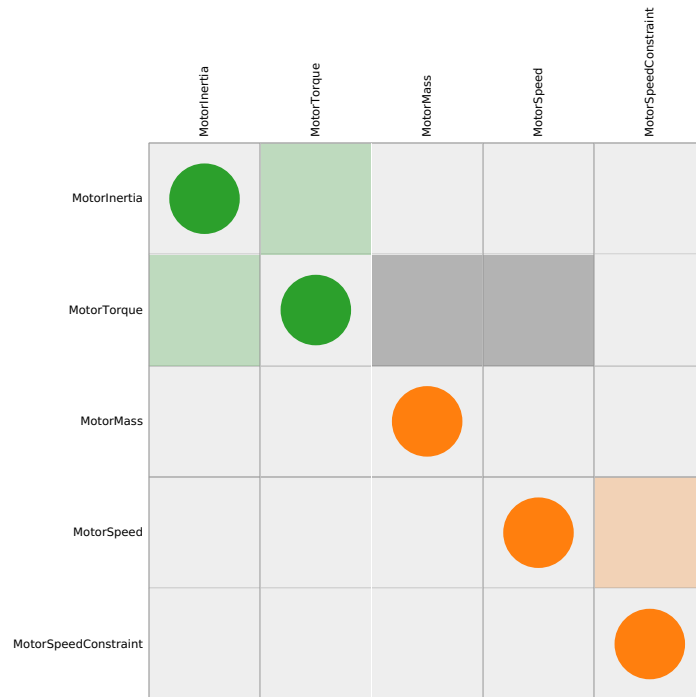


Figure 5: N2 matrix - GEMSEO

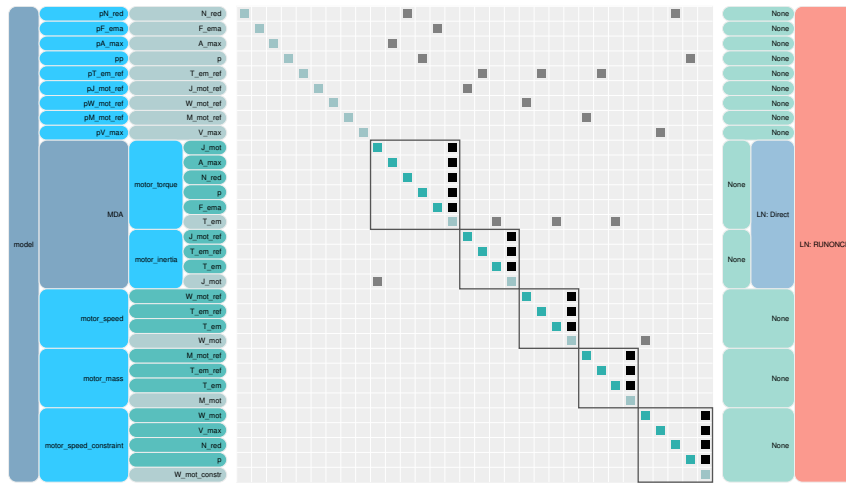


Figure 6: N2 matrix - openMDAO

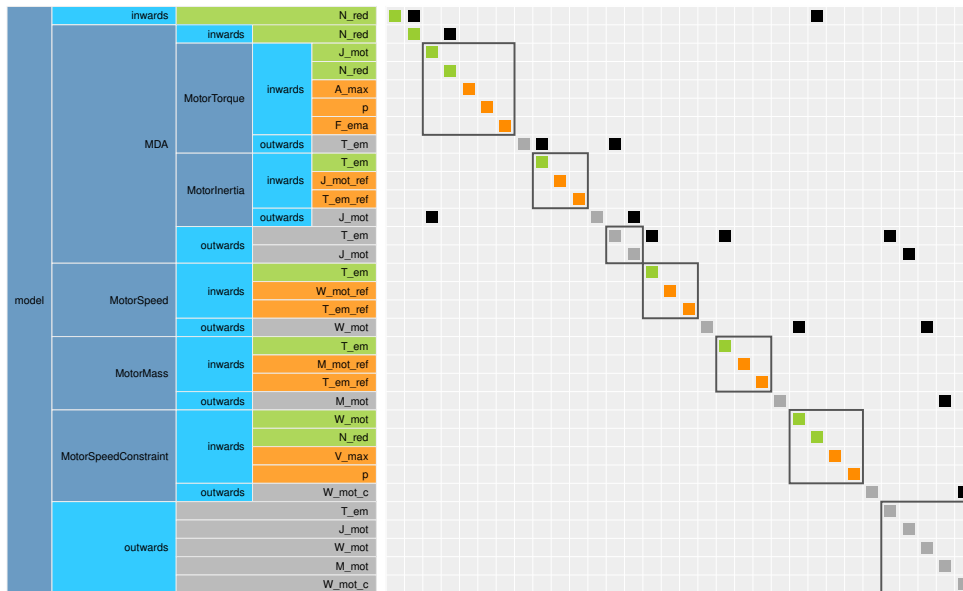


Figure 7: N2 matrix - CoSApp

4 CONCLUSION

The main objective of the study was to compare the three frameworks of an MDAO platform for use by a person with previous foundations in the field of multidisciplinary optimization but without any knowledge and training of each of the platforms investigated. The online documentation was the only source from which to draw all the information necessary to use the software. Despite the objectivity of the criteria selected to compare the platforms on the selected study case, it is necessary to consider a judgment bias related to the learning sequence of the different software. A user who learns to use one software before another may have a different opinion than another user who learns to use them in a different order, particularly regarding handling the main features. The documentation of OpenMDAO is rich in examples and the distinction between the theory for the beginner and advanced user helps to follow smoothly a learning path of the main features of increasing difficulty. On the other hand, GEMSEO presents a more modular

and open documentation, also rich in tutorials, which puts in the foreground the theoretical building bricks of MDAO which, in second place, allows the user to build his first optimization problem. A strength of GEMSEO is the wealth of documentation related to post-process tools and the sections dedicated to machine learning and the creation of surrogate models. For its part, CoSApp mainly documents its practical functionalities with many tutorials which make it a suitable tool for design, beyond the MDAO purpose. The optimization is therefore located in a single section and does not cover the core of the documentation, as is the case with the other two frameworks.

This study highlighted the need for rigor in the MDAO field. Although the concept of components for OpenMDAO, disciplines for GEMSEO and systems for CoSApp are based on the same theoretical definition, using different terms in codes intended to fulfil the same purpose denotes a lack of disciplinary rigor. Moreover, this comparative use of the three frameworks has also highlighted the lack of interoperability between the various tools available in open source software category. The GEMSEO roadmap mentions the intent to work on interoperability with OpenMDAO, while OpenMDAO does not mention any other software. An exchange between the various tools can benefit the entire MDAO community by allowing the users to adopt functions from others and bring valuable elements, as well as stimulating rigor in this field of research.

Given the large number of features these frameworks offer, and at the same time, the breadth of comparative criteria that could be used, an extension of this work could be the exploration of broader problems with more variables. Moreover, another study case could be the connection to external software of high fidelity which opens the way to another criterion such as the management of different levels of fidelity in the models employed. The switch from one model to another could be tested and the influence on the accuracy of the solution analyzed.

REPLICATION OF RESULTS

The results presented in this paper were generated by the source code available at the following Github repository: <https://github.com/RobertoDiGiuseppe/Benchmark>.

ANNEXES

```

1 class MotorTorque(System):
2
3     def setup(self):
4         self.add_inward('J_mot')
5         self.add_inward('N_red')
6         self.add_inward('A_max', fA_max)
7         self.add_inward('p', fp)
8         self.add_inward('F_ema', fF_ema)
9         self.add_outward('T_em')
10
11     def compute(self):
12         self.T_em = self.J_mot * self.A_max * self.N_red / self.p + self
            .F_ema * self.p / self.N_red

```

Listing 1: Discipline definition in CoSApp

```

1 class MotorTorque(MDODiscipline):

```

```

2
3     def __init__(self):
4         super().__init__(grammar_type = 'SimpleGrammar')
5         self.input_grammar.update(["J_mot", "A_max", "N_red", "p", "
F_ema"])
6         self.output_grammar.update(["T_em"])
7         self.default_inputs = {"J_mot": array([1]),
8                               "N_red": array([1]),
9                               "A_max": array([fA_max]),
10                              "p": array([fp]),
11                              "F_ema": array([fF_ema])
12        }
13     def _run(self):
14         J_mot, A_max, N_red, p, F_ema = self.get_inputs_by_name(["J_mot"
, "A_max", "N_red", "p", "F_ema"])
15         self.local_data["T_em"] = J_mot * A_max * N_red / p + F_ema * p
/ N_red

```

Listing 2: Discipline definition in GEMSEO

```

1     class MotorTorque(ExplicitComponent):
2
3         def __init__(self, derivative_method='full_analytic', **kwargs):
4             super(MotorTorque, self).__init__(**kwargs)
5
6         def setup(self):
7             self.add_input('J_mot', val=1.)
8             self.add_input('A_max', val=1.)
9             self.add_input('N_red', val=1.)
10            self.add_input('p', val=1.)
11            self.add_input('F_ema', val=1.)
12            self.add_output('T_em', val=1.)
13
14        def compute(self, inputs, outputs):
15            J_mot = inputs['J_mot']
16            A_max = inputs['A_max']
17            N_red = inputs['N_red']
18            p = inputs['p']
19            F_ema = inputs['F_ema']
20            outputs['T_em'] = J_mot * A_max * N_red / p + F_ema * p / N_red

```

Listing 3: Discipline definition in OpenMDAO

REFERENCES

- [1] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013. URL <https://doi.org/10.2514/1.J051895>.
- [2] S. Delbecq, M. Budinger, and A. Reysset. Benchmarking of monolithic mdo formulations and derivative computation techniques using openmdao. *Structural and Multidisciplinary Optimization*, 2020. <https://doi.org/10.1007/s00158-020-02521-7>.
- [3] B. Malone and M. Papay. Modelcenter: an integration environment for simulation based design. In *Simulation Interoperability Workshop*, 1999.

- [4] ESTECO. modeFRONTIER, 1998. URL <https://engineering.esteco.com/modefrontier/>.
- [5] N. Solutions. Optimus. URL <https://www.noesissolutions.com/our-products/optimus>.
- [6] Siemens. Heeds. URL <https://plm.sw.siemens.com/en-US/simcenter/integration-solutions/heeds/>.
- [7] Altair. Hyperstudy. URL <https://altair.com/hyperstudy/>.
- [8] Ansys. optiSLang. URL <https://www.ansys.com/products/connect/ansys-optislang>.
- [9] A. Van der Velden, B. Wujek, and P. Koch. isight-fd: A tool for multi-objective data analysis. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5988, 2008.
- [10] DATADVANCE. psevent. URL <https://www.3ds.com/fr/produits-et-services/simulia/produits/isight-et-simulia-execution-engine/>.
- [11] V. RD. visualdoc. URL <http://www.vrand.com/products/visualdoc/>.
- [12] D. Seider, P. M. Fischer, M. Litz, A. Schreiber, and A. Gerndt. Open source software framework for applications in aeronautics and space. In *2012 IEEE Aerospace Conference*, page 1–11, Big Sky, MT, Mar 2012. IEEE. ISBN 978-1-4577-0557-1. doi:10.1109/AERO.2012.6187340. URL <http://ieeexplore.ieee.org/document/6187340/>.
- [13] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi:10.1007/s00158-019-02211-z.
- [14] F. Gallard, P.-J. Barjhoux, R. Olivanti, and A. Gazaix. *GEMS, a Generic Engine for MDO Scenarios : Key Features in Application*. doi:10.2514/6.2019-2991. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2019-2991>.
- [15] S. SA. Cosapp - collaborative system approach. <https://cosapp.readthedocs.io/en/latest/>.
- [16] A. B. Lambe and J. R. R. A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, Aug 2012. ISSN 1615-147X, 1615-1488. doi:10.1007/s00158-012-0763-y.
- [17] S. G. Johnson. The nlopt nonlinear-optimization package. <http://github.com/stevengj/nlopt>.

- [18] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272, 2020. doi:10.1038/s41592-019-0686-2.



THERMO-MECHANICAL LEVEL-SET TOPOLOGY OPTIMIZATION OF A LOAD CARRYING BATTERY PACK FOR ELECTRIC AIRCRAFT

Alexandre T.R. Guibert¹, Murtaza Bookwala¹, Ashley Cronk², Y. Shirley Meng³, and H. Alicia Kim^{1,2}

1: Structural Engineering Department
Jacobs School of Engineering
University of California San Diego
Matthews Lane, La Jolla, CA 92093, USA
{aguibert,mbookwala,alicia}@ucsd.edu, <http://m2do.ucsd.edu/>

2: Materials Science and Engineering Department
Jacobs School of Engineering
University of California San Diego
Matthews Lane, La Jolla, CA 92093, USA
acronk@eng.ucsd.edu, <http://smeng.ucsd.edu/>

3: Pritzker School of Molecular Engineering
The University of Chicago
5801 S Ellis Avenue, Chicago, 60637, Illinois, USA
shirleymeng@uchicago.edu, <https://voices.uchicago.edu/smeng/>

Abstract. *A persistent challenge with the development of electric vertical take-off and landing vehicles (eVTOL) to meet flight power and energy demands is the mass of the load and thermal management systems for batteries. One possible strategy to overcome this problem is to employ optimization techniques to obtain a lightweight battery pack while satisfying structural and thermal requirements. In this work, a structural battery pack with high-energy-density cylindrical cells is optimized using the level-set topology optimization method. The heat generated by the batteries is predicted using a high-fidelity electrochemical model for a given eVTOL flight profile. The worst-case scenario for the battery's heat generation is then considered as a source term in the weakly coupled steady-state thermo-mechanical finite element model used for optimization. The objective of the optimization problem is to minimize the weighted sum of thermal compliance and structural compliance subjected to a volume constraint. The methodology is demonstrated with numerical examples for different sets of weights. The optimized results due to different weights are compared, discussed, and evaluated with thermal and structural performance indicators. The optimized pack topologies are subjected to a transient thermal finite element analysis to assess the battery pack's thermal response.*

Keywords: Multiphysics topology optimization, level-set method, structural battery pack, electrochemical model, electric vehicles

1 INTRODUCTION

The aerospace sector is undergoing a significant transformation due to the emergence of electric aircraft and the advancement of urban air mobility (UAM) concepts. One of the solutions to UAM is the electric Vertical Take-Off and Landing (eVTOL) vehicles, which have the potential to alleviate ground traffic congestion and provide a secure and environmentally friendly alternative to conventional individual transportation. However, eVTOL vehicles pose fundamental engineering challenges such as noise control and power requirements. To make eVTOL vehicles a viable transportation option, there is a crucial need to enhance their power density while ensuring safety by carefully designing the vehicles' battery packs. By utilizing topology optimization, it is possible to optimize the distribution of materials in a battery pack with the objective of improving the structural and thermal performance while minimizing its weight which ultimately contributes to the overall efficiency of the aircraft. This can eliminate a mechanical and thermal load management system which is otherwise needed, thus reducing the mass and volume budget. Therefore, the integration of topology optimization in the development of eVTOL vehicles presents a promising opportunity to move closer to a sustainable and efficient form of UAM.

In this work, we present a methodology for the multiphysics optimization of a battery pack for electric aircraft using the level set method. The volumetric heat generation rate from the batteries is estimated using an electrochemical model and the worst-case scenario is considered for design optimization. The electrochemical model, heat transfer model, and elasticity model are introduced in Section 2.1, 2.2, and 2.3 respectively. The level-set topology optimization method and the optimization problem and workflow are discussed in Section 3. The methodology is demonstrated with numerical examples in Section 4.

2 MULTIPHYSICS MODEL

2.1 High fidelity electrochemical model

The cell behavior is modeled using the Doyle Fuller Newman (DFN) model [1–4] which describes the electrochemical response of lithium-ion cells. It is solved for the electric potentials and lithium-ion concentration distribution in a cell given some electric loads such as an applied current, voltage, or power. The DFN model in its one-dimensional version assumes that the charge/discharge process is mainly unidirectional, i.e., only the effects from one current collector to another are considered. It also assumes that the solid particles are spherical. The lithium-ion diffusion in these particles is considered by having a pseudo second dimension associated with the particles' radius. The governing equations for the DFN model are the following:

1. Molar conservation in the electrolyte,
2. Solid state Fickian diffusion in the the solid particles,
3. Charge conservation in the electrolyte and in the particles based on Ohm's law,
4. Intercalation kinetics based on the Butler-Volmer equation which effectively couples the macro dimension and the pseudo dimension.

The DFN model is implemented using the open-source library PyBaMM [5–7].

2.2 Heat transfer model

The governing equation for the steady-state thermal model considering thermal conduction only is,

$$\nabla \cdot (\kappa \nabla T) + Q = 0, \quad (1)$$

where κ is the thermal conductivity, T is the temperature, and Q is the volumetric heat generation rate. The latter is determined by the electrochemical model by adding the volumetric heat generation due to Ohmic heating Q_{Ohm} from the electrical resistance of the cell, irreversible heating from the electrochemical reactions Q_{rxn} , and reversible heating due to entropic changes Q_{rev} . The heat transfer model is implemented using the finite element method and automatic differentiation software, FEniCS and DOLFIN [8–10].

2.3 Elasticity model

The governing equation of the linear elasticity problem is,

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}) + \mathbf{b} = 0, \quad (2)$$

where \mathbf{b} is the body force, \mathbf{u} is the displacement, and $\boldsymbol{\sigma}$ is the Cauchy stress tensor. The temperature distribution solved by the heat transfer model resulting in thermal strain ε_T ,

$$\varepsilon_T = \alpha \Delta T, \quad (3)$$

where α is the coefficient of thermal expansion and ΔT is the temperature increase, that is, $\Delta T = T - T_{ref}$ where T_{ref} is the reference temperature. The constitutive law then becomes,

$$\boldsymbol{\sigma} = \mathbb{C} : (\boldsymbol{\varepsilon} - \varepsilon_T \mathbf{1}), \quad (4)$$

where \mathbb{C} is the elastic constitutive tensor, $\mathbf{1}$ is the unit tensor, and $\boldsymbol{\varepsilon}$ is the strain tensor defined as $\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^\top)$. The linear elasticity model is implemented in its weak form using FEniCS and DOLFIN [8–10].

3 TOPOLOGY OPTIMIZATION

3.1 Level-set topology optimization

The optimized design is described by an implicit function $\phi(\mathbf{x})$ such that

$$\begin{cases} \phi(\mathbf{x}) \geq 0, \mathbf{x} \in \Omega \\ \phi(\mathbf{x}) = 0, \mathbf{x} \in \Gamma \\ \phi(\mathbf{x}) < 0, \mathbf{x} \notin \Omega \end{cases} \quad (5)$$

where Ω is the solid domain such that $\Omega \subset \mathcal{D}$ where \mathcal{D} is the design domain and $\mathcal{D} \setminus \Omega$ is the void domain. The boundary of the solid domain is Γ described by the zero level-set. The design is iteratively updated by the discretized Hamilton-Jacobi equation,

$$\phi_i^{k+1} = \phi_i^k - \Delta t |\nabla \phi_i^k| V_{n,i}, \quad (6)$$

where k is the iteration number, i is a boundary point, Δt is the pseudo time-step, and V_n is the design velocity normal to Γ [11, 12]. The finite element analysis is carried out

on a fixed grid, i.e., a non-conforming mesh is used. Thus, as the design evolves, the zero level-set cuts finite elements such that for a given element some nodes of that element might be inside the optimized design and some might be outside. To address this issue, the following material interpolation scheme is used,

$$\begin{aligned}\kappa_e &= (\gamma_{min}(1 - \gamma_e) + \gamma_e) \kappa, \\ E_e &= (\gamma_{min}(1 - \gamma_e) + \gamma_e) E,\end{aligned}\tag{7}$$

where γ_e is the fraction of the element where $\phi(\mathbf{x}) > 0$, and γ_{min} is a small value, taken as 10^{-4} for this study, to ensure that the stiffness and conductivity matrices are non-singular. The optimization is implemented using ParaLeSTO [13] which is a modular open-source level-set topology optimization code [14, 15].

3.2 Optimization workflow and formulation

The objective function of the optimization problem \mathcal{J} is the weighted sum of the normalized thermal and structural compliance, i.e.,

$$\mathcal{J}(\gamma, T(\gamma), \mathbf{u}(\gamma)) \triangleq k_1 \frac{C_S}{C_{S0}} + k_2 \frac{C_T}{C_{T0}},\tag{8}$$

where C_S and C_T are the structural compliance and thermal compliance defined as

$$\begin{aligned}C_S &= \mathbf{F}_S^\top \mathbf{U}, \\ C_T &= \mathbf{F}_T^\top \mathbf{T},\end{aligned}\tag{9}$$

where \mathbf{F}_T and \mathbf{F}_S are the thermal force vector and structural force vector respectively. The normalization factors C_{S0} and C_{T0} are reinitialized every 5 iterations. Moreover, let $k \equiv k_1 = 1 - k_2$. Thus, Eq.(8) becomes

$$\mathcal{J}(\gamma, T(\gamma), \mathbf{u}(\gamma)) = k \frac{C_S}{C_{S0}} + (1 - k) \frac{C_T}{C_{T0}}.\tag{10}$$

Additionally, the volume for the optimized part is constrained such that a fraction ξ of the design domain is prescribed. Finally, the optimization problem to be solved can be expressed as

$$\begin{aligned}\text{Minimize} & \quad \mathcal{J}(\gamma, T(\gamma), \mathbf{u}(\gamma)) \\ \text{w.r.t.} & \quad \gamma \\ \text{subject to} & \quad \text{Vol}(\Omega) \leq \xi \text{Vol}(\mathcal{D}) \\ & \quad \mathcal{R}_1(\gamma, T(\gamma)) = 0 \\ & \quad \mathcal{R}_2(\gamma, \mathbf{u}(\gamma)) = 0\end{aligned}\tag{11}$$

where \mathcal{R}_1 and \mathcal{R}_2 are the residuals of the thermal and elasticity models respectively. The optimization workflow is presented in Fig.1.

4 NUMERICAL RESULTS

4.1 Set up and boundary conditions

The LG CHEM INR21700-M50L cell is used for its high energy density and the properties are taken from [16]. The power profile is computed using [17] where the energy is

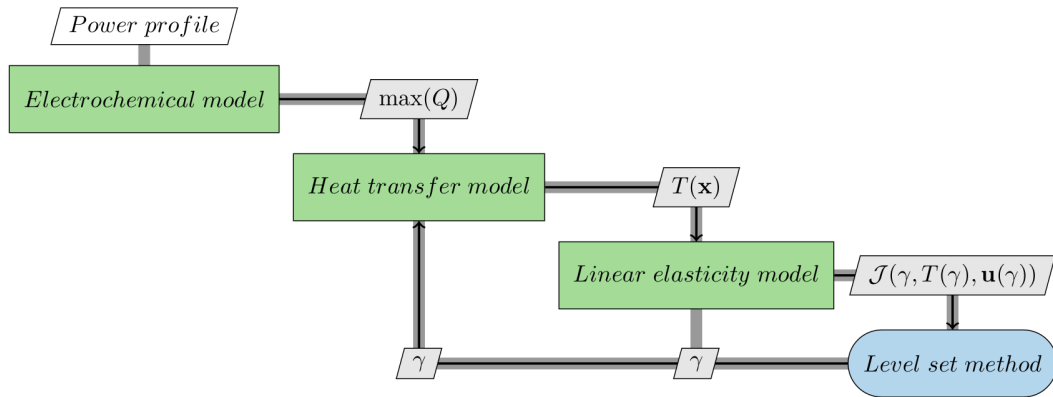


Figure 1: Optimization workflow

taken as 17.8 Wh [18]. The power profile, the voltage profile obtained with the electrochemical model, and the estimated volumetric heat generation rate are presented in Fig.2.

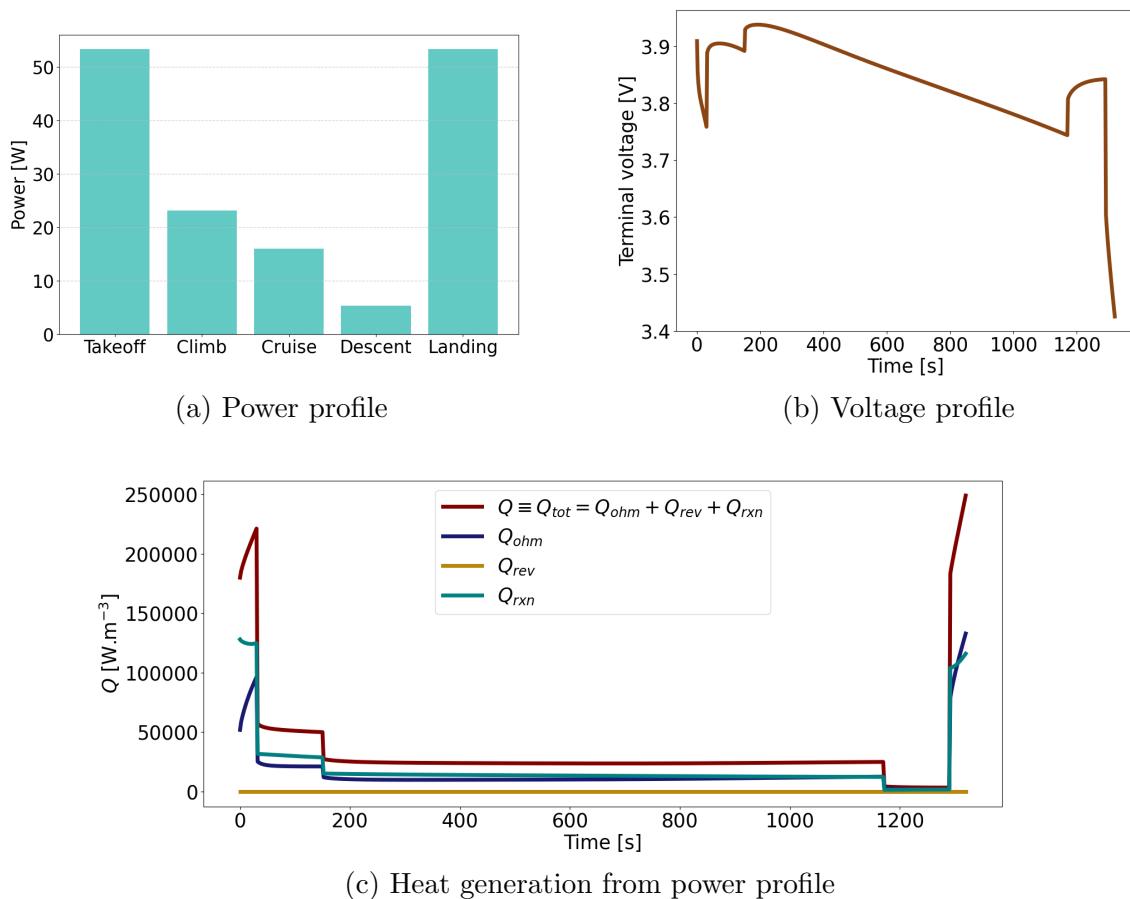


Figure 2: Electrochemical response of 21700 cells for a given power profile

A 3x2 configuration with a spacing of 5 mm between the cells is chosen for the battery pack. The cells are 70 mm in height with a diameter of 21 mm. The pack is cooled at the top and bottom assuming a constant fixed temperature. Structurally the pack is analogous to a cantilever beam with one side fixed and the other subjected to a traction load of 15 N/mm². The boundary conditions and the geometry are presented in Fig.3.

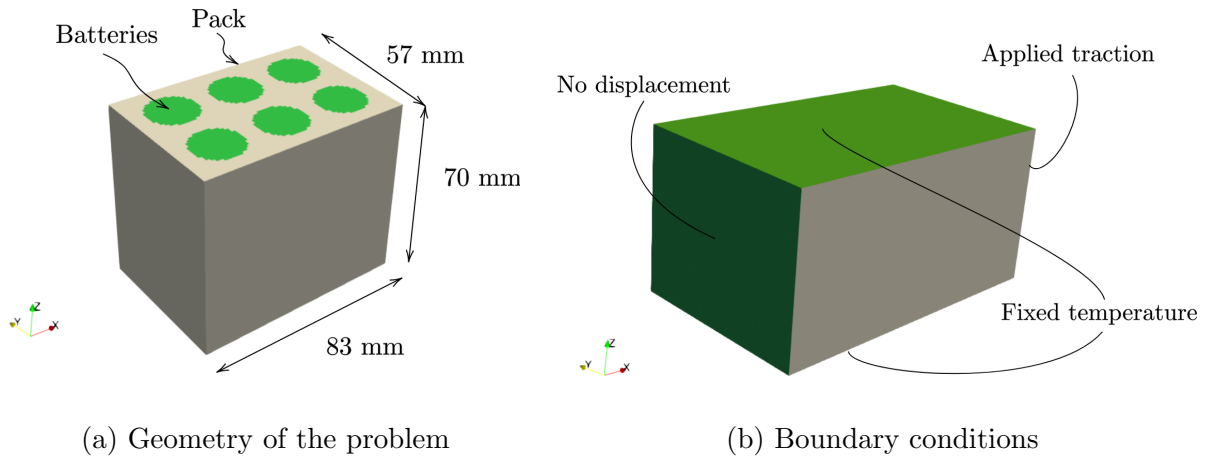


Figure 3: Geometry and boundary conditions for the numerical examples where only Dirichlet boundary conditions and non-zero Neumann boundary conditions are given

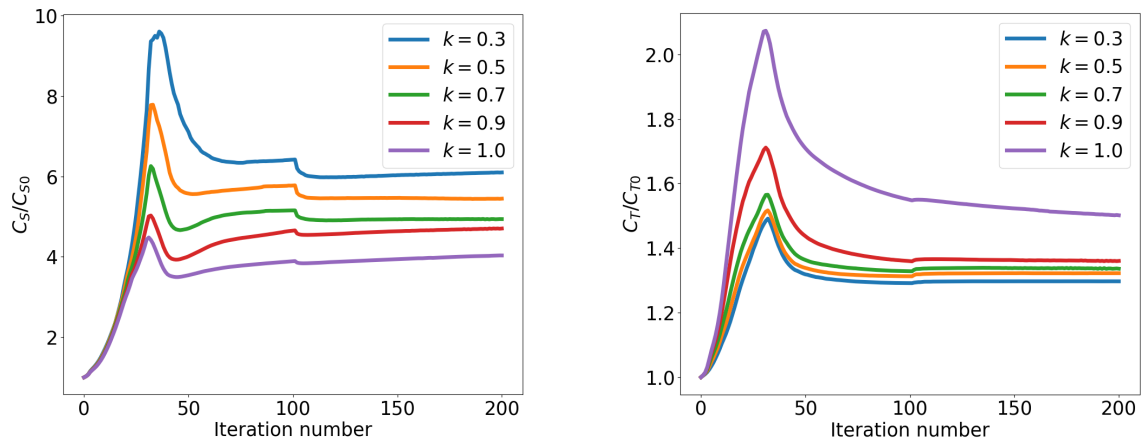
The battery pack is discretized with $48 \times 32 \times 40$ linear hexahedra elements leading to a total of 265,188 degrees of freedom. The material properties for the study are given in Table 1.

Table 1: Properties for the thermo-mechanical analysis

Property	Aluminum	Cell
Thermal conductivity κ [W/m.K]	220.0	1.25
Young's modulus E [GPa]	68.0	1.5
Poisson's ratio ν [-]	0.32	0.2
Thermal expansion α [K^{-1}]	21×10^{-6}	10×10^{-6}

4.2 Optimization results

The following weights are investigated $k = 1.0, 0.9, 0.7, 0.5, 0.3$. The convergence history for the thermal and structural compliance for the 5 cases is given in Fig.4. First, the volume constraint is met, and then the objective is gradually minimized until it reaches the minimum bound. The jump at iteration 100 is caused by a modification in the move limit value, which limits the amount of change allowed from one iteration to the next. The optimization results for 3 selected cases are presented in Fig.5. As the value of k is decreased, we observe a shift in the topology between the cells - from a stiffener-like structure to one consisting of elements that resemble beams. Additionally, more material is allocated towards the ends of the batteries to facilitate their connection to the heat sinks. Finally, the maximum displacement and maximum temperature of the final design as a function of the weights given to each compliance are given in Fig.6. Once thermal compliance is taken into account, a significant reduction in the normalized maximum temperature is observed. However, as the value of k continues to decrease, the decrease in maximum temperature becomes less pronounced. This trend is similar to what is observed in the convergence plot shown in Fig.4.



(a) Structural compliance convergence history (b) Thermal compliance convergence history

Figure 4: Convergence history of structural and thermal compliances for different values of k . Note that the jump at iteration 100 is due to a reduction of the move limit value

4.3 Transient response of the battery pack

The effective volumetric heat capacity ρc_p is taken as $1,767,574 \text{ J}\cdot\text{m}^{-3}\cdot\text{K}^{-1}$ and $2,430,000 \text{ J}\cdot\text{m}^{-3}\cdot\text{K}^{-1}$ for the cell and the pack, respectively [16, 19, 20]. The optimized topologies are analyzed using a transient heat conduction model implemented using FEniCS. An implicit Euler scheme that is unconditionally stable is chosen for the time discretization. The temperature time history and temperature distribution are presented in Fig.7. The maximum temperatures during take-off and landing do not appear to be significantly affected by thermal compliance considerations. These phases are short and the steady state is not reached. The overall maximum temperature does not seem to be much reduced when considering the topology optimized for steady-state considerations. Consequently, it seems worthwhile to consider transient heat conduction topology optimization specifically for takeoff and landing or for the entire flight but that is beyond the scope of this study. On the other hand, in the cruise segment where steady-state is reached, considering thermal compliance in the optimization formulation leads to a reduction of more than 10 % of the maximum temperature. Such a reduction could mitigate battery degradation. More research is needed to quantify this effect.

5 CONCLUSIONS

In this study, we presented a methodology for conducting thermo-mechanical level-set topology optimization of a load-carrying battery pack in an eVTOL and employed an electrochemical model to predict heat generation from the batteries based on a given flight profile. A multi-objective function that considered both thermal and structural compliance was minimized. We demonstrated the approach for several cases and analyzed them using a transient heat conduction model. The methodology can be adapted and reused for different power profiles, boundary conditions, and pack configurations. In the future, we plan to expand our research to encompass various power profiles and boundary conditions, examine degradation and different battery chemistries, and increase the scale of the thermo-mechanical model.

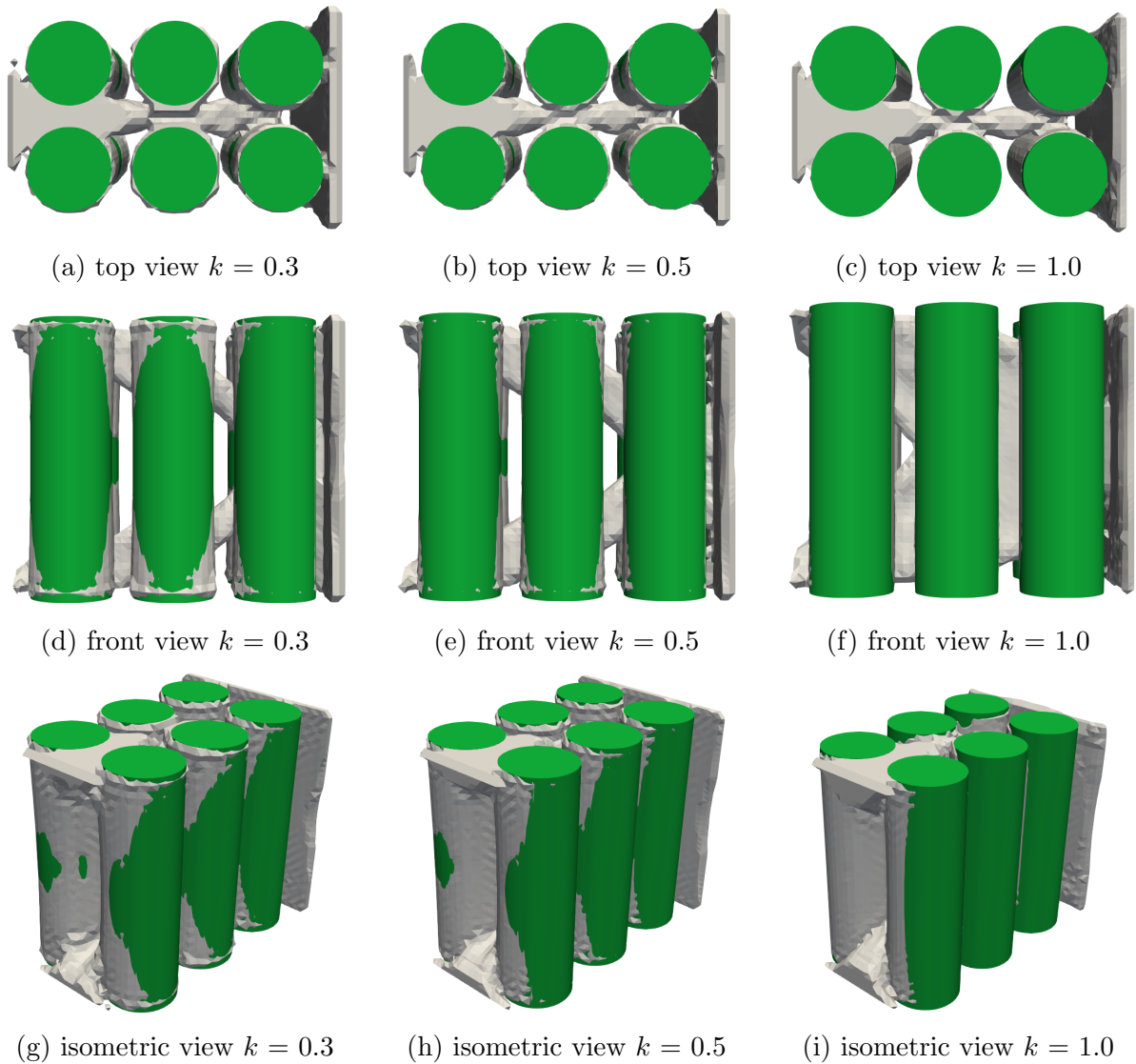


Figure 5: Optimization results for selected cases

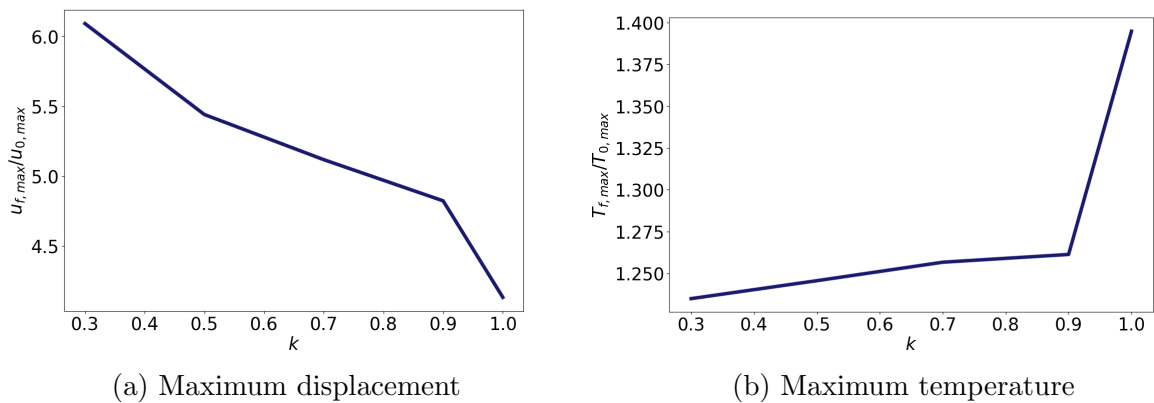


Figure 6: Maximum displacement and temperature of the final design normalized by the maximum state of the initial design as a function of k

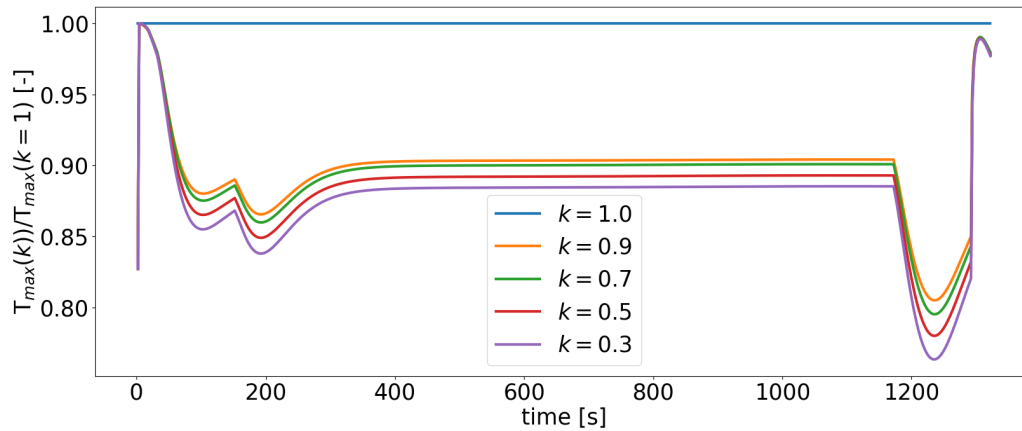


Figure 7: Maximum temperature time history normalized by the maximum temperature time history for $k = 1$

ACKNOWLEDGEMENTS

The authors acknowledge the support from NASA under award No. 80NSSC21M0070.

REFERENCES

- [1] M. Doyle, T. F. Fuller, and J. Newman. Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell. *Journal of the Electrochemical society*, 140(6):1526–1533, 1993. doi:10.1149/1.2221597.
- [2] W. Ai, L. Kraft, J. Sturm, A. Jossen, and B. Wu. Electrochemical thermal-mechanical modelling of stress inhomogeneity in lithium-ion pouch cells. *Journal of The Electrochemical Society*, 167(1):013512, 2019. doi:10.1149/2.0122001JES.
- [3] R. Timms, S. G. Marquis, V. Sulzer, C. P. Please, and S. J. Chapman. Asymptotic Reduction of a Lithium-ion Pouch Cell Model. *SIAM Journal on Applied Mathematics*, 81(3):765–788, 2021. doi:10.1137/20M1336898.
- [4] R. Deshpande, M. Verbrugge, Y.-T. Cheng, J. Wang, and P. Liu. Battery cycle life prediction with coupled chemical degradation and fatigue mechanics. *Journal of the Electrochemical Society*, 159(10):A1730, 2012. doi:10.1149/2.049210jes.
- [5] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson, and S. J. Chapman. Python Battery Mathematical Modelling (PyBaMM). *Journal of Open Research Software*, 9(1):14, 2021. doi:10.5334/jors.309.
- [6] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019. doi:10.1007/s12532-018-0139-4.
- [7] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi:10.1038/s41586-020-2649-2.

- [8] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [9] A. Logg and G. N. Wells. Dofin: Automated finite element computing. *ACM Transactions on Mathematical Software (TOMS)*, 37(2):1–28, 2010.
- [10] A. Logg, K.-A. Mardal, and G. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.
- [11] P. D. Dunning and H. A. Kim. Introducing the sequential linear programming level-set method for topology optimization. *Structural and Multidisciplinary Optimization*, 51:631–643, 2015.
- [12] R. Sivapuram, P. D. Dunning, and H. A. Kim. Simultaneous material and structural optimization by multiscale topology optimization. *Structural and multidisciplinary optimization*, 54:1267–1281, 2016.
- [13] S. Kambampati and C. M. Jauregui. Paralesto, Feb. 2023. URL <https://doi.org/10.5281/zenodo.7613753>.
- [14] J. Hyun, C. Jauregui, H. A. Kim, and A. Neofytou. On development of an accessible and non-intrusive level-set topology optimization framework via the discrete adjoint method. In *AIAA SCITECH 2022 Forum*, page 2548, 2022.
- [15] A. T. Guibert, J. Hyun, A. Neofytou, and H. A. Kim. Implementation of a plug-and-play reusable level-set topology optimization framework via comsol multiphysics. In *AIAA SCITECH 2023 Forum*, page 1675, 2023.
- [16] C.-H. Chen, F. Brosa Planella, K. O’Regan, D. Gastol, W. D. Widanage, and E. Kendrick. Development of Experimental Techniques for Parameterization of Multi-scale Lithium-ion Battery Models. *Journal of The Electrochemical Society*, 167(8): 080534, 2020. doi:10.1149/1945-7111/ab9050.
- [17] X.-G. Yang, T. Liu, S. Ge, E. Rountree, and C.-Y. Wang. Challenges and key requirements of batteries for electric vertical takeoff and landing aircraft. *Joule*, 5(7):1644–1659, 2021.
- [18] Batemo. Lg chem inr21700-m50l, Oct 2022. URL <https://www.batemo.de/products/batemo-cell-library/lg-chem-inr21700-m50l/>.
- [19] K. O’Regan, F. B. Planella, W. D. Widanage, and E. Kendrick. Thermal-electrochemical parameters of a high energy lithium-ion cylindrical battery. *Electrochimica Acta*, 425:140700, 2022.
- [20] M. Steinhardt, J. V. Barreras, H. Ruan, B. Wu, G. J. Offer, and A. Jossen. Meta-analysis of experimental results for heat capacity and thermal conductivity in lithium-ion batteries: A critical review. *Journal of Power Sources*, 522:230829, 2022.



TOPOLOGY OPTIMIZATION OF A SOLID GRAIN HYBRID ROCKET LAUNCHER

Mirko Melis^{1*}, Alain Souza¹ and Frederico Afonso¹

1: IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais, 1049-001
{mirko.melis, alain.souza, frederico.afonso}@tecnico.ulisboa.pt,
<https://www.idmec.tecnico.ulisboa.pt>

Abstract. *To reduce the environmental impact of conventional rocket fuels a hybrid fuel combining a liquid oxidizer and a solid grain is being considered. However, this solution increases the complexity of the combustion process when compared to solid or liquid options. Thus, to obtain a fuel that simultaneously minimizes the environmental impact while providing the highest performance possible a Multidisciplinary Design Optimization (MDO) problem should be stated. We propose to develop a MDO framework that couples topology optimization to design the solid grain geometry considering manufacturing constraints with a propulsion model to predict the specific impulse generated. In the end, a multi-objective function comprising structural and performance goals is established.*

Keywords: Topology optimization, hybrid propulsion, multidisciplinary design optimization, rocket launcher

1 INTRODUCTION

Hybrid Rocket Engines (HREs) are rocket propulsion systems with fuels and oxidizers in different phases of matter, mainly a solid fuel that has a liquid or gas oxidizer. Traditional HREs have a liquid oxidizer that flow across a solid fuel grain [1].

Figure 1 presents a schematization of a general HRE.

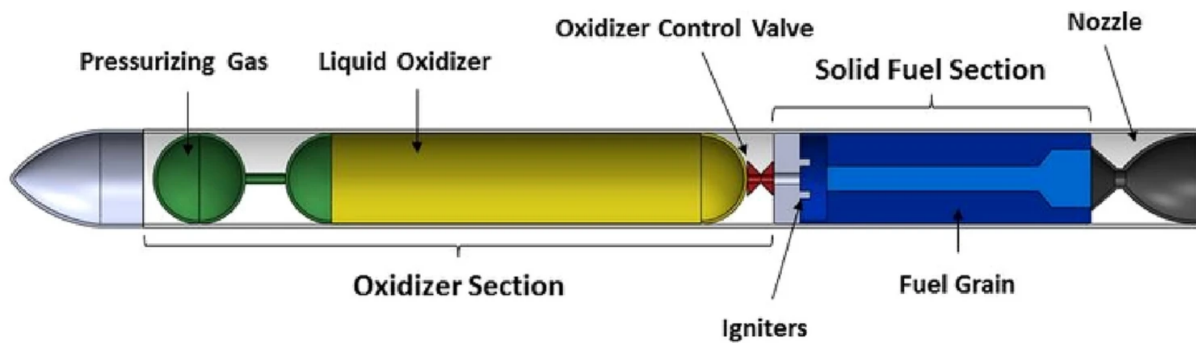


Figure 1: Hybrid Rocket Engine [2]

The idea of a HREs is to have the two propellants in different phases and to store the oxidizer in a tank separate from the inert fuel. This renders them less susceptible to chemical explosion than conventional LRE or SRE designs. The heat of the flame and blowing conditions cause the solid fuel to change its phase to vapor where it is then released from the fuel wall and contributes to further reactions. Fuel vaporizing or pyrolyzing at the face of the solid fuel grain causes the surface to regress. HREs are an attractive choice as an alternative to LREs and SREs, because they combine ideas from both LREs and HREs and try to benefit from the advantages of these more established engines [3].

The performance of HREs is frequently challenged by the low fuel mass addition, which depends on the fuel geometry, and the flow rate of oxidizer. Often HREs have varying performance conditions that shift over the run duration. The primary changes that affect performance are the oxidizer and fuel ratio [4].

The main and unique proprieties of HREs are summarized below [4]:

- Higher Safety and Manufacturing conditions;
- Performance Parameters coupled with Geometry;
- Less Mechanical Complexity than LREs;
- Less Inert Mass than LREs;
- Less Toxic Exhaust Products than SREs;
- Simpler Control Mechanism like LREs.

A comparison of the properties with LREs and SREs is reported in Table 1, where it is clearly possible to see how the HREs represent a combination of them.

HREs bring several safety and environmental advantages over the other two systems, but it has never been commercially widespread. The main limiting factors of HREs are the low regression rate, due to the nature of the combustion process, and the low combustion efficiency. The main factors influencing the regression rate are propellant composition, geometry of the grains, injector configuration and scale. Regarding the

geometry issues, in the last years, new research trends have been developed that promote additive manufacturing and rapid prototyping as possible solutions and a beginning point for further studies [4].

Table 1: Comparison between LRE, SRE and HRE [5]

	Liquid	Solid	Hybrid
Performance (I_{sp})	250-450 sec	200-280 sec	230-300 sec
Cost	High	Low	Moderate
Complexity	High	Low	Moderate
Throttle-ability	Yes	No	Yes
Restart	Yes	No	Yes
Explosive Hazard	Moderate	High	None
Handling & Transport Risk	Moderate	High	Very Low
Storability	Low	High	Moderate to High
Operational Risk	High	High	Very Low
Environmental Impact	Low to High	Moderate to High	Very Low

1.1 Internal ballistic

To explain the basic function of the HRE internal ballistic, it is necessary to begin with the geometric definition of the following parameters, considered for a cylindrical fuel grain illustrated in Fig.2.

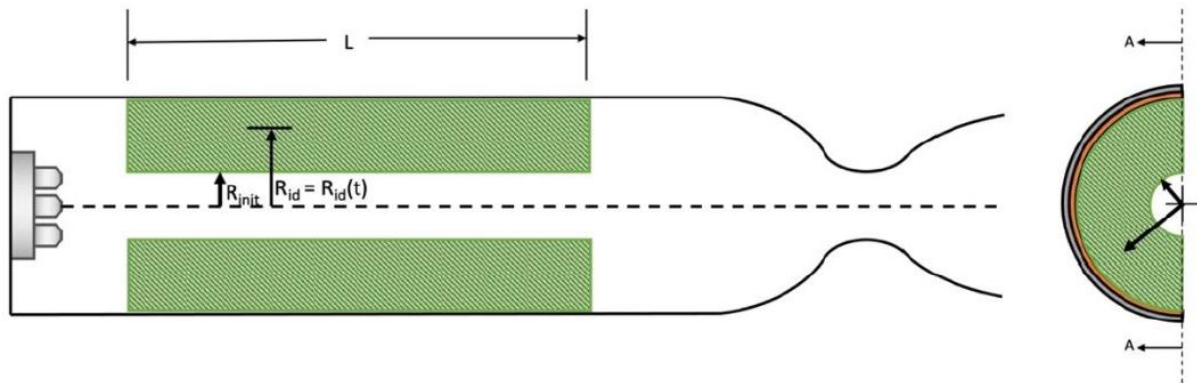


Figure 2: Schematic geometric representation of an HRE [4]

$$A_p = \pi R_{id}^2 \quad (1)$$

$$\dot{M} = \dot{M}_{ox} + A_p \dot{r} \rho_f \quad (2)$$

$$A_b = 2\pi R_{id} L \quad (3)$$

Eq. 1 describes the port area A_p as a function of the inner diameter R_{id} . Eq. 2 determines the mass flow \dot{M} by two factors, the oxidizer flow \dot{M}_{ox} and the fuel flow due to the regression, this latter written as a function of A_p , regression rate \dot{r} and fuel density ρ_f . Eq. 3 describes the burnt surface A_b as a function of R_{id} and of length port L .

The main parameter of the internal ballistic is the regression rate \dot{r} . The regression equation from classical texts is reported in Eq. 4, where the coefficients a and n are experimentally determined and are specific to fuel and oxidizer combinations respectively. On the other hand, the mass flux G relation is obtained analytically and presented in Eq. 5. This latter parameter is the driving one for the rocket performance, because it drives the regression of the port section.

$$\dot{r} = a G^n \quad (4)$$

$$G = \frac{\dot{M}}{A_p} \quad (5)$$

The correlation of the regression rate with mass flux is typical of HREs, unlike the SREs that have relationship with the pressure, as described by the Saint/Robert law [6]. So, HREs have a weak correlation with pressure and the dependency on mass flux makes their analysis more difficult. By combining the previous two equations one obtains Eq. 6. Considering the mass flux as constant and simple geometries, it shows an inverse proportionality to the port radius.

$$\dot{r} = a \left(\frac{\dot{M}}{\pi R_{id}^2} \right)^n \quad (6)$$

Finally, comparing Equations 1 and 3, it is possible to notice two of the main HRE problems. The burned surface is always less than the cross-section area and this leads to an inconsistent burn rate with time and an inability to easily scale for larger performances.

1.2 Geometries

The regression rate increase is one of the main focus of research and development of HREs [7]. A possible method to increase the fuel grain surface burn area is the casting of multiple fuel ports, with a large pre-combustion chamber or multiple injectors. This is done to reach highly efficient and stable combustion, by improving the fuel mass flows for oxidizer mass flux levels. However, this multiple port approach presents several disadvantages [8, 9]:

- Fuel regression rate reduction with the increase in the number of ports, due to the drop in oxidizer mass flux and the engine diameter size growth;
- Uneven port burning that produces an excessive un-burned mass fraction. Un-burned fuel slivers can break off the grain and throw chunks out the rocket;
- Increased risk of motor instability related to dynamic flow interactions between ports and/or the presence of a large pre-combustion chamber;
- Requirement for a substantial pre-combustion chamber or individual port injectors and embedded support structure needed for multi-port design regression.

In the last years, rapid prototyping by means of additive manufacturing has been identified to be a technique to produce geometrically complex grain shapes, that may solve the HREs performance problems. New concepts like helix ports, coaxial ports, swirls, spiral-star patterns with central ports have been tested, showing improvements in regression rate [4, 9]. Many researchers considered ABS with Gas Oxygen as propellant, due to the easy shape building of this material using Fusion Deposition Modelling (FDM) processes. Presented analysis has demonstrated that the rotational flow velocity within the fuel port is a primary driver for the regression rate amplification. Centrifugal flow patterns introduced by the embedded fuel port structures dramatically increase fuel regression rates, by enhancing surface skin friction, and reduce the effect of radial boundary layer “blowing” outflow. These two mechanisms work together to enhance the convective heat transfer to the fuel surface [4, 9]. Lastly, it is important to underline the strong dependence between the grain geometry and the thrust profile.

1.3 Printable Fuels

In order to reach sufficient levels of regression rates, it is necessary to implement certain structural and chemical modifications to the fuel grains. A possible solution is the use of paraffin-based fuels, which inherently exhibit a higher regression rate when compared to other polymeric fuels due to the lower viscosity that promotes droplet entrapment within the lame zone [10]. Despite these advantages, paraffin-based fuels suffer from poor mechanical strength compared to polymeric fuels, which led to the utilization of additives such as elastomers and thermoplastics at expense of regression rate [11].

About the possibility of using printable fuels to improve performance, researchers are considering Material Extrusion (ME) processes to print thermoplastic hybrid rocket fuel grains as ABS, HIPS, HDPE and PMMA. In particular, ABS present good properties to replace the HTPB, as reported in the studies conducted by Stephen Whitmore at Utah State University [12, 13].

Whitmore studies led to the development of a model to evaluate the effects of helical fuel ports on regression rates decomposing it into two components, the skin friction enhancement and the radial wall blowing suppression. A comparison between helical geometries of ME-printed ABS burned using gaseous oxygen (GOX) reported an increase of 75% in regression rates, attributed to the rising of skin friction levels due to the helical rotation [14]. ME printed ABS samples were also found to exhibit higher consistency in burn performance, due to the automated nature of the additive manufacturing process [15]. Additionally, ME printing has been highlighted for its efficient use of volume in HREs. Despite its affordability and simplicity, one of the main limitations of thermoplastic material extrusion is the immiscibility of the polymers into efficient, novel, composite hybrid rocket fuels [12].

As mentioned at the beginning of this section, the use of composite fuels can be necessary to improve the performance of HREs, and especially for this reason the ME methods have high importance. Through this process it is possible to build structural frameworks of composite fuel, where a second material can be casted or pulverized in printed molds, obtaining intricate shapes and superior ballistic performances. The additives of the polymeric matrix are usually pulverized coal, petroleum coke, graphite, and paraffin, as mentioned at the beginning of this section. ME printed ABS with casted paraffin composite solid fuel grains are a very new and attractive type of fuel, because the melting

point of the paraffin is lower than the glass transition temperature of the ABS, making the choice of this process perfect for this combination of materials [12].

In literature, there are many studies about increasing the performance of ABS-Paraffin fuel and here are reported some examples. Firstly, a research about a fabricated fuel grain with helical and straight ports with minimal post-processing, facing the incomplete combustion of ABS and marginal regression rate enhancement of hybrid rockets. Ballistic tests conducted using GOX resulted in an amplification of the regression rate of about 6 to 7 times under a low O/F ratio of 0.6 [16]. Secondly, another study on printed ABS substrates with swirl patterns as a framework for casted paraffin, tested using methane and GOX, reported an increase of 20% in regression rate [17]. Thirdly, another research conducted by McCulley, about composite fuel grains with varying amounts of ME-printed ABS and paraffin burned with N₂O, showed a maximum regression rate of 3 mm/s under an O/F ratio of 3.9 [18]. Finally, McKnight made a comparison between star-swirl port samples in ME-printed ABS and different configurations of diaphragm samples in Paraffin and Acrylic. The first ones reported an average regression rate between 0.58 and 1.00 mm/s, whereas the second ones marked between 0.48 and 4.79 mm/s, clearly proving the higher regression rate of paraffin [19]. Over the already mentioned low mechanical strength of the paraffin, there are other negative aspects. The most significant drawback of pristine paraffin is the liquefaction behaviour, opposite to the softening phenomenon in thermoplastics, that makes hard the ME-printing. Hence, printed thermoplastic scaffolds are necessary to incorporate paraffin-based fuels, dramatically reducing the regression rates [12].

In conclusion, it is possible to affirm that the ME printed composite fuel grains made of ABS and paraffin, mostly with helical port, represent one of the best-studied solutions in the field of HRE fuels printable with Additive Manufacturing. So, it could be considered a good beginning point for future research and optimization.

2 METHODOLOGY

Figure 3 portrays the workflow of the multidisciplinary design optimization methodology developed in this work. It illustrates all the key functions, along with their respective input and output parameters, as well as the data structs used, highlighting their main parameters.

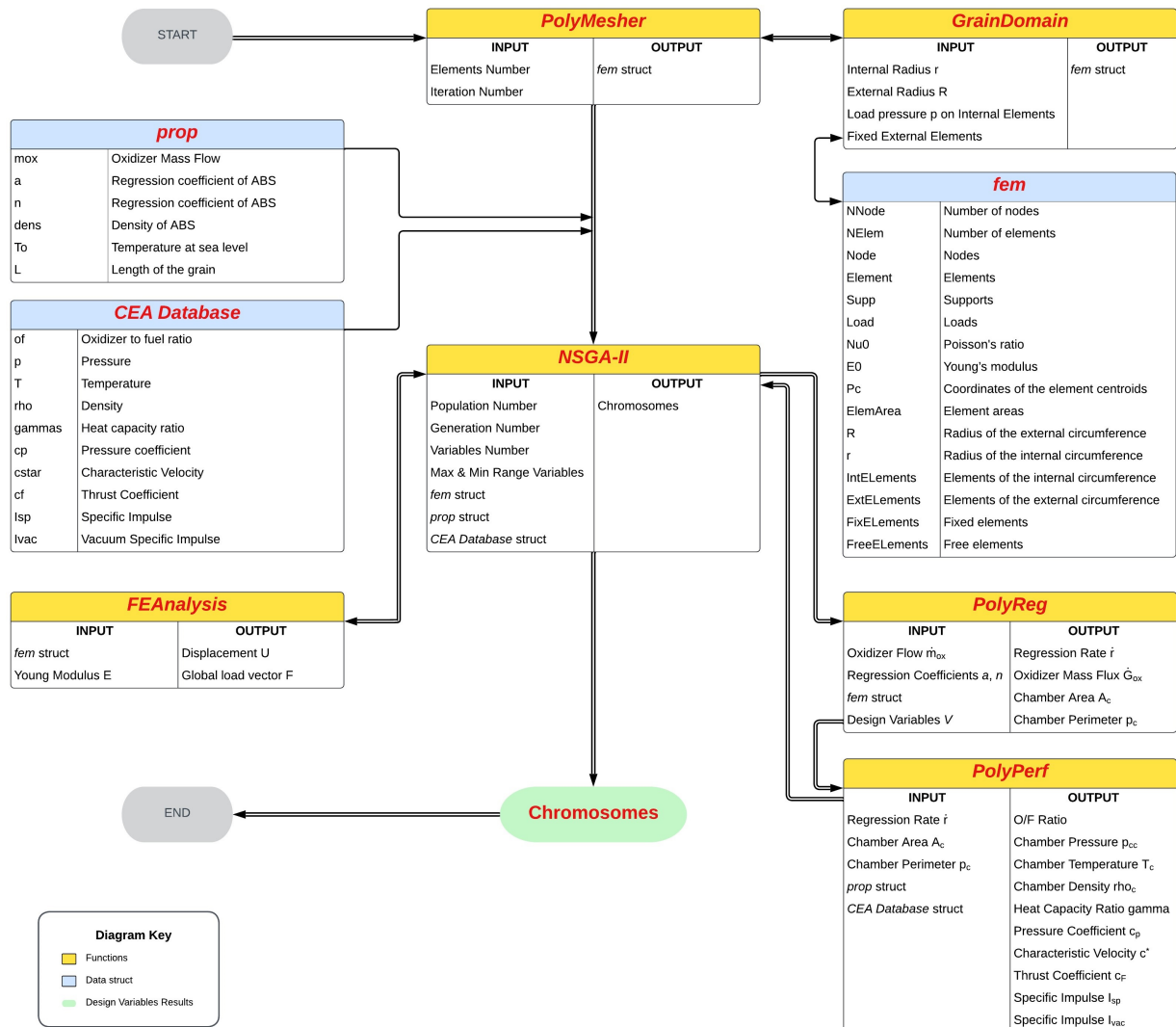


Figure 3: Workflow overview highlighting the main functions and data structures utilized

2.1 Model Definition

This first part consists of propellants selection, creation and definition of the geometry, and all the material and propulsive parameters that will form the model under analysis.

The optimization was performed on a two-dimensional domain due to the greater simplicity of modeling and computation. Based on the surveyed literature and the data availability [20], the ABS was selected as the fuel and gaseous oxygen as the oxidizer.

The grain size and propulsive values were chosen taking into account the state of the art and the possibility of combustion tests to evaluate the results obtained. Specifically, the works of Yu et al., Mahjub et al. and McFarland et al. were considered most [20–22].

Particularly, the regression parameters, introduced in Eq. 4, were taken from the work of McFarland [21], considering 3D-printed ABS itself. On the other hand, the value of the oxidizer flow was established as a function of the previously established geometrical parameters, in order to obtain results comparable with those present in the state of the art. Figure 4 presents the considered domain and Table 2 reports all the main model parameters.

Table 2: Model parameters [23] [20]

Geometry	Material	Propulsion
External Radius / Diameter $R = 2.5 \text{ cm} \quad / \quad D = 5 \text{ cm}$	Young Modulus $E = 2.5 \text{ GPa}$	Oxidizer Flow $\dot{m}_{ox} = 5 \text{ g/s}$
Internal Radius / Diameter $r = 0.75 \text{ cm} \quad / \quad d = 1.5 \text{ cm}$	Poisson's ratio $\nu = 0.36$	Regression coefficient $a = 0.07$
Length $L = 10 \text{ cm}$	Density $\rho = 1.07 \text{ g/m}^3$	Regression coefficient $n = 0.0442$

2.2 Meshing Process

After selecting and defining the parameters, the next step involved creating the mesh of the model. The MATLAB (®) codes `PolyMesher` and `PolyTop` were used for model creation and optimization, respectively [24, 25]. These codes are implemented within the main code, `PolyScript`. The decision to use these codes as a starting point was made for several reasons. Firstly, their versatility in creating meshes for domains with specific geometries makes them particularly suitable, especially when a polygonal mesh is preferred. Additionally, the availability of the codes and well-drafted documentation made them easily accessible, thanks to the work of prof. Paulino on Topology Optimization at Princeton University [26, 27]. Furthermore, the codes exhibit simple, efficient, and robust characteristics, further supporting their selection as the starting point for the mesh creation and optimization process.

The Voronoi tessellation implemented within the meshing process allows the use of polygons for domain creation and requires a brief introduction. The Voronoi tessellation starts with a set of n distinct points or seeds P and creates, for each of these points, sets of other domain points that have a smaller distance to the seed than any other seed. These sets are called Voronoi cells. To achieve a higher level of regularity, the algorithm focuses on a specific category of Voronoi tessellations known as Centroid Voronoi Tessellation (CVT). In a CVT, each point y coincides with the centroid y_c of its corresponding region. For a variational characterization of a CVT, it is possible to compute an energy functional based on the deviation of each region from its reference seed. By minimizing this energy functional, it is possible to converge to an optimal tessellation. `PolyMesher` implements Lloyd's algorithm to compute CVTs and more uniform polygonal meshes. This algorithm iteratively replaces the given generating seeds with the centroids of their corresponding Voronoi regions. It can be seen as a descent method for the energy functional [24]. Figure 4 shows the Voronoi tessellation with Lloyd's iteration method.

The domain is created within the `GrainDomain` function, which is called within the `PolyMesher` function responsible for generating the mesh for the selected domain. The input data for `GrainDomain` includes the inner radius r and outer radius R of the cylindrical domain, as well as the applied pressure p_c as a load condition. As for the load and constraint conditions, highlighted in Fig.4c, the structure is constrained at the nodes along the outer circumference of the domain, while a radial pressure p_c of 1 *bar* is applied at the nodes along the inner circumference of the domain.

The input data for `PolyMesher` are the function containing the domain (`Domain`), the desired number of elements (`NElem`), the maximum number of Lloyd's iterations (`MaxIter`), and optionally, an initial set of seeds (`P`). A double symmetry of the mesh was enforced, and a number of 500 elements were chosen to ensure that the area of each element is larger than the average resolution of FDM printing (thus avoiding printing issues or the need for subsequent AM filters) and to obtain a higher number of iterations and data in the regression analysis, due to the significant number of elements. The obtained mesh is reported in Fig.4.

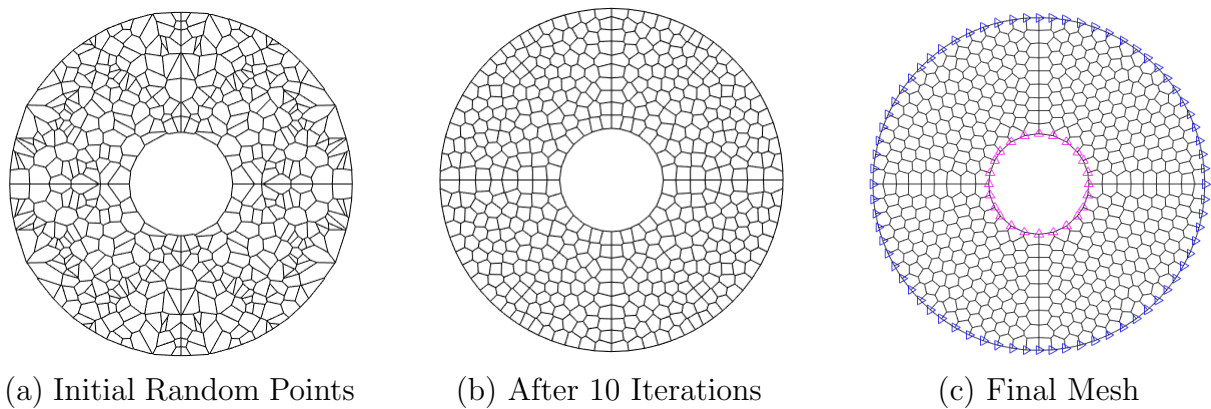


Figure 4: Meshing process with Voronoi tessellation and Lloyd's iteration method

2.3 Structural Analysis

Structural analysis involves applying the Finite Element Method (FEM) to calculate the compliance of the entire structure, which is computed as the dot product of forces and displacements of individual nodes [25]. The subsequent goal of optimization is to minimize this parameter, corresponding to the minimization of structural deformations. This objective is pursued to ensure greater structural integrity of the 3D-printed grain, avoiding excessive deformations during combustion and unexpected variations in rocket propulsion parameters and performance. To perform this analysis, the `FEAnalysis` function within the `PolyTop` code is utilized, and its documentation is referred to for a deeper understanding of the implemented code [25]. `FEAnalysis` enables FEM analysis of meshes composed of polygonal elements. It assembles the stiffness matrix of the entire structure, starting from the stiffness matrix of each polygonal element, and calculates the displacement of each individual node, considering the forces acting on the nodes resulting from the boundary conditions. `FEAnalysis` has been used as an objective function both within `PolyTop` and as a standalone implementation in `NSGA-II`, as will be evident in the subsequent sections. Therefore, the function takes into account the design variables, namely the volume fraction of each element, in the evaluation of the stiffness value for each element.

2.4 Regression Analysis

The grain regression analysis has been implemented through the `PolyReg` function, which has been developed from scratch. `PolyReg` simulates the grain regression based on the geometry of the structure (`fem` struct), the design variables V , and the propulsion parameters (`prop` struct), which have been described in Section 2.1 and are listed in Table 2. Figure 5 illustrates the flowchart of the code.

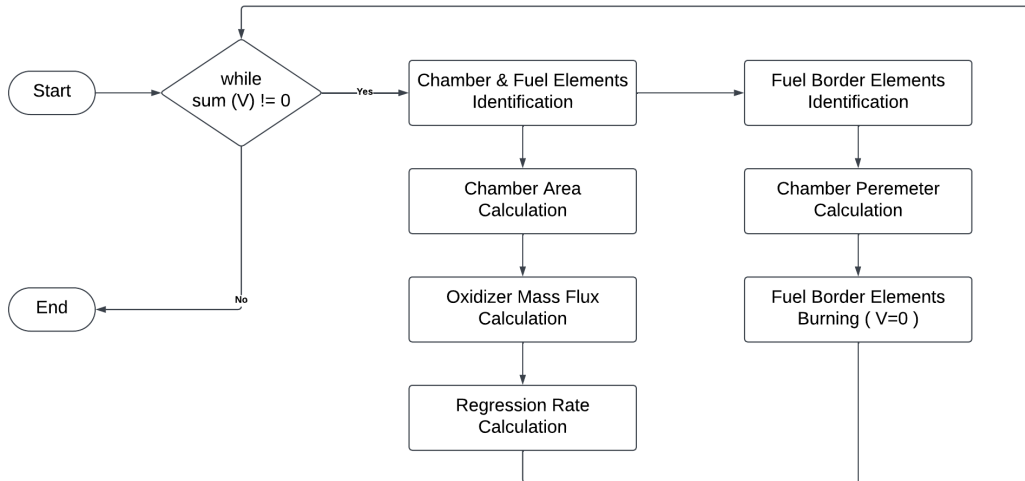


Figure 5: PolyReg Flowchart

Oxidizer mass flux G and regression rates \dot{r} are calculated as follows:

$$\dot{r} = a G_{ox}^n \quad (7)$$

$$G_{ox} = \frac{\dot{m}_{ox}}{A_c} \quad (8)$$

Where a, n are the regression coefficient reported in Table 2, \dot{m}_{ox} is the oxidizer mass flow and A_c is the combustion chamber area.

2.5 Performance Analysis

The calculation of performance is carried out using the `PolyPerf` function, which is also developed from scratch. `PolyPerf` utilizes a database created using NASA CEA Software $\text{\textcircled{R}}$ [28], relying on chemical equilibrium to perform performance analysis. The inputs of this function are the `prop` struct and the `PolyReg` output, i.e. regression rate \dot{r} , chamber area A_c and chamber perimeter p_c .

NASA CEA Software is based on the principle that by knowing the chemical equilibrium compositions of a chemical system, it is possible to calculate the theoretical thermodynamic properties of the system. The documentation of NASA CEA Software is referred to [28–30] for a deeper understanding of the software.

Within `PolyPerf`, various propulsive parameters are calculated, including fuel mass flow \dot{m}_f , combustion chamber mass flow \dot{m}_{cc} , O/F Ratio O/F , and combustion chamber pressure p_{cc} , as described in Eq.9, 10, 11 and 13 [1]. The latter two parameters are used as input for the `CEA Database` data structure, from which the corresponding performance

values are extracted. The workflow of the function is depicted in Fig.6, which also includes all the output parameters of the **CEA Database**. It should be noted that other parameters used as input for the initial iteration include the temperature value $T_0 = 298 K$, the air heat capacity ratio value of $\gamma = 1.4$ (whose values are iteratively updated) and the specific gas constant value of $R = 287 J/(KgK)$.

$$\dot{m}_f = \dot{r} \rho_f p_c L \quad (9)$$

$$\dot{m}_{cc} = \dot{m}_{ox} + \dot{m}_f \quad (10)$$

$$O/F = \frac{\dot{m}_{ox}}{\dot{m}_f} \quad (11)$$

$$p_0 = \frac{\dot{m}_{cc}}{\zeta_d A_{th}} \sqrt{\frac{T_0 R}{\gamma} \left(\frac{\gamma + 1}{2} \right)^{\frac{\gamma+1}{\gamma-1}}} \quad (12)$$

$$p_{cc} = p_0 \left(\frac{T_{cc}}{T_0} \right)^{\frac{\gamma}{\gamma-1}} \quad (13)$$

Regarding the equations presented in this paragraph, some specifications need to be made. In Eq.10, L represents the length of the grain. Equation 12 represents the chamber pressure p_{cc} assuming an ideal nozzle, where ζ_d denotes the discharge correlation factor (due to cooling, changes in specific heat ratios, incomplete combustion) that is considered to be unity. A_{th} represents the throat area of the nozzle and is considered equal to the chamber area A_c . Equation 13 corrects the previous chamber pressure p_0 calculation using the combustion gas burning rate. This assumes an adiabatic expansion with a constant specific heat ratio, which is valid only for small pressure changes [1]. In this equation, T_{cc} represents the chamber temperature, a value provided iteratively by the **CEA Database**.

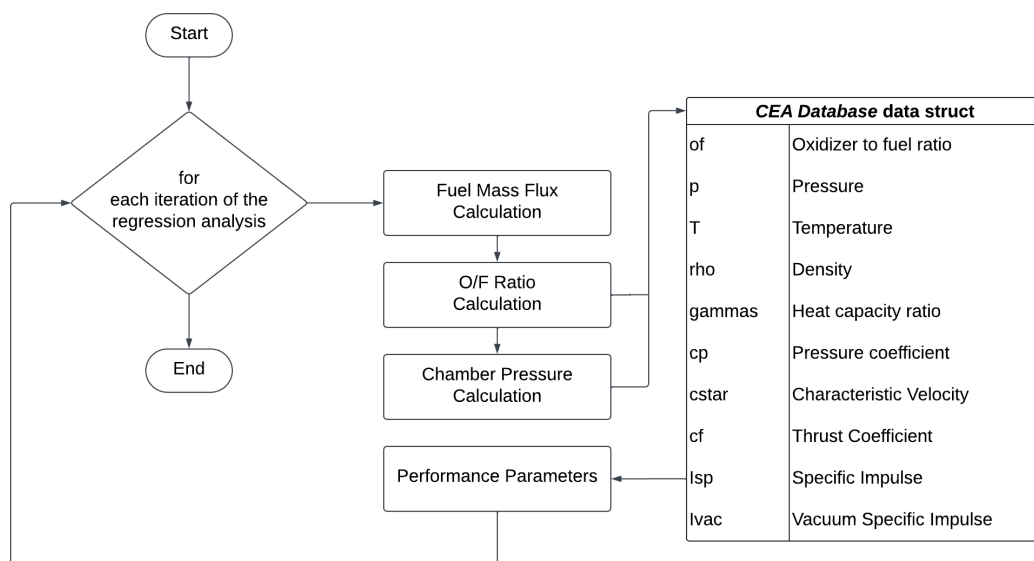


Figure 6: PolyPerf Flowchart

2.6 Topology Optimization

As mentioned in Section 2.2, in the early part of this work, the `PolyTop` function implemented in the `PolyScript` code, along with `PolyMesher`, was used. `PolyTop` performs the topological optimization of the domain generated by `PolyMesher` by implementing the Optimality-Criteria Algorithm (OC) using the SIMP Method. In the later part of this work, it was replaced by `NSGA-II` to leverage the characteristics of genetic algorithms for multidisciplinary optimization of the structural and propulsion aspects. This could not be implemented in `PolyTop` due to the lack of sensitivities regarding the propulsion part, as there are no explicit connections with the geometry.

`PolyTop` internally uses numerous functions, and again, reference is made to the respective documentation for in-depth understanding [25]. The inputs of this function are the geometric and structural parameters, contained in the `fem` struct, and the optimization parameters, contained in the `opt` struct. These are described in Table 3. The design variables considered are the volume fractions of each 'free' element of the domain. The free elements are defined as all elements within the domain to which no loads or constraints are applied, i.e., all elements except those forming the boundary of the outer and inner circumferences. The constraint function is formed by the volume fraction, which represents the percentage of final domain area compared to the initial area. The default objective function is compliance, addressed in Section 2.3, which is minimized until convergence is reached. The output of `PolyTop` consists of the optimal values of the design variables, considering the imposed parameters. Figure 7 shows an example of topology optimization of the domain, performed with an 80% volume fraction constraint.

Table 3: Fields list of the `opt` data struct [25]

<code>opt.zMin</code>	Lower bound for design variables
<code>opt.zMax</code>	Upper bound for design variables
<code>opt.zIni</code>	Initial array of design variables
<code>opt.MatIntFnc</code>	Handle to material interpolation function
<code>opt.P</code>	Matrix that maps design to element variables
<code>opt.VolFrac</code>	Specified volume fraction constraint
<code>opt.Tol</code>	Convergence tolerance on design variables
<code>opt.MaxIter</code>	Max. number of optimization iterations
<code>opt.OCMove</code>	Allowable move step in the OC update scheme
<code>opt.OCEta</code>	Exponent used in the OC update scheme

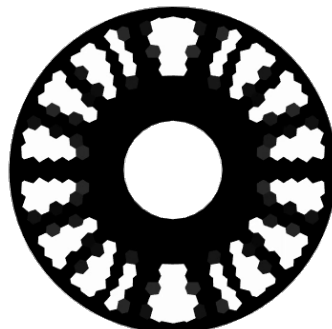


Figure 7: Optimized domain with 80% Volume Fraction constraint

2.7 Multidisciplinary Optimization

Multidisciplinary optimization was implemented using the **NSGA-II** algorithm [31]. The choice of the optimizer fell on genetic algorithms due to the difficulty in computing sensitivities. Genetic algorithms, with their strong numerical nature, can be helpful in cases where the gradient of the objective function is unknown. The only obstacle in this case is the large number of variables, specifically the 404 free elements in the considered mesh, which led to high computational costs and long execution times.

NSGA-II is a non-domination based genetic algorithm for multi-objective optimization, and its documentation is referred to [31] for a complete understanding of the code. In the code, two objective functions were set, namely compliance and specific impulse (I_{sp}). There are 404 design variables, and their lower and upper limits are set to 0 and 1, respectively. It should be noted that, in order to perform the structural analysis, zeros are considered as very small values ($1e-4$) within the **FEAnalysis** function. The remaining input parameters include the population size (**pop**) and the number of generations (**gen**), which determine the size of the initial pool of chromosomes and the number of cross-breeding iterations between chromosome families. Initially, analyses were conducted with $pop = 1000$, which is an order of magnitude larger than the number of design variables, and $gen = 100$, which is the same order of magnitude as the number of design variables. This principle is considered as a rule of thumb and was taken as a starting point, considering a substantial pool relative to the problem at hand. We were aware that further evaluations would be made depending on the results and the required analysis times of the initial tests [32].

Figure 3 in the beginning of this chapter depicts the complete workflow of the main multidisciplinary optimization performed. It includes all the functions used, along with their input and output parameters, as well as all the data structs utilized, highlighting their key parameters. It can be observed that the workflow begins with the model definition, involving the construction of the **fem** struct using the **GrainDomain** and **PolyMesher** functions. The core of the process is represented by the **NSGA-II** genetic algorithm, which takes all the data structs (**fem**, **prop**, **CEA Database**) as input. **NSGA-II** invokes the **FEAnalysis** function to compute the structural objective function and the **PolyReg** and **PolyPerf** functions in sequence to calculate the propulsion objective function. At the end of the analysis, the chromosomes are obtained, representing the set of design variables for each population family along with their respective objective function values. These chromosomes are then used to form the Pareto front of the obtained solutions, which represents their visualization on a graph with the two considered objective functions as the x and y axes.

3 PRELIMINARY RESULTS

3.1 Regression Results

To validate the regression analysis performed using the `PolyReg` function, a combustion test simulation was conducted for which experimental results were known. The geometric and propulsive conditions reported by Yu et al. [20] were recreated, and the regression coefficients mentioned in Section 2.1 and Table 2 were used. The grain considered has a cylindrical type with $R = 8 \text{ mm}$ and $r = 2.3 \text{ mm}$. Fuel and oxidizer were consistently considered as ABS and pure gaseous oxygen, respectively. Specifically, the regression coefficients for printed ABS with 100% packing density were considered. Figure 8 presents the data obtained using the `PolyReg` function alongside the experimental data reported in the work of Yu et al. [20], regarding the printed ABS with 100% packing density and solid ABS. One can observe that the regression rate trends are completely overlapping with the experimentally obtained ones, indicating the effectiveness of the developed code. Furthermore, it is noticeable that the regression rate of printed ABS is higher than that of solid ABS, and for a detailed understanding of this phenomenon, reference is made to the aforementioned paper [20].

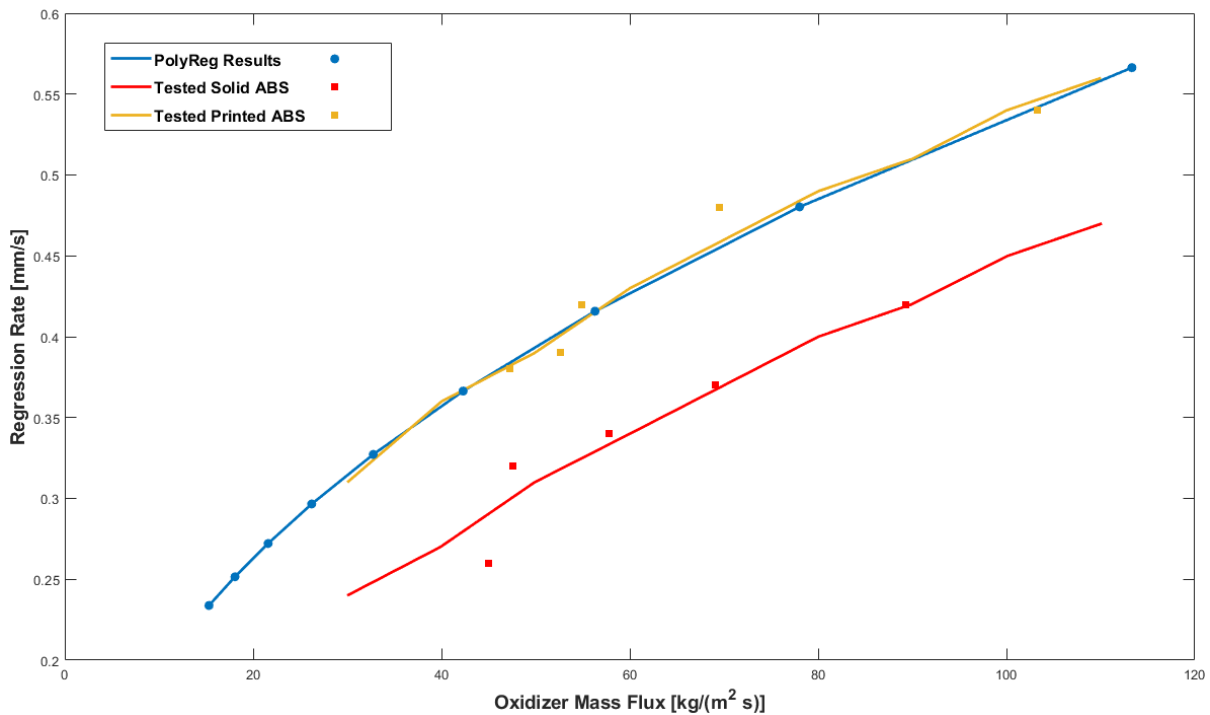


Figure 8: Comparison of regression rate values obtained experimentally and by `PolyReg` [20]

3.2 Multidisciplinary Design Optimization Results

Several multidisciplinary optimizations were conducted with the aim of minimizing compliance and maximizing specific impulse. The same mesh, as previously shown, was used to allow for comparisons between different analyses. Various genetic parameters were tested to obtain the best possible analyses, balancing both the quality of the results and the predicted simulation times. Figures 9, 10 and 11 depict the three main analyses performed, which yielded the best results. All the figures show the obtained solutions plotted in blue on a graph with compliance on the x-axis and specific impulse on the y-axis. Table 4 presents the key data of the performed analyses, including the initial genetic parameters, the values of the objective functions, and the corresponding figure for each analysis.

Table 4: Analysis results

Analysis	Population	Generation	$I_{sp_{max}}$	$Compliance_{min}$	Figure
1	1000	100	108.4 s	$3.8 \cdot 10^{-3}$	Figure 9
2	2000	50	98.5 s	$4.2 \cdot 10^{-3}$	Figure 10
3	2000	90	100.3 s	$3.2 \cdot 10^{-3}$	Figure 11

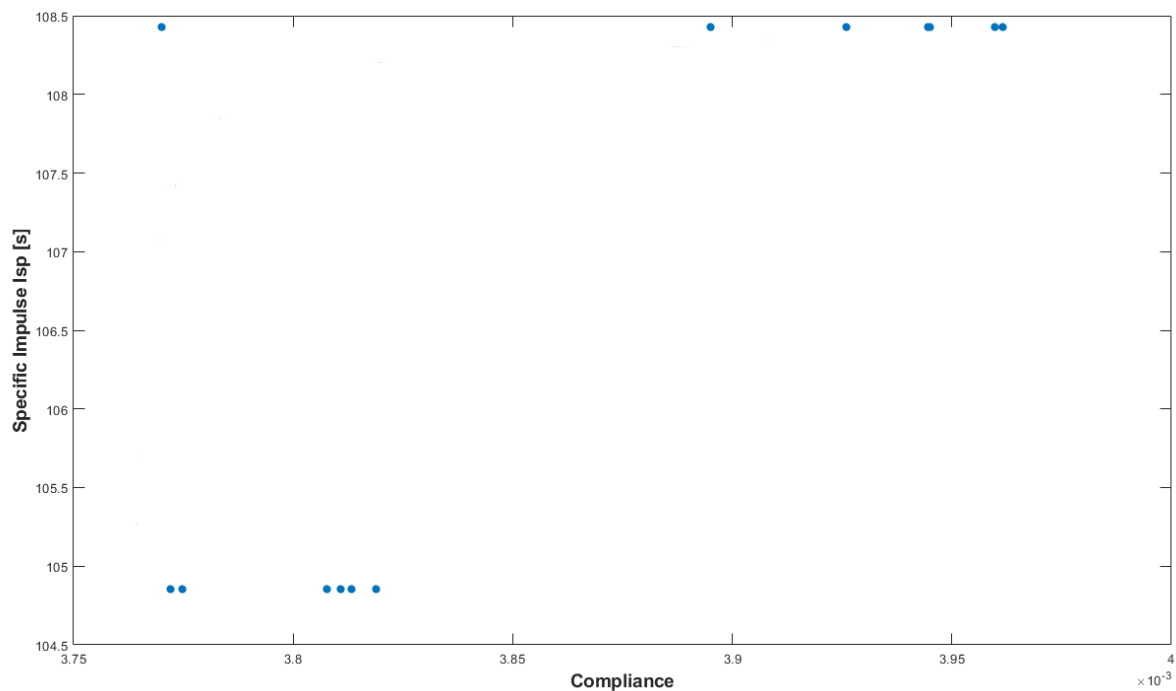


Figure 9: Pareto front of the MTO analysis with 1000 populations and 100 generations

Upon analyzing the obtained Pareto fronts resulting from the multidisciplinary design optimizations, it becomes evident that the values of the objective functions are significantly influenced by the initial population. While the Pareto fronts generally exhibit the expected behaviour, it is important to note that a considerable portion of the designs still hold a rank higher than 1, indicating that the solutions have not yet fully converged.

This is expected due to the high-dimensionality of the problem, with numerous design variables and complex interactions.

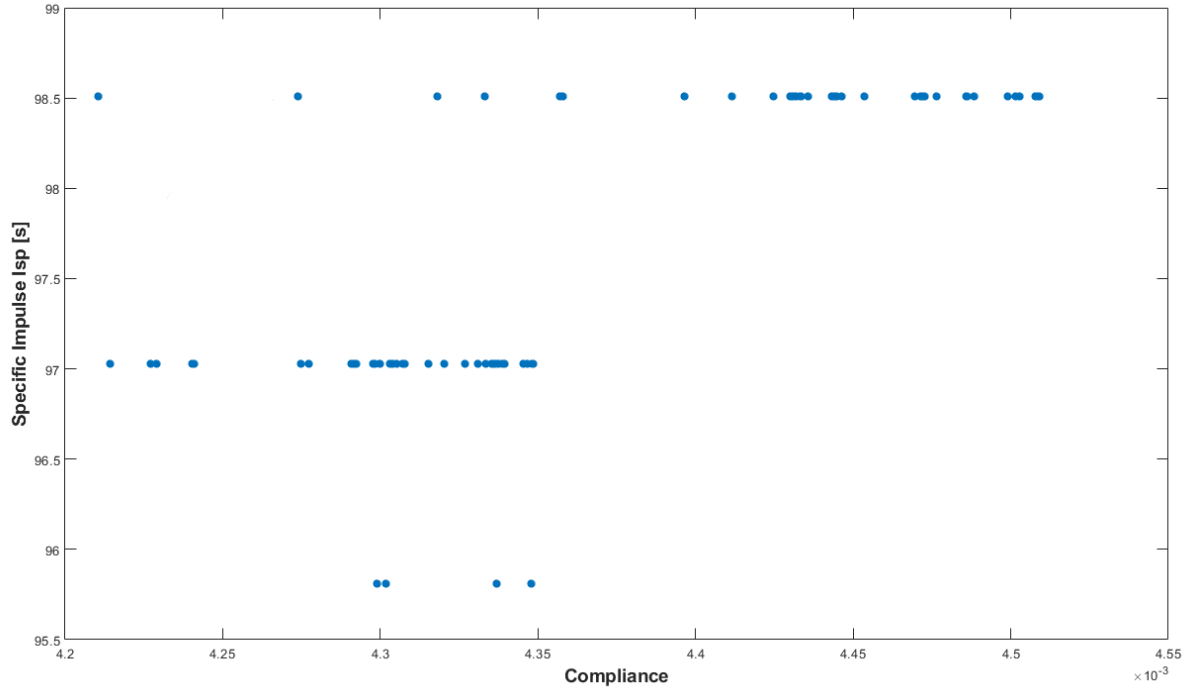


Figure 10: Pareto front of the MTO analysis with 2000 populations and 50 generations

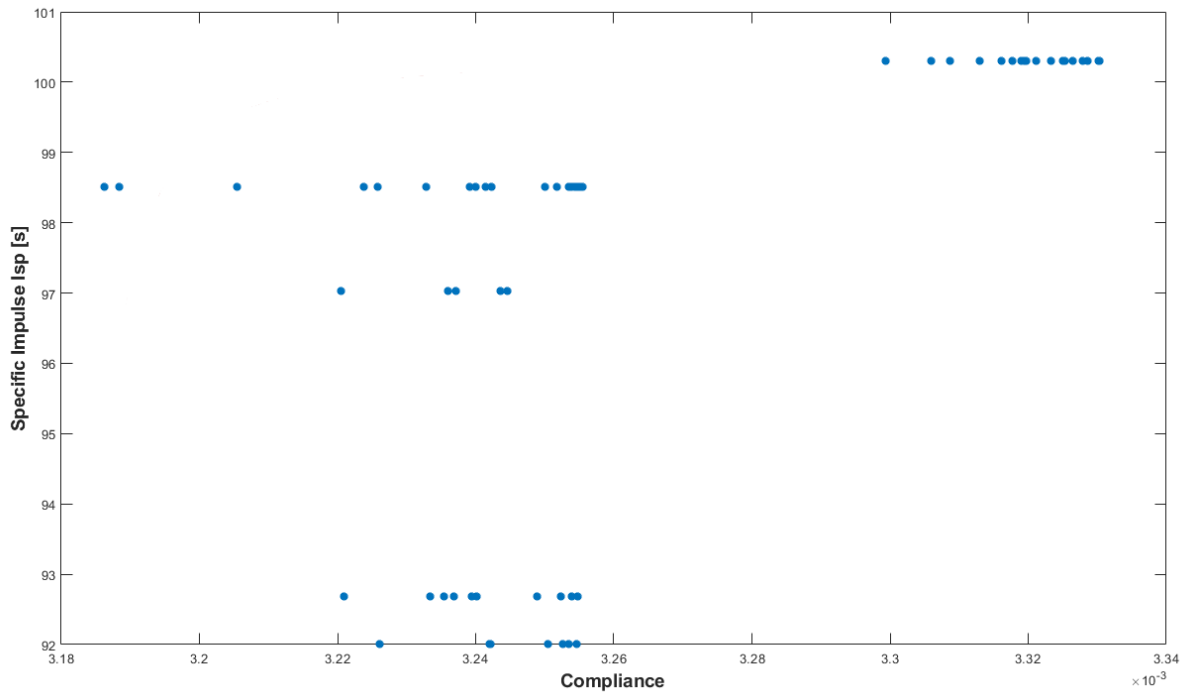


Figure 11: Pareto front of the MTO analysis with 2000 populations and 90 generations

4 CONCLUSIONS

The regression analysis, based on the work of Yu et al. [20], effectively simulated the combustion test by utilizing the same geometric and propulsive data.

The results obtained from the multidisciplinary design optimizations using genetic algorithms have provided valuable insights.

One noteworthy observation is the considerable influence of the initial population configuration on the outcomes of the optimizations conducted, as expected given the large number of design variables. By observing Table 4, in Section 3.2 it is evident that there are significant variations in the optimized values of compliance and specific impulse (Isp) among the different analyses. This suggests that increasing the number of populations may be necessary to obtain more robust and reliable results. However, it is important to consider the trade-off between the quality of results and the computational time, as the latter tends to exponentially increase with larger population sizes.

In terms of the specific impulse maximization, no well-defined geometric pattern emerged, contrasting with the compliance minimization which involved topological optimization. This indicates a weak correlation between Isp and the geometric shape of the grain, warranting further investigation into the underlying factors that contribute to Isp .

Overall, these findings highlight the complexity and multifaceted nature of the optimization process, emphasizing the need for thorough analysis and exploration to achieve optimal performance in hybrid rocket fuel design.

5 FUTURE PERSPECTIVES

The findings presented in this paper should be considered as an initial phase of a broader study that requires further exploration. Additional analyses will be conducted to obtain more robust results, and the implementation of further investigations or research avenues is also being considered to expand the horizons of this research.

These possibilities include:

- **Multidisciplinary Design Optimization Analyses**

Leveraging advanced MDO strategies can further enhance the efficiency and effectiveness of the optimization process. This will lead to the discovery of optimal designs that maximize performance while considering multiple disciplines simultaneously.

- **Experimental Validation**

Future studies should prioritize the execution of comprehensive experimental campaigns to validate the performance of hybrid rocket engines under real-world conditions, by conducting side-by-side comparisons of: different fuel combinations, propellant formulations, and combustion strategies.

- **Exploration of Novel Fuel Combinations**

Analyzing and evaluating different fuel combinations is crucial to expanding the performance capabilities of hybrid rocket engines. Future research should focus on investigating alternative propellants, exploring high-energy additives, and considering sustainable and eco-friendly fuel options. By studying the impact of these fuel combinations on performance and optimization, it is possible to unlock new possibilities and advance the work towards greener and more efficient propulsion systems.

- **Advancement towards 3D Geometries**

Transitioning from 2D to 3D geometries will provide greater freedom and flexibility for optimization, enabling the exploration of more complex designs. Future research should focus on developing advanced computational models capable of capturing the intricate flow and combustion phenomena in three-dimensional space.

- **Integration of Computational Fluid Dynamics (CFD) Analyses**

Implementing CFD analyses into the design and optimization process will significantly contribute to improved thermal and propulsion results. By studying the evolution of the flow field, heat transfer, and combustion processes, CFD simulations can provide valuable insights into the performance characteristics of hybrid rocket engines.

ACKNOWLEDGMENTS

This work was supported by Fundação para a Ciência e a Tecnologia (FCT), through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] G. P. Sutton and O. Biblarz. *Rocket propulsion elements*. John Wiley Sons Inc., 2017.
- [2] P. M. Zolla, M. T. Migliorino, D. Bianchi, F. Nasuti, R. C. Pellegrini, and E. Cavallini. A computational tool for the design of hybrid rockets. *Aerotecnica Missili Spazio*, 100(253–262), 2021. doi:10.1007/s42496-021-00085-3.
- [3] K. K. Kuo and M. J. Chiaverini. *Fundamentals of Hybrid Rocket Combustion and Propulsion*. American Institute of Aeronautics and Astronautics, 2007.
- [4] M. P. King. Analysis for hybrid rocket fuel regression using stereolithographic geometry. Master’s thesis, Air Force Institute of Technology, 2020.
- [5] W. Henry. Hybrid rocket education at the u.s. air force academy. *28th Joint Propulsion Conference and Exhibit*, 1992. doi:10.2514/6.1992-3300.
- [6] R. A. Braeunig. Rocket and space technology, 2012. URL <http://www.braeunig.us/space/propuls.htm#thrust>.
- [7] Y. Pal, S. N. Mahottamananda, S. K. Palateerdham, S. Subha, and A. Ingenito. Review on the regression rate-improvement techniques and mechanical performance of hybrid rocket fuels. *FirePhysChem*, 1(4), 2021. doi:10.1016/j.fpc.2021.11.016.
- [8] A. Karabeyoglu. Hybrid rocket propulsion for future space launch. *Aero/Astro 50th Year Anniversary*, 2008.
- [9] S. A. Whitmore, S. D. Walker, D. P. Merkley, and M. Sobbi. High regression rate hybrid rocket fuel grains with helical port structures. *Journal of Propulsion and Power*, 31(6), 2015. doi:10.2514/1.B35615.
- [10] A. Petrarolo, M. Kobald, and S. Schlechtriem. Optical analysis of the liquid layer combustion of paraffin-based hybrid rocket fuels. *Acta Astronautica*, 158, 2019. doi:10.1016/j.actaastro.2018.05.059.
- [11] M. Boiocchi, P. Milova, L. Galfetti, L. Landro, and A. Golovko. A wide characterization of paraffin-based fuels mixed with styrene-based thermoplastic polymers for hybrid propulsion. 2016. doi:10.1051/eucass/201608241.
- [12] C. Oztan. Utilization of additive manufacturing in hybrid rocket technology: A review. *Acta Astronautica*, 180, 2021. doi:10.1016/j.actaastro.2020.11.024.
- [13] S. A. Whitmore, Z. Peterson, and S. Eilers. *Analytical and Experimental Comparisons of HTPB and ABS as Hybrid Rocket Fuels*. 2011. doi:10.2514/6.2011-5909.
- [14] S. A. Whitmore and S. D. Walker. Engineering model for hybrid fuel regression rate amplification using helical ports. *Journal of Propulsion and Power*, 33(2), 2017. doi:10.2514/1.B36208.
- [15] S. A. Whitmore, Z. W. Peterson, and S. D. Eilers. Comparing hydroxyl terminated polybutadiene and acrylonitrile butadiene styrene as hybrid rocket fuels. *Journal of Propulsion and Power*, 29(3), 2013. doi:10.2514/1.B34382.
- [16] L. Pabarcus. Development and preliminary testing of paraffin hybrid rocket fuel grains with helical port structures. *AIAA Propulsion and Energy 2019 Forum*, 2019. doi:10.2514/6.2019-4187.

-
- [17] Z. Wang, X. Lin, F. Li, and X. Yu. Combustion performance of a novel hybrid rocket fuel grain with a nested helical structure. *Aerospace Science and Technology*, 97, 2020. doi:10.1016/j.ast.2019.105613.
- [18] J. M. McCulley. Design and testing of digitally manufactured paraffin acrylonitrile-butadiene-styrene hybrid rocket motors, 2012.
- [19] B. R. McKnight, J. E. Boyer, P. K. Nardozzo, and A. Cortopassi. Design and testing of an additively manufactured advanced hybrid rocket motor propulsion unit for cubesats (puc). *51st AIAA/SAE/ASEE Joint Propulsion Conference*, 2015. doi:10.2514/6.2015-4036.
- [20] X. Yu, H. Yu, W. Zhang, L. T. DeLuca, and R. Shen. Effect of penetrative combustion on regression rate of 3d printed hybrid rocket fuel. *Aerospace*, 9(11), 2022. doi:10.3390/aerospace9110696.
- [21] M. McFarland and E. Antunes. Small-scale static fire tests of 3d printing hybrid rocket fuel grains produced from different materials. *Aerospace*, 6(7), 2019. doi:10.3390/aerospace6070081.
- [22] A. Mahjub, Q. Azam, M. Z. Abdullah, and N. M. Mazlan. Cad-based 3d grain burnback analysis for solid rocket motors. In *Proceedings of International Conference of Aerospace and Mechanical Engineering 2019*. Springer Singapore, 2020. doi:10.1007/978-981-15-4756-0_28.
- [23] omnexus.specialchem.com. URL <https://omnexus.specialchem.com/selection-guide/acrylonitrile-butadiene-styrene-abs-plastic>.
- [24] C. Talischi, G. Paulino, A. Pereira, and I. Menezes. Polymesh: a general-purpose mesh generator for polygonal elements written in matlab. *Structural and Multidisciplinary Optimization*, 45:309–328, 2012. doi:10.1007/s00158-011-0706-z.
- [25] C. Talischi, G. Paulino, A. Pereira, and I. Menezes. Polytop: a matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. *Structural and Multidisciplinary Optimization*, 45, 2012. doi:10.1007/s00158-011-0696-x.
- [26] G. Paulino. URL <http://paulino.princeton.edu/index.html>.
- [27] C. Talischi, G. Paulino, A. Pereira, and I. Menezes. Polymesh: a general-purpose mesh generator for polygonal elements written in matlab. In *11th U.S. National Congress on Computational Mechanics*, Minnesota, USA, July 2011.
- [28] NASA. Nasa cea run. URL <https://cearun.grc.nasa.gov/>.
- [29] B. J. McBride and S. Gordon. *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. I. Analysis*, 1994.
- [30] B. J. McBride and S. Gordon. *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications. II. Users Manual and Program Description*, 1996.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2), 2002. doi:10.1109/4235.996017.
- [32] J. R. R. A. Martins and A. Ning. *Engineering Design Optimization*. Cambridge University Press, 2022.



MULTI-OBJECTIVE AEROELASTIC ANALYSIS AND OPTIMIZATION USING SURROGATE MODELS

Alessandra Lunghitano^{1*}, Frederico Afonso¹ and Afzal Suleman^{1,2}

1: IDMEC
Instituto Superior Técnico
Universidade de Lisboa
Av. Rovisco Pais, 1049-001, Lisboa, Portugal
{alessandra.lunghitano, frederico.afonso}@tecnico.ulisboa.pt,
<https://www.idmec.tecnico.ulisboa.pt>

2: Center for Aerospace Research
Department of Mechanical Engineering
University of Victoria
V8W 2Y2, Victoria, BC, Canada
suleman@uvic.ca, <https://www.uvic-cfar.com/>

Abstract. *Aeroelastic analysis plays a crucial role in the design and evaluation of aircraft structures, ensuring their structural integrity and dynamic stability under aerodynamic loads. However, conducting detailed aeroelastic simulations using high-fidelity computational methods can be computationally expensive. This paper proposes an approach to reduce the computational cost of aeroelastic analyses through the use of surrogate modeling techniques based on machine learning. Surrogate models act as efficient approximations of the complex aeroelastic simulations, providing accurate predictions of critical parameters while significantly reducing the computational cost. The Goland wing is utilized as a the baseline configuration, and aeroelastic simulations are performed using the SHARPy framework, an open-source Python tool developed by Imperial College London, to generate the data-set required for surrogate model training. The trained surrogate regression models are capable of predicting important aeroelastic quantities, including flutter speed, mass, and lift and drag coefficients. Moreover, a multi-objective optimization framework is employed to maximize the lift-to-drag ratio and minimize the structural mass while considering a flutter constraint, improving the effectiveness of the preliminary design process.*

Keywords: Aeroelasticity, multidisciplinary design optimization, multi-objective optimization, wing design, surrogate models

1 INTRODUCTION

The study of aeroelastic phenomena, namely flutter, at the early aircraft design stages is becoming crucial given the trend of increasing wing slenderness to improve aerodynamic performance [1]. To face this challenge, researchers are planning to leverage from Multidisciplinary Design Optimization (MDO) strategies to achieve optimal solutions that are able to maximize performance while complying with flutter constraints [2]. However, the main hurdle that these MDO strategies face is the high computational cost associated with aeroelastic analyses, especially those presenting a nonlinear behavior [2].

With the goal of minimizing the computational cost in mind, different approaches are being considered, namely multi-fidelity models [3, 4], conventional [5] and machine learning-based [6] surrogate models, and reduced-order models [7].

Low-fidelity (LF) [8], high-fidelity (HF) [9, 10], and multi-fidelity (MF) [11, 12] models have all been tested in different flutter constrained MDO problems of wings. These models span from LF models such as potential flow theory and panel methods with corrections [8, 11, 12], accounting for both compressibility and viscous effects, to Euler [11] and HF Reynolds Averaged Navier-Stokes (RANS) solvers [9, 10, 12]. Most of these optimization problems aimed at maximizing range [8, 9, 11], although fuel burn [12] and structural mass [10] minimization has also been chosen.

Even though reduced-order and surrogate models have been applied to aeroelasticity problems for some time [13], their usage for MDO problems considering flutter is scarce [14] in open literature, in particular for MDO problems considering aerodynamic performance and structural weight. The same observation is noticed for machine learning-based strategies. Nevertheless, this research field is starting to emerge. For instance, Sohst et al. [15] developed a MDO strategy that resorts to multi-fidelity solvers and surrogate models to design strut-braced and high aspect-ratio wings considering flutter and stress constraints. They found that although the flutter constraint was respected in the optimization process, unaccounted buckling was observed in a nonlinear assessment of the optimized strut-braced wing aircraft design. Toffol and Ricci [16] developed a methodology combining in-house codes with surrogate models to optimize the structural layout of a conventional aircraft such that its mass is minimized while considering stress and flutter constraints. Moreover, the scientific community has been conducting work on specific unsteady nonlinear aeroelasticity problems that arise from unconventional features, such as limit cycle oscillations [17–20].

The current paper aims to provide a different surrogate-based optimization methodology to conceptually design wings considering aerodynamics and structural metrics while respecting a flutter constraint.

2 METHODS AND TOOLS

The developed methodology follows the 3 common steps in an offline¹ surrogate-based optimization: (i) data-set definition and generation; (ii) surrogate model training, validation and testing; and (iii) optimization process. All these tasks were done using available open-source Python codes and especially implemented Python scripts for this work.

Firstly, the Goland’s wing [21] was selected to be the baseline model for this work.

¹In other words, no infill points are used to update the surrogate model during the optimization process.

The reason for this choice was two-fold: (i) the existence of experimental data to validate the implementation; and (ii) since it was already used to validate SHARPy [22], the open-source code chosen for the aeroelastic simulations. SHARPy provides a coupled aeroelastic solution, considering the Unsteady Vortex Lattice Method (UVLM) [23] and Geometrically Exact Beam Model (GEBM) [24, 25] for aerodynamics and structural analyses, respectively. Krylov and modal projections are used in SHARPy to linearize aerodynamic and structural subsystems, respectively. These are then used for estimating the flutter speed by means of the iterative p-k method at the nonlinear aeroelastic equilibrium. The flutter speed was predicted in SHARPy implementation used for this work to be under 5% of relative error, in agreement with the value obtained by SHARPy developers [26].

In what concerns the design variables, the choice rested on wing aspect-ratio (AR), sweep angle (Λ), torsional stiffness (GJ), and angle of attack (α). The upper and lower boundaries of these parameters are listed in Table 1 alongside the initial value that corresponds to the Goland's Wing.

Table 1: Design variables and respective boundaries used for the data-set generation and optimization problem.

Design Variable	Initial	Lower Boundary	Upper Boundary	Units
Aspect-ratio (AR)	6.67	6	16	-
Sweep angle (Λ)	0	0	40	deg
Torsional stiffness (GJ)	0.99×10^6	0.70×10^6	1.70×10^6	N.m ²
Angle of attack (α)	-	-5	15	deg

Latin Hypercube Sampling (LHS) was chosen to generate a representative data-set of this design space, although other sampling techniques could have been employed. For this work, 1480 samples were built using the LHS implementation from pyDOE2 [27] open-source package. Subsequently, aeroelastic simulations were run in SHARPy for each sample to determine lift (C_L) and drag (C_D) coefficients of the wing at cruise conditions (a speed of $V_{cr} = 30$ m/s and sea level conditions), its mass, and flutter speed. Then, this data-set was divided in 60% for training the surrogate model and 30% for validating it, reserving 10% for testing.

Different machine learning models from the scikit-learn Python library [28] were tested to select the one that yields the highest R^2 and the lowest mean average error (MAE) in the validation phase. These included the following regression-based models: Bayesian Ridge, Decision Tree Regressor, Extra Trees Regressor, Lasso, ARD Regression, and Linear Regression. In the end, the Extra Trees Regressor was selected since it presented the highest R^2 and the lowest MAE, as can be seen in Table 2.

To have a more clear understanding of the suitability of the chosen surrogate model for aeroelastic analyses throughout the design space, a visualization of parameters used to compose the objective function (lift-to-drag ratio and mass) and constraint (flutter speed) is provided in Figure 1 for the testing data-set.

As one can notice from Figure 1, the surrogate model predicts well both the mass ($R^2 = 9.99 \times 10^{-1}$) and lift coefficient ($R^2 = 9.98 \times 10^{-1}$). Slightly higher discrepancies between surrogate and SHARPy results ($R^2 = 9.97 \times 10^{-1}$) are observed in the drag coefficient, especially for higher values. However, it is for the flutter speed that the differences

Table 2: Comparison between different machine learning models, available in the scikit-learn python library [28] to evaluate their suitability for capturing the aeroelastic features of the problem at hand.

Method	R^2 (-)	Mean Average Error (%)
Bayesian Ridge	2.48×10^{-1}	26.7
Decision Tree Regressor	5.83×10^{-1}	19.88
Extra Tree Regressor	9.17×10^{-1}	8.96
Lasso	2.47×10^{-1}	26.7
ARD Regression	2.50×10^{-1}	26.7
Linear Regression	2.49×10^{-1}	26.7

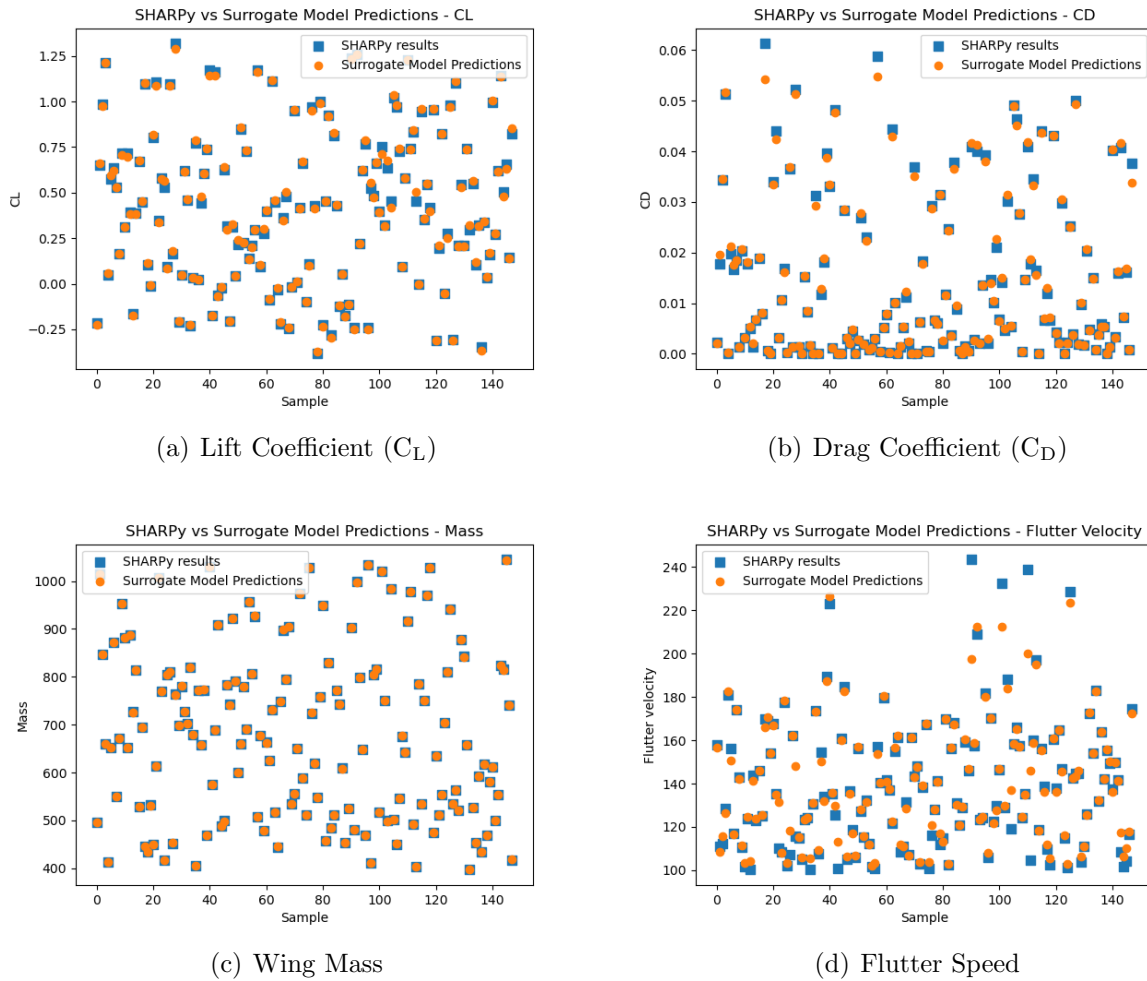


Figure 1: Visual comparison of the results obtained using the SHARPy simulations (in blue) and the surrogate model (in orange) in the testing phase of the surrogate model.

are higher ($R^2 = 9.17 \times 10^{-1}$). This is an expected outcome since flutter speed ($V_{flutter}$) is estimated through an iterative algorithm, namely the p-k method. A remark should be made in regard to the drag coefficient which for several designs is two orders of magnitude lower than usual values (in the order of 10^{-2}) since no viscous nor compressibility² effects

²This latter effect is not relevant for the considered cruise speed although it might become relevant if cruise speed is increased.

are considered in the aeroelastic simulations. There are some effort being conducted by SHARPy team [29] to include viscous considerations in the UVLM, which are typically not of major importance when the focus is on aeroelastic phenomena. Furthermore, some high relative errors were noticed during the validation phase for the lowest values of drag that are diluted in the MAE. As a consequence, the lift-to-drag ratio is expected to be two orders of magnitude higher than usual. Nevertheless, these values were deemed acceptable for testing the methodology developed for this work.

The last step is solving the optimization problem whose statement is:

$$\begin{aligned}
 & \text{minimize} && f(-C_L/C_D, \text{mass}) \\
 & \text{with respect to} && x = (\text{AR}, \Lambda, GJ, \alpha) \quad , \\
 & \text{subject to} && c = V_{cr} - (V_{flutter}/1.5) \leq 0
 \end{aligned} \tag{1}$$

where f , x , and c denote the classical notation for objective functions, design variable set, and constraints, respectively. Since this is a multi-objective problem and the run time of the surrogate model is inexpensive a gradient-free optimizer was chosen to better explore the design space, namely the implementation in pygmo [30] of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [31].

3 PRELIMINARY RESULTS

A population size of 100 individuals and 284 generations were considered for the first optimization problem. The Pareto front followed the expected trend: (i) when C_L/C_D is the objective, the optimizer prioritize aspect-ratio increase and zero angle of attack to reduce the induced drag; and (ii) the opposing effect is noticed for mass minimization, where the optimizer led the aspect-ratio to its minimum value and increased the angle of attack.

Additionally, as reported earlier, the lowest values of drag coefficient calculated by the surrogate model present higher relative errors and provide an underestimation of it in regard to the SHARPy simulations. These discrepancies between optimal solutions obtained with the surrogate model and aeroelastic simulations are shown in Table 3.

Table 3: Relative differences between surrogate model and aeroelastic simulations for the designs that maximize C_L/C_D and minimize mass.

Objective	Difference in C_L/C_D (%)	Difference in Mass (%)
C_L/C_D maximization	14.9	0.03
Mass minimization	5.1	0.21

Regarding the structural mass estimation the relative difference is lower than 1% highlighting the accuracy of the surrogate model is estimating this parameter.

4 CONCLUDING REMARKS

A surrogate-based optimization methodology for designing wings considering both performance and aeroelasticity metrics was successfully implemented recurring to open-source Python codes. It was tested to solve a multi-objective optimization problem, using 1480 samples generated by means of LHS to train and validate an Extra Trees Regressor surrogate model. The testing showed that the chosen surrogate model adequately represents the

SHARPy solver. However, this methodology still requires further improvements, namely in what concerns the surrogate model accuracy to predict the flutter speed and lift-to-drag ratio. Additional optimizations with different parameters, for instance increasing the cruise speed, need to be conducted to gain further insights. Moreover, a multi-fidelity approach to run HF RANS simulations on the wing nonlinear aeroelastic equilibrium at cruise conditions could improve the accuracy of the aerodynamic objective.

ACKNOWLEDGEMENTS

This work was supported by Fundação para a Ciência e Tecnologia (FCT), through IDMEC, under LAETA, project UIDB/50022/2020.

REFERENCES

- [1] F. Afonso, J. Vale, Éder Oliveira, F. Lau, and A. Suleman. A review on non-linear aeroelasticity of high aspect-ratio wings. *Progress in Aerospace Sciences*, 89:40–57, 2017. doi:10.1016/j.paerosci.2016.12.004.
- [2] E. Jonsson, C. Riso, C. A. Lupp, C. E. Cesnik, J. R. Martins, and B. I. Epureanu. Flutter and post-flutter constraints in aircraft design optimization. *Progress in Aerospace Sciences*, 109:100537, 2019. doi:10.1016/j.paerosci.2019.04.001.
- [3] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. *SIAM Review*, 60(3):550–591, 2018. doi:10.1137/16M1082469.
- [4] M. Giselle Fernández-Gonino, C. Park, N. H. Kim, and R. T. Haftka. Issues in Deciding Whether to Use Multifidelity Surrogates. *AIAA Journal*, 57(5):2039–2054, 2019. doi:10.2514/1.J057750.
- [5] R. Yondo, E. Andrés, and E. Valero. A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in Aerospace Sciences*, 96:23–61, 2018. doi:10.1016/j.paerosci.2017.11.003.
- [6] J. Li, X. Du, and J. R. Martins. Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, 134:100849, 2022. doi:10.1016/j.paerosci.2022.100849.
- [7] G. Mendonça, F. Afonso, and F. Lau. Model order reduction in aerodynamics: Review and applications. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(15):5816–5836, 2019. doi:10.1177/0954410019853472.
- [8] C. Riso and C. E. S. Cesnik. Impact of Low-Order Modeling on Aeroelastic Predictions for Very Flexible Wings. *Journal of Aircraft*, 60(3):662–687, 2023. doi:10.2514/1.C036869.
- [9] A. C. Gray, C. Riso, E. Jonsson, J. R. R. A. Martins, and C. E. S. Cesnik. High-Fidelity Aerostructural Optimization with a Geometrically Nonlinear Flutter Constraint. *AIAA Journal*, 61(6):2430–2443, 2023. doi:10.2514/1.J062127.

- [10] E. Jonsson, C. Riso, B. B. Monteiro, A. C. Gray, J. R. R. A. Martins, and C. E. S. Cesnik. High-fidelity gradient-based wing structural optimization including geometrically nonlinear flutter constraint. *AIAA Journal*, 61(7):3045–3061, 2023. doi:10.2514/1.J061575.
- [11] D. Bryson, M. Rumpfkeil, and R. Durscher. Framework for Multifidelity Aeroelastic Vehicle Design Optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Denver, Colorado, 2017. doi:10.2514/6.2017-4322.
- [12] A. S. Thelen, D. E. Bryson, B. K. Stanford, and P. S. Beran. Multi-Fidelity Gradient-Based Optimization for High-Dimensional Aeroelastic Configurations. *Algorithms*, 15(4):131, 2022. doi:10.3390/a15040131.
- [13] D. J. Lucia, P. S. Beran, and W. A. Silva. Reduced-order modeling: new approaches for computational physics. *Progress in Aerospace Sciences*, 40(1):51–117, 2004. doi:10.1016/j.paerosci.2003.12.001.
- [14] A. Cea and R. Palacios. Parametric Reduced Order Models for the Aeroelastic Design of Flexible Vehicles. In *AIAA SCITECH 2022 Forum*, San Diego, CA, USA & Online, January 2022. doi:10.2514/6.2022-0727.
- [15] M. Sohst, J. Lobo do Vale, F. Afonso, and A. Suleman. Optimization and comparison of strut-braced and high aspect ratio wing aircraft configurations including flutter analysis with geometric non-linearities. *Aerospace Science and Technology*, 124:107531, 2022. doi:10.1016/j.ast.2022.107531.
- [16] F. Toffol and S. Ricci. Preliminary Aero-Elastic Optimization of a Twin-Aisle Long-Haul Aircraft with Increased Aspect Ratio. *Aerospace*, 10(4):374, 2023. doi:10.3390/aerospace10040374.
- [17] S. Missoum, C. Dribusch, and P. Beran. Reliability-Based Design Optimization of Nonlinear Aeroelasticity Problems. *Journal of Aircraft*, 47(3):992–998, 2010. doi:10.2514/1.46665.
- [18] B. Stanford and P. Beran. Computational strategies for reliability-based structural optimization of aeroelastic limit cycle oscillations. *Structural and Multidisciplinary Optimization*, 45:83–99, 2012. doi:10.1007/s00158-011-0663-6.
- [19] H. Shukla and M. Patil. Nonlinear state feedback control design to eliminate subcritical limit cycle oscillations in aeroelastic systems. *Structural and Multidisciplinary Optimization*, 88:1599–1614, 2017. doi:10.1007/s11071-017-3332-5.
- [20] K. Lee, D. Barton, and L. Renson. Modelling of physical systems with a hopf bifurcation using mechanistic models and machine learning. *Mechanical Systems and Signal Processing*, 191:110173, 2023. doi:10.1016/j.ymsp.2023.110173.
- [21] M. Goland. The Flutter of a Uniform Cantilever Wing. *Journal of Applied Mechanics*, 12(4):197–208, 1945. doi:10.1115/1.4009489.

- [22] A. del Carre, A. Muñoz-Simón, N. Goizueta, and R. Palacios. SHARPy: A dynamic aeroelastic simulation toolbox for very flexible aircraft and wind turbines. *Journal of Open Source Software*, 4(44):1885, 2019. doi:10.21105/joss.01885.
- [23] S. Maraniello and R. Palacios. Parametric Reduced-Order Modeling of the Unsteady Vortex-Lattice Method. *AIAA Journal*, 58(5):2206–2220, 2020. doi:10.2514/1.J058894.
- [24] H. Hesse, R. Palacios, and J. Murua. Consistent Structural Linearization in Flexible Aircraft Dynamics with Large Rigid-Body Motion. *AIAA Journal*, 52(3):528–538, 2014. doi:10.2514/1.J052316.
- [25] A. Cea and R. Palacios. Geometrically nonlinear effects on the aeroelastic response of a transport aircraft configuration. *Journal of Aircraft*, 60(1):205–220, 2023. doi:10.2514/1.C036740.
- [26] S. Maraniello and R. Palacios. State-Space Realizations and Internal Balancing in Potential-Flow Aerodynamics with Arbitrary Kinematics. *AIAA Journal*, 57(6): 2308–2321, 2019. doi:10.2514/1.J058153.
- [27] M. Baudin, M. Christopoulou, Y. Collette, J.-M. Martinez, A. D. Lee, R. Sjögren, and D. Svensson. pyDOE2: An experimental design package for python. <https://pythonhosted.org/pyDOE/#>, January 2020.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [29] S. Düssler, N. Goizueta, A. Muñoz-Simón, and R. Palacios. Modelling and numerical enhancements on a uvfm for nonlinear aeroelastic simulation. In *AIAA SCITECH 2022 Forum*, San Diego, CA & Virtual, January 2022. doi:10.2514/6.2022-2455.
- [30] F. Biscani and D. Izzo. A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5(53):2338, 2020. doi:10.21105/joss.02338.
- [31] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi:10.1109/4235.996017.



AEROELASTIC MULTIDISCIPLINARY OPTIMIZATION: ENHANCING AIRCRAFT PERFORMANCE AND SAFETY THROUGH NUMERICAL TOOLS AND EXPERIMENTAL VALIDATION

Sergio Ricci

Department of Aerospace Science and Technology
Politecnico di Milano
Milano Bovisa Campus, 20156 Milano, Italy
sergio.ricci@polimi.it, <https://www.aero.polimi.it/>

Abstract. *Aeroelastic Multidisciplinary Design Optimization (MDO) has emerged as a powerful approach to address the complex design challenges faced in the aerospace industry. The integration of numerical tools such as computational fluid dynamics (CFD) and finite element analysis (FEA), enable engineers to explore a wide range of design variables and constraints efficiently. These tools provide insights into aerodynamic performance, structural integrity, and control system stability, aiding in the identification of optimal or near-optimal designs. While numerical simulations and optimization algorithms provide valuable insights into the behavior and performance of aircraft designs, experimental validation remains crucial to confirm and refine the results but mainly to bridge the gap between simulations and real-world performance. However, this represents a challenge still today due to the complexity and cost of wind tunnel and in-flight testing, together with the safety issues. In recent years, a lot of interest arose around the possibility to perform at affordable cost low speed aeroelastic wind tunnel test. Despite specific phenomena such as the ones related to highly transonic regime cannot be investigated, the low-speed wind tunnel tests can be helpful if used like a design tool, and not just as a verification step. Aiming at this goal, it is necessary to develop a set of dedicated equipments, technologies and ad hoc implemented test strategies. In conclusion, aeroelastic multidisciplinary optimization, supported by numerical tools and experimental validation through wind tunnel testing, appears like a comprehensive framework for designing high-performance and safe aircraft. The integration of these approaches allows engineers to achieve optimal designs that balance aerodynamic performance, structural integrity, and control system stability, ultimately pushing the boundaries of aircraft design in terms of efficiency and safety. Further research in this field holds the potential to revolutionize the aviation industry by delivering advanced aircraft designs with enhanced performance characteristics, helping the zero-emission long term targets. The presentation, after a quick recap on basics concerning aeroelastic MDOs, will present a summary of ten years of aero-servo-elastic testing at large POLIMI's wind tunnel.*

Keywords: fluid-structure interaction, computational fluid dynamics, finite element analysis, optimal design, numerical simulations, wind tunnel testing



LEVEL SET TOPOLOGY OPTIMIZATION WITH AUTOMATIC DIFFERENTIATION

Andreas Neofytou^{1*}, H. Alicia Kim¹

1: Structural Engineering Department, University of California San Diego, CA 92093, USA

Abstract. *Automatic differentiation is implemented in level set topology optimization. The modularized LSTO architecture used here allows for the combination of the classical level set method with automatic differentiation. Stress minimization is used to conduct a comparative study in the computational efficiency, memory requirements and their effects on scaling of different operator overloading AD libraries. Then, the sparsity of the level set method in combination with a hybrid AD mode is employed to improve the efficiency and memory consumption of the operator overloading technique in automatic differentiation.*

Keywords: Level set topology optimization (LSTO), automatic differentiation (AD), operator overloading (OO), sparsity, hybrid implementation

1 INTRODUCTION

Topology optimization has been proven to be a useful tool for innovative designs in the aerospace industry. An essential part in the process is the computation of robust and accurate gradients of the objective and constraint functions with respect to the design variables. These design sensitivities drive the optimization procedure in order to converge to the optimized designs. Traditionally the computation of sensitivities in topology optimization is done using analytical methods such as the adjoint approach, which provide accurate sensitivities efficiently. However, the analytical derivation can be challenging and error prone as the complexity of the problem at hand increases. Cases governed by coupled multiphysics such as fluid-structure interaction problems can be classified as being challenging to differentiate analytically. Moreover, automation and interoperability become difficult since manual differentiation needs to be repeated to consider different objectives and constraints.

An alternative approach for the computation of sensitivities is automatic differentiation (AD) [1]. AD provides automated sensitivities by systematically applying the chain rule of differentiation to computer programs. The main idea is that even complex codes can be broken down into a sequence of elementary operations that are simple enough to differentiate. This automates the sensitivity computation process while the accuracy of the obtained sensitivities can be as accurate as the analytical methods.

Several works explored the use of AD in topology optimization. Laniewski and Rokicki [2] developed a topology optimization framework for a variety of multi-physics fluid flow problems. Nørgaard et al. [3] applied AD in density-based topology optimization for transient wave propagation and the lattice Boltzmann method. Chandrasekhar et al. [4] revisited AD illustrated the use of AD through the JAX library [5] in an educational paper and Dilgen et al. [6] employed AD to perform topology optimization of turbulent flows.

In this work automatic differentiation is explored in the context of level set topology optimization (LSTO). The modularized level set topology optimization used here enables the AD application in combination with the classical level set method. The operator overloading approach [7] is the main interest here due to its non-intrusive nature that requires only minor modifications to the original code. The investigations include comparing operator overloading libraries in terms of efficiency, using an example of stress minimization. The main contribution of this work is to combine the sparsity of the level set method with operator overloading to improve the efficiency and memory consumption when computing element sensitivities. One of the advantages of the level set method used here, is that sensitivities are only required within a narrow band close to the level set boundary. Thus, operator overloading is only needed to compute these boundary sensitivities rather than the sensitivities of the entire domain. The obtained results indicate that by taking advantage of this sparsity the computational time and memory consumption can be improved. We further investigate the use of a hybrid operator overloading mode.

2 MODULARIZED LSTO

An overview of the LSTO used in this work is provided in this section. Further details can be found in [8] and [9]. Using the level set field a structure can be defined as

$$\begin{aligned}
\phi(\mathbf{x}) &\geq 0, & \mathbf{x} &\in \Omega, \\
\phi(\mathbf{x}) &= 0, & \mathbf{x} &\in \Gamma, \\
\phi(\mathbf{x}) &< 0, & \mathbf{x} &\notin \Omega,
\end{aligned} \tag{1}$$

where $\phi(\mathbf{x})$ is the level set function evaluated at point \mathbf{x} , Ω is the structural domain, and Γ is the boundary. The level set advection equation [10] is solved to update the level set function at each iteration

$$\frac{d\phi}{dt} + |\nabla\phi(x)|V_n(x) = 0, \tag{2}$$

which upon discretization becomes

$$\phi_p^{k+1} = \phi_p^k - \Delta t |\nabla\phi_p^k| V_{n,i} \tag{3}$$

where p is a nodal point on the level set grid, k is the iteration number, and $\nabla\phi$ is the level set function gradient, computed using the Hamilton-Jacobi weighted essentially non-oscillatory scheme (HJ-WENO).

To obtain the velocities required to solve Eq. 2, the following linearized optimization problem is solved,

$$\begin{aligned}
\min_{V_{n,i}} \quad & \Delta F = \sum_i^{np} \left(\left(\frac{\partial F}{\partial \Omega} \right)_i A_i z_i \right) \\
\text{s.t.} \quad & G_j = \sum_i^{np} \left(\left(\frac{\partial G_j}{\partial \Omega} \right)_i A_i z_i \right) \leq g_j^t \quad j = 1, 2, \dots, N_g, \\
& -m_i \leq z_i \leq m_i
\end{aligned} \tag{4}$$

where np is the number of boundary points, $z_i = V_{n,i}\Delta t$ is the movement of the i_{th} boundary point, the pseudo time step $\Delta t = 1$, A_i is the area of the discretized boundary associated with the i_{th} boundary point, g_j^t is the target constraint value for the j_{th} constraint and m_i is the boundary movement limit. The shape sensitivities are expressed by $\frac{\partial F}{\partial \Omega}$ for the objective and $\frac{\partial G_j}{\partial \Omega}$ for the j_{th} constraint [11–14].

The LSTO method used in this work, enables the use of the discrete adjoint method and automatic differentiation. This is done by converting element sensitivities into shape sensitivities [9]. Figure 1 illustrates the overall process.

In density-based topology optimization such as SIMP, the map between optimization and the partial differential equation (PDE) solve is direct since the same analysis domain is used both for the solution of the PDE and the design representation. For LSTO this mapping is not straightforward, since the geometry is represented on a different grid than the analysis mesh. In the LSTO used here, a mapping functionality has been developed with a twofold function: First, the geometry (box 1) is mapped onto the analysis domain using the Ersatz material approximation as shown in Figs. 2 (a) and (b). Element densities are defined based on the area fractions computed for each element as the area of the element over which $\phi \geq 0$ divided by the total area of the element. Elements with ($\phi \geq 0$) have a density $\gamma = 1$, elements with ($\phi < 0$) are void with $\gamma \ll 1$ and elements that are intersected by the boundary have intermediate density values $0 \leq \gamma \leq 1$. The material properties are interpolated using the density field to solve the PDE(box 3). These

element sensitivities can be obtained using any AD method and software (box 4). The second function of the mapping, is to transfer the element sensitivities onto the boundary. This procedure is briefly described in Figs 2 (c)-(d). More details can be found in [9]. To compute the shape sensitivity for a given boundary point i , a small perturbation is assigned to it as shown in Fig. 2 (c) and (d). As a result to this perturbation, the densities of the elements near this boundary point will change due to the new position of the boundary. The boundary movement is associated approximately with the change in the density of each element m as

$$\frac{\partial \gamma_m}{\partial z_i} \approx \frac{\gamma_m - \gamma_m^\delta}{\delta} \quad (5)$$

where γ_m is the density of the m_{th} element before the perturbation, γ_m^δ is the density of the m_{th} element after the perturbation, and δ is the magnitude of the perturbation. The element based sensitivities can be converted into the boundary shape sensitivities, $(\frac{\partial F}{\partial \Omega})_i$ and $(\frac{\partial G_j}{\partial \Omega})_i$ in Eq. 4 through the following relation

$$\left(\frac{\partial F}{\partial \Omega}\right)_i = \frac{\partial F}{\partial z_i} = \sum_m^{N_e} \frac{\partial F}{\partial \gamma_m} \frac{\partial \gamma_m}{\partial z_i} \quad (6)$$

$$\left(\frac{\partial G_j}{\partial \Omega}\right)_i = \frac{\partial G_j}{\partial z_i} = \sum_m^{N_e} \frac{\partial G_j}{\partial \gamma_m} \frac{\partial \gamma_m}{\partial z_i} \quad (7)$$

where N_e is the number of elements whose densities are affected by the perturbation.

After the mapping, the shape sensitivities can be provided to the optimization algorithm (box 6) to solve the optimization problem in Eq. 4 update the level set function using Eq. 3 (box 1).

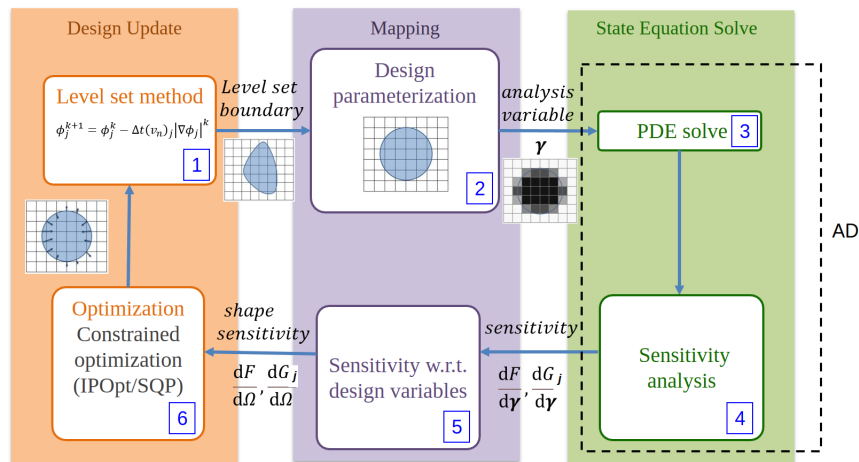


Figure 1: Modularized LSTO architecture.

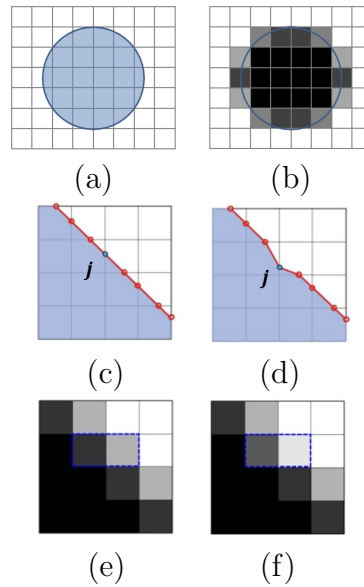


Figure 2: Perturbation scheme to compute shape sensitivities at the boundary points based on neighbouring element sensitivities.

3 OPERATOR OVERLOADING IN LSTO

The overall implementation of OO in LSTO is illustrated by the pseudo code in Algorithm 1. Figure 6 shows the computational efficiency comparison between different AD tools for different mesh sizes. The two C++ OO libraries ADOLC and CoDiPack, and the python semi-symbolic differentiation library Dolfin Adjoint are tested. The computational time for the analytical derivation using the discrete adjoint method is also provided for comparison.

For the numerical investigations the L-Beam problem shown in Fig. 3 is used considering linear elasticity. The objective of interest is p-norm stress,

$$\sigma_{pn} = \left(\int_{\Omega} \sigma_{vm}^p d\Omega \right)^{(1/p)} \quad (8)$$

where σ_{vm} is the von Mises stress. The design variable field, is used to interpolate the elemental stiffness matrix \mathbf{K}_e for the m_{th} element as

$$\mathbf{K}_{e,m} = (E_{min} + \gamma_m(E - E_{min}))\mathbf{K}_{e,0} \quad (9)$$

where E is the Young's modulus of the material, E_{min} is the Young's modulus of a void element, $(E_{min} + \gamma_m(E - E_{min}))$ is the Young's modulus of a an element cut by the boundary with intermediate density value and $K_{e,0}$ is the stiffness matrix of an element with density $\gamma = 1$.

The total system matrix is assembled as

$$\mathbf{K} = \sum_m^{N_e} \mathbf{K}_{e,m} \quad (10)$$

The discretized linear elasticity equation can then be expressed as

Algorithm 1 LSTO with OO

```

1:  $n = 0$  ▷ Iteration index
2:  $\phi$  ▷ Initialise level set function
3: while  $\Delta F \geq tolerance$  and  $n < MaxIter$  do
4:   area fractions  $\leftarrow$  LSTO boundary ▷ Discretize level set boundary
5:   Begin AD tape
6:    $\gamma_{OO}[N_e] \leftarrow Initialise$  ▷  $\gamma_{OO}$  = OO active variable vector,  $N_{BSE}$  = Number of elements
7:    $\gamma =$  area fraction ▷ Density field based on the element area fractions
8:    $i \leftarrow i + 1$ 
9:   for  $m \leftarrow 0$  to  $N_e$  do ▷ Loop through the number of elements,  $N_e$ 
10:     $\mathbf{K}_{e,m} = (E_{min} + \gamma_{OO,m}(E - E_{min}))\mathbf{K}_{e,m}$  ▷ Interpolation based on Eq. 9
11:     $\mathbf{K} \leftarrow assemble, \mathbf{K}_{e,0}$  ▷ Assemble the total stiffness matrix, Eq. 10
12:  end for
13:   $\mathbf{K}\mathbf{u} = \mathbf{F}$  ▷ Solve governing equation, Eq. 11
14:  objective  $\leftarrow \sigma_{pn}(\gamma_{OO}, \mathbf{u})$  ▷ Compute Objective
15:  End AD tape
16:   $\frac{d\sigma_{pn}}{d\gamma} \leftarrow$  Reverse mode OO ▷ Total derivative directly obtained by reverse mode
  OO
17:   $\frac{\partial \sigma_{pn}}{\partial \Omega} \leftarrow \frac{d\sigma_{pn}}{d\gamma} \frac{\partial \gamma}{\partial \mathbf{z}}$  ▷ Compute shape sensitivities using perturbation mapping (6)
18:   $V_{n,i} \leftarrow \frac{\partial \sigma_{pn}}{\partial \Omega}$  ▷ Solve the linearized optimization in Eq.4 problem for the velocities,  $V_{n,i}$ 
19:   $\phi = \phi_{new}$  ▷ Update the level set function using Eq. 3
20: end while

```

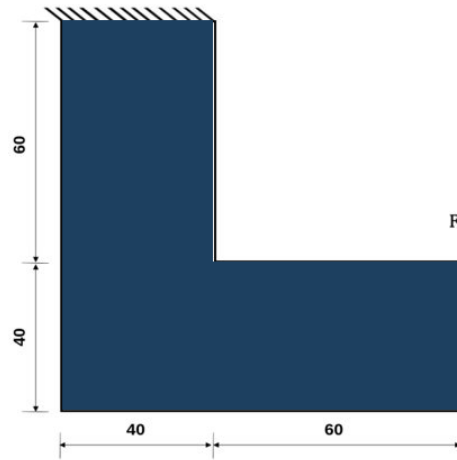


Figure 3: LBeam example

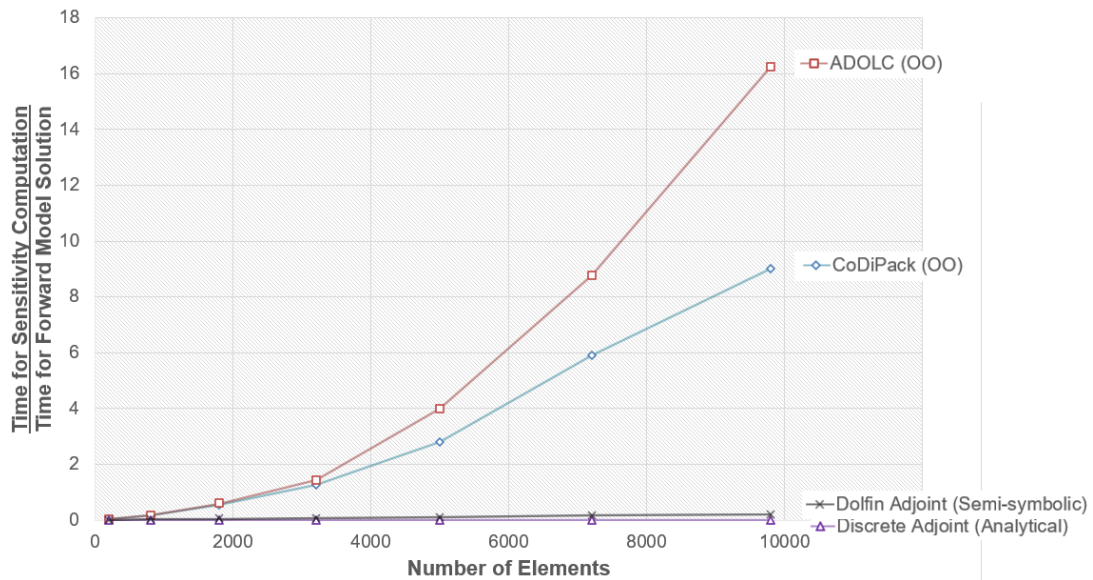


Figure 4: Comparison of computational time between different AD libraries

$$\mathbf{Ku} = \mathbf{f} \tag{11}$$

where \mathbf{u} is the displacement vector and \mathbf{f} is the load vector.

Among the AD approaches, Dolfin Adjoint is the most efficient, with a computational time close to the analytical approach. The C++ libraries have lower efficiencies as shown in Fig. 6. This is expected since the symbolic differentiation of Dolfin Adjoint operates on a mathematical level whereas the OO technique of the C++ libraries track and record all the individual operations.

3.1 OO Considering the LSTO Sparsity

In the LSTO used in this work only the sensitivities of elements near the boundary are needed to compute the boundary shape sensitivities. We utilize this sparsity to improve the solution time and memory consumption AD in topology optimization. This is shown

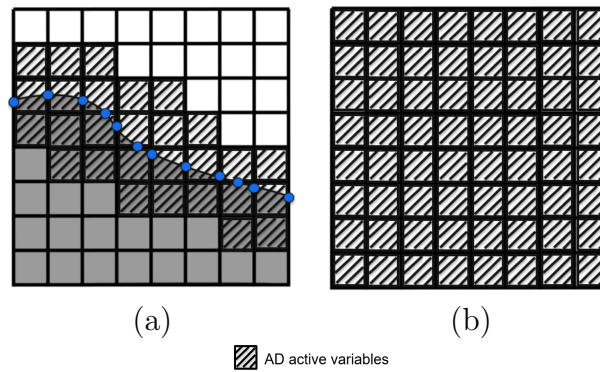


Figure 5: (a) Sparse LSTO OO implementation and (b) Full OO implementation.

in Fig. 5. Only the densities for these elements are assigned as active OO variables instead of the densities in the entire mesh. These elements are referred to as boundary sensitivity elements (BSE). This reduces the number of variables for which the OO AD library needs to track operations and differentiate.

Figure 5 illustrates a comparison of the average time per iteration between the original implementation and the sparse implementation using the CoDiPack library. As can be seen, efficiency is gained by employing the sparse implementation and this is more evident as the mesh size increases.

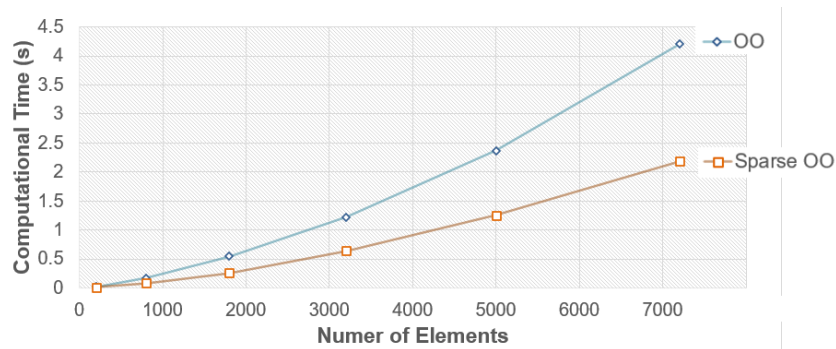


Figure 6: Computational time comparison between sparse OO implementation and original OO using Codipack

3.2 LSTO With Hybrid AD

The hybrid AD mode provides significant gains in memory consumption [18]. Here, the sparsity of the level set method is exploited to make this approach even more efficient.

The main concept is that rather than using AD to obtain the total derivative of the objective function with respect to the design variables, the total expression is analytically derived and the AD is used only to obtain the partials. In the case of the p-norm objective for example, the total derivative for the p-norm stress objective can be expressed as

$$\frac{d\sigma_{pn}}{d\gamma} = \frac{\partial\sigma_{pn}}{\partial\gamma} + \frac{\partial\sigma_{pn}}{\partial\mathbf{u}} \frac{\partial\mathbf{u}}{\partial\gamma} \quad (12)$$

The unknown implicit term, $\frac{\partial\mathbf{u}}{\partial\gamma}$, can be obtained by the total derivative of the residual vector, $\mathbf{R}(\mathbf{u}(\gamma), \gamma) = \mathbf{K}\mathbf{u} - \mathbf{F} = 0$, with respect to the design variable vector, γ

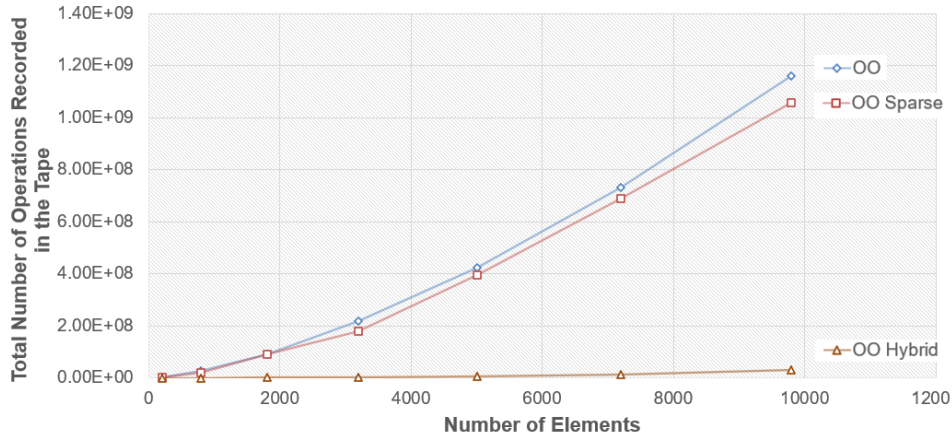


Figure 7: Total number of operations recorded in ADOLC tape for OO, Sparse OO and Hybrid OO

$$\frac{d\mathbf{R}}{d\gamma} = \frac{\partial\mathbf{R}}{\partial\gamma} + \frac{\partial\mathbf{R}}{\partial\mathbf{u}} \frac{\partial\mathbf{u}}{\partial\gamma} = \mathbf{0} \Rightarrow \frac{\partial\mathbf{u}}{\partial\gamma} = -\frac{\partial\mathbf{R}}{\partial\gamma} \left[\frac{\partial\mathbf{R}}{\partial\mathbf{u}} \right]^{-1} \quad (13)$$

Replacing Eq. 13 into Eq. 12 the total objective sensitivity can be obtained as

$$\frac{d\sigma_{pn}}{d\gamma} = \frac{\partial\sigma_{pn}}{\partial\gamma} - \boldsymbol{\lambda}^T \frac{\partial\mathbf{R}}{\partial\gamma} \quad (14)$$

in which $\boldsymbol{\lambda}$ is the adjoint variable vector computed by solving the the adjoint linear system

$$\frac{\partial\mathbf{R}}{\partial\mathbf{u}} \boldsymbol{\lambda} = \frac{\partial\sigma_{pn}}{\partial\mathbf{u}} \quad (15)$$

The idea then is that AD can be used to compute the Jacobian matrix given by the partials $\frac{\partial\mathbf{R}}{\partial\gamma}$ and $\frac{\partial\mathbf{R}}{\partial\mathbf{u}}$ and the vectors $\frac{\partial\sigma_{pn}}{\partial\gamma}$ and $\frac{\partial\sigma_{pn}}{\partial\mathbf{u}}$. Once the partials are obtained, Eqs. 14 and 15 can be solved manually to obtain the total sensitivity vector $\frac{d\sigma_{pn}}{d\gamma}$.

In Fig. 7, the number of operations stored in the ADOLC tape for the original OO implementation, the sparse OO and the hybrid OO are given. As can be seen, there is a significant reduction in the storage of operations when the hybrid mode is used.

Fig. 8 shows the computational time spent by all the different OO implementations using CoDiPack. The computational time for the analytical derivation using the discrete adjoint method is also provided. We note that even within the same OO library, the implementation algorithms can lead to different levels of efficiency. As expected, the analytical derivation is the most efficient. The highest computational time is spent by the hybrid mode implementation. Although the hybrid mode results in big memory savings as indicated by Fig. 7, due to the more involved manual implementation for obtaining the partials the efficiency is lost. However, when the hybrid mode is combined with the sparsity of LSTO, the resulting implementation is the most efficient after the analytical differentiation.

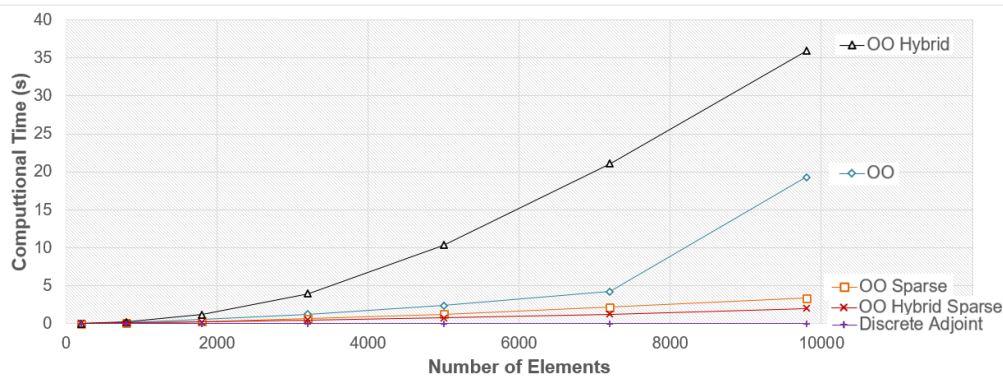


Figure 8: Computational time comparison of all the different OO implementations using Codipack

4 CONCLUSION

In this paper the use of AD for LSTO is investigated. The sparsity of LSTO can be utilized to improve the memory consumption and solution time of OO. The combination of this sparsity with the hybrid OO results in the most efficient and memory effective implementation. Future directions of the current research, include the application of these implementations in multiphysics topology optimization problems.

REFERENCES

- [1] A. Griewank. Evaluating derivatives. *SIAM: Philadelphia*, 2000.
- [2] L. Laniewski-Wołk and J. Rokicki. Adjoint lattice boltzmann for topology optimization on multi-gpu architecture. *Computers and Mathematics with Applications*, 71: 833–848, 2016.
- [3] S. A. Norgaard, M. Sagebaum, N. R. Gauger, and B. S. Lazarov. Applications of automatic differentiation in topology optimization. *Structural and Multidisciplinary Optimization*, 56:1135–1146, 2017.
- [4] A. Chandrasekhar, S. Sridhara, and K. Suresh. Auto: a framework for automatic differentiation in topology optimization. *Structural and Multidisciplinary Optimization*, 64:4355–4365, 2021.
- [5] J. Bradbury, R. Frostig, P. Hawkins, M. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. Jax: composable transformations of python+numpy programs. URL <http://github.com/google/jax>, 2018.
- [6] C. B. Dilgen, S. B. Dilgen, D. R. Fuhrman, O. Sigmund, and B. S. Lazarov. Topology optimization of turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 331:363–393, 2018.
- [7] J. D. Pryce and J. K. Reid. Ad01, a fortran 90 code for automatic differentiation. Report RAL-TR-1998-057, Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, OX11 0QX, U.K, 1998.

- [8] C. M. Jauregui, J. Hyun, A. Neofytou, J. S. Gray, and H. A. Kim. Avoiding reinventing the wheel: reusable open-source topology optimization software. *Structural and Multidisciplinary Optimization*, 66(145), 2023.
- [9] S. Kambampati, H. Chung, and H. A. Kim. A discrete adjoint based level set topology optimization method for stress constraints. *Computer Methods in Applied Mechanics and Engineering*, 377:113563, 2020.
- [10] S. Osher, R. Fedkiw, and K. Piechor. Level set methods and dynamic implicit surfaces. *Applied Mechanics Reviews*, 57(3):B15, 2004.
- [11] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, 2004.
- [12] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1-2): 227–246, 2003.
- [13] P. D. Dunning and H. A. Kim. Introducing the sequential linear programming level-set method for topology optimization. *Structural and Multidisciplinary Optimization*, 51(631-643), 2015.
- [14] R. Picelli, S. Townsend, C. Brampton, J. Norato, and H. Kim. Stress-based shape and topology optimization with the level set method. *Computer Methods in Applied Mechanics and Engineering*, 329:1–23, 2018.
- [15] G. Allaire, C. Dapogny, and P. Frey. Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, 282: 22–53, 2014.
- [16] N. Jenkins and K. Maute. An immersed boundary approach for shape and topology optimization of stationary fluid-structure interaction problems. *Structural and Multidisciplinary Optimization*, 54:1191–1208, 2016.
- [17] S. Cai, W. Zhang, J. Zhu, and T. Gao. Stress constrained shape and topology optimization with fixed mesh: A b-spline finite cell method combined with level set function. *Computer Methods in Applied Mechanics and Engineering*, 278:361–387, 2014.
- [18] C. A. Mader, J. R. R. A. Martins, J. J. Alonso, and E. van der Weide. Adjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4), 2008.



DEVELOPMENT OF A FE CODE FOR ADJOINT-BASED COUPLED AEROSTRUCTURAL OPTIMISATION

Luca Scalia*, Rauno Cavallaro and Andrea Cini

Department of Aerospace Engineering
Universidad Carlos III de Madrid
lscalía@pa.uc3m.es, https://aero.uc3m.es/asd_lab-home/

Abstract. *With the current environmental challenges, there is a relevant pressure on the aeronautical sector towards more sustainable aircraft. As a consequence, both new design methodologies and unconventional aircraft layouts are under investigation. Optimisation considering the strong coupling between different disciplines is becoming a paradigm to support new efficient designs. In this paper, an existing framework to carry out high-fidelity aerodynamic/structural optimisation is enhanced with the development of a structural solver. The framework is modular, and uses algorithmic differentiation to provide discrete-exact adjoints. The structural solver is being developed to tackle large-scale problems, providing responses that can be used as objectives and/or constraints in the optimisation, and maintaining the framework requirements on modularity and sensitivity evaluation. Preliminary capabilities of the structural solver in a purely structural optimisation, or within an aerostructural optimisation are proved.*

Keywords: Aerostructural optimisation, adjoint method, algorithmic differentiation, finite element method, aerospace, new generation wings, highly flexible wings

1 INTRODUCTION

1.1 Sustainable aviation challenge

The current leitmotiv in aviation is reducing carbon footprint. As claimed at the 77th Annual General Meeting of The International Air Transport Association (IATA) held in Boston 2021 [1], the objective is to achieve net-zero carbon emissions by 2050. This resolution is pushing aircraft manufacturers towards devising new ways to make aviation more sustainable, acting at different levels: (i) viable and sustainable aviation fuels, (ii) reduction of inefficiencies in air traffic management and airspace infrastructure, (iii) design of aerodynamically and structurally more efficient airframes, (iv) novel powertrains. Two concepts are shown in Fig.1, depicting high aspect-ratio wings (a) and an electric aircraft featuring distributed propulsion (b). The Transonic Truss-Braced Wing (TTBW) from Boeing (Fig1.1a), possesses a thin and high aspect ratio wing stabilized by a lower diagonal struts. According to the wind tunnel test results, it could consume up to 10% less jet fuel [2]. In January 2023, NASA selected the Boeing TTBW for its Sustainable Flight Demonstrator project [3]. Some recently awarded research projects are even considering



(a) Boeing TTBW rendering [4].

(b) Airbus ZEROe concept.

Figure 1: Two examples of new aircraft concepts.

the synergies between such new technologies [5].

Design of high aspect-ratio wings, or similarly, novel airframes poses exceptional challenges. The need of correctly modelling physic phenomena without legacy experience leads towards higher-fidelity and highly coupled challenging approaches. The strong aerostructural (aerodynamic-structural) coupling is one of them.

For these reasons, significant effort has been dedicated in the last decades to the development of high-fidelity digital means and approaches supporting the aerostructural coupled design of novel airframes.

1.2 State of the art on aerostructural optimisation

The first publications on aerostructural optimisation dates back to the seventies: due to the lack of computational power Haftka [6] used a lower-fidelity aerodynamic solver (based on the Lifting Line Method). Only within the last 25 years the available computational power allowed high-fidelity solvers within gradient-based optimisation featuring a large number of design variables. The work of Maute et al. [7] investigated the best formulation for cases featuring large displacements, confirming that the three-field formu-

lations (modelling of structural, aerodynamic and mesh domains) are more robust than the two-fields ones, although at the expense of a 25% average increase in computational performance. The authors resorted to a direct method for computing the sensitivities and an Euler model for the aerodynamic simulation. The work of Martins et al. [8] is one of the first attempts to efficiently use the adjoint method in a multidisciplinary optimisation, guaranteeing a relatively independence of the computational time from the number of design variables. Barcelos and Maute [9] increased the fidelity of the analysis by accounting for turbulence with the Spalart-Allmaras model [10]. The application was on the Nasa's Aeroelastic Research Wing (ARW2) [11], whose shape and stringer's thickness were changed to maximise the lift-over-drag-ratio or minimise the drag. Derivatives were computed through a direct method.

Last decade has witnessed a definitive shift toward the adjoint method for sensitivity evaluation. Kenway et al.[12] presented the coupling between the aerodynamic solver *SUmb* [13] (employing automatic differentiation) and the linear structural solver *TACS* [14] (that integrates its own adjoint solver). A Newton-Krylov methods was selected for the solution of the aerostructural systems and the related adjoint problem. The scalability of the approach was proved on a testcase with around 16 millions cells for CFD and around 1 million structural degrees of freedom. The work of Wang et al. [15] is focused on rotorcraft; the discipline solvers were FUN3D [16] (implementing its discrete consistent adjoint-based method) and DYMORE5 [17] (employing complex variable method for structural sensitivities with respect to the aerodynamic loads). To conclude, Sanchez et al. [18] describes a method for coupled sensitivities evaluation based on fixed-point iterations on the adjoint variables assisted by algorithmic differentiation. The approach removes the need for an accurate construction of the Jacobian and poses no limits on the complexity of the primal problem. One drawback was the monolithic architecture of the platform.

1.3 Objective of this work

The starting point of this research is the work by Bombardieri et al [19], presenting a framework for high-fidelity gradient-based aerostructural optimisation supported by automatic differentiation. The procedure is based on the same fixed-point iteration strategy for coupled sensitivity computation as defined in [18]. The adjoint problem was modified to increase the decoupling between the different modules of the framework. The solvers exchanged information through a *python* wrapper. The resulting framework is highly flexible and modular, allowing the introduction of new features, as new types of design variables or replacing a solver, without much effort. The solver *SU2* is used for the CFD analysis while the structural part of the problem is solved by *pyBeam*. Both use the C++ library *CoDiPack* [20] for gradient evaluation in computer programs. The algorithm selected for the gradient-based optimisation is the Sequential Least Square Quadratic Programming (SLSQP) [21] available with the *SciPy* library [22]. The in-house solver *pyBeam* was written to demonstrate the viability of the formulation, therefore it lacks the flexibility and performance required for more realistic high-fidelity structural models.

The main aim of this work is extending the capability of the above-mentioned framework to enable aerostructural optimisation concurrently considering wing planform- and wingbox-levels design variables and responses. A second aim of this research is allowing the framework to solve industrial problem. That entails the need to implement advanced

techniques to increase computational performances (e.g., parallelization of the structural solver both in the primal and adjoint phases).

The paper is structured as follows. Section 2 reports an overview of the technical background and the current framework, with focus on the updated mathematical formulation. Section 3 offers a description of the FEM solver and its development, as well as a preliminary verification of both the primal and adjoint solvers and a purely structural optimisation. Section 4 shows a preliminary aerostructural optimisation application of the current state of the software.

2 TECHNICAL BACKGROUND

This section presents the technical background behind the formulation of the aerostructural optimisation problem. First, the equations that govern the coupled aerodynamic and structural behaviour of the system are reviewed, with emphasis on the iterative solution and specialization to the problem at hand. The focus is then shifted to the adjoint problem for evaluation of the gradients; the formulation is fine-tuned to be synergistic with capabilities offered by the employed algorithmic differentiation (AD) libraries.

2.1 The aerostructural primal problem

The modeling of the aerostructural problem is based on the so-called 3-fields formulation [9, 23], and namely the fluid, the solid and the mesh field. Such a formulation is considered a suitable method [24] for problems interested by significant structural displacements.

2.1.1 The aerodynamic problem

The model of interest is the transonic fluid flow governed by the Navier-Stokes equations. The equations are formulated with the Arbitrary Lagrangian Eulerian (ALE) [25, 26] approach to deal with moving mesh boundary. The expression for the fluid flow residual \mathcal{F} is as follows

$$\mathcal{F}(\mathbf{w}, \mathbf{z}) = \frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \Psi^c(\mathbf{w}, \mathbf{z}) - \nabla \cdot \Psi^v(\mathbf{w}, \mathbf{z}) - \mathbf{Q}(\mathbf{w}) = \mathbf{0} \quad (1)$$

where $\mathbf{w} = [\rho, \rho \mathbf{v}, \rho E]$ and \mathbf{z} are the vector of state variables and the fluid mesh grid point positions respectively, being the former dependent on fluid density, flow velocity and total energy per unit of mass (namely ρ , \mathbf{v} and E). The terms Ψ^c , Ψ^v and $\mathbf{Q}(\mathbf{w})$ are the convective and viscous fluxes and a generic source term.

The solver of the aerodynamic field is the open-source multi-physics software *SU2* [27–29]. The code is differentiated by the AD tool *CoDiPack* [20] in reverse mode, hence, it is an ideal platform to perform adjoint-based gradient evaluation. Moreover, a top-level interface in *python* is provided that promotes modularity and integration into multi-code optimisation applications.

2.1.2 The structural problem

The structural problem is formulated under the hypothesis of large displacements and small strains

$$\mathcal{S}(\mathbf{u}_s, \mathbf{f}_s) = \mathbf{f}_{int}(\mathbf{u}_s) - \mathbf{f}_s = \mathbf{0} \quad (2)$$

where \mathcal{S} , \mathbf{f}_{int} , \mathbf{f}_s and \mathbf{u}_s are the residual, the internal forces, the applied external loads and the nodal displacements respectively. The structural problem is solved by the in-house finite element solver *AUGUSTO++*. The software has been wrapped at the top-level with a python interface, guaranteeing a straightforward coupling with SU2 (or other software) for multiphysics applications. *AUGUSTO++* is also differentiated by CodiPack, enabling the evaluation of sensitivities. The solver has been developed using classic parallelization techniques to tackle large-dimension problems.

Only the case of linearly behaving structures is considered in this work, i.e.,

$$\mathbf{f}_{int} = \mathbf{K}\mathbf{u}_s \quad (3)$$

where the global stiffness matrix, \mathbf{K} , depends on the structural design variables $\boldsymbol{\alpha}_s$.

2.1.3 The fluid mesh deformation problem

Structural displacements lead to deformations of the fluid domain. Such an effect is modelled considering the mesh behaving as a linear pseudo-elastic medium [30], i.e.,

$$\mathcal{M}(\mathbf{z}, \mathbf{u}_f) = \mathbf{K}^m \mathbf{z} - \tilde{\mathbf{f}}(\mathbf{u}_f) = \mathbf{0} \quad (4)$$

where \mathbf{K}^m is a fictitious stiffness matrix and $\tilde{\mathbf{f}}$ are fictitious loads applied to the mesh that must be compatible with the displacements on the fluid boundaries \mathbf{u}_f . The mesh deformation is also handled by *SU2* solver.

2.1.4 The interface operator

Loads and displacement are transferred between the fluid and solid mesh boundaries by means of a spline matrix \mathbf{H}_{MLS} , defined through a Moving Least Square (MLS) algorithm [31]. The relationship between the displacements and loads, defined on the structural mesh (\mathbf{u}_s , \mathbf{f}_s) and the same quantities defined on the aerodynamic mesh boundary (\mathbf{u}_f , \mathbf{f}_f) can be stated as

$$\mathbf{u}_f = \mathbf{H}_{MLS} \mathbf{u}_s \quad (5a)$$

$$\mathbf{f}_s = \mathbf{H}_{MLS}^T \mathbf{f}_f \quad (5b)$$

The spline matrix \mathbf{H}_{MLS} is a function of the position of the structural nodes, \mathbf{x}_s , and of the fluid nodes on the moving boundary, \mathbf{x}_f .

The in-house spline module is coded in C++ and relies on the ANN library [32]. As for the other solvers, the spline module comes with a top-level *python* interface.

2.1.5 Solution of the aerostructural system

The Fluid Structure Interaction (FSI) problem, specialized for linear structures, is stated as follows:

$$\mathbf{F}(\mathbf{w}, \mathbf{z}) - \mathbf{w} = \mathbf{0} \quad (6a)$$

$$\mathbf{F}_f(\mathbf{w}, \mathbf{z}) - \mathbf{f}_f = \mathbf{0} \quad (6b)$$

$$\mathbf{M}(\mathbf{u}_{tot}) - \mathbf{z} = \mathbf{0} \quad (6c)$$

$$\mathbf{H}_{MLS}^T \mathbf{f}_f - \mathbf{f}_s = \mathbf{0} \quad (6d)$$

$$\mathbf{K}(\boldsymbol{\alpha}_s) \mathbf{u}_s - \mathbf{f}_s = \mathbf{0} \quad (6e)$$

$$\mathbf{H}_{MLS} \mathbf{u}_s - \mathbf{u}_f = \mathbf{0} \quad (6f)$$

$$\mathbf{u}_{tot} - \mathbf{u}_f - \mathbf{u}_{F_\alpha} = \mathbf{0} \quad (6g)$$

The system of equations will be briefly referred to with $\mathcal{G} = \mathbf{0}$. Notice that for the fluid domain the fixed-point operator, \mathbf{F} , has been used instead of the residual form one, \mathcal{F} . The reasons behind this choice are summarized in [33]. For the fluid mesh domain a pseudo fixed-point form is employed. The system is closed by the two interface equations, Eq.(6d) and Eq.(6f) while Eq.(6g) states that the total displacement of the wing (cumulative displacement) $\mathbf{u}_{F_{tot}}$ is given by the deflection during loads application, \mathbf{u}_f , and the redesign during the optimisation process of the jig-shape, \mathbf{u}_{F_α} . The state variables are gathered into the \mathbf{y} vector, as described in Tab.1

Table 1: State variable vector \mathbf{y}

symbol	description
\mathbf{u}_s	displacements defined on the structural mesh
\mathbf{u}_f	displacements defined on the fluid mesh boundary
\mathbf{w}	aerodynamic state variables
\mathbf{z}	fluid mesh nodes displacements
\mathbf{f}_s	loads defined on the structural mesh
\mathbf{f}_f	loads defined on the fluid mesh boundary
\mathbf{u}_{tot}	cumulative displacement of the wing surface

The solution strategy is based on a Block Gauss-Seidel (BGS) approach. Nonlinearity of the governing equations (in the current case, only aerodynamic ones) requires inner iterations. Thanks to the modularity of the solvers, it is straightforward to set-up an orchestrator calling the solvers and interchanging relevant data according to the most appropriate solution strategy. The algorithm for solving the FSI problem is described in Alg.(1). The output of the FSI problem is the primal solution $\mathbf{y}^* = [\mathbf{w}^*, \mathbf{z}^*, \mathbf{u}_{tot}^*, \mathbf{u}_f^*, \mathbf{u}_s^*, \mathbf{f}_s^*, \mathbf{f}_f^*]^T$.

Algorithm 1 The aerostructural primal solver

```

Initialise  $[\mathbf{w} \ \mathbf{z} \ \mathbf{u}_s \ \mathbf{u}_f \ \mathbf{f}_s \ \mathbf{f}_f] = \mathbf{0}$ 
while  $N \leq N_{FSI}$  do
  transfer displacements to fluid mesh  $\mathbf{u}_s \rightarrow \mathbf{u}_f$ 
  run mesh deformation solver  $\mathbf{u}_f \rightarrow \mathbf{z}$ 
  run CFD solver  $\mathbf{z} \rightarrow \mathbf{w}, \mathbf{f}_f$ 
  transfer loads on structural mesh  $\mathbf{f}_f \rightarrow \mathbf{f}_s$ 
  run structural solver  $\mathbf{f}_s \rightarrow \mathbf{u}_s$ 
end while

```

2.2 The aerostructural optimisation problem

The aerostructural optimisation problem can be written as

$$\begin{aligned}
& \text{Minimize} && J(\mathbf{y}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) \\
& \text{w.r.t.} && \boldsymbol{\alpha} \\
& \text{subject to} && \mathcal{G}(\mathbf{y}(\boldsymbol{\alpha}), \boldsymbol{\alpha}) = 0
\end{aligned} \tag{7}$$

Where J is the objective function (e.g., the drag coefficient or the weight of the structure) and \mathcal{G} represents the residual of the FSI problem. The objective function J is not specified for the moment and its dependence on the state variables \mathbf{y} (listed in Tab.1) and design variables $\boldsymbol{\alpha}$ is left general.

2.2.1 The design variables

The aerostructural optimisation problem features both structural ($\boldsymbol{\alpha}_s$) and aerodynamic ($\boldsymbol{\alpha}_a$) design variables. Typical structural design variables are the wing components and subelement thicknesses or parameters to describe the stacking sequence of composite plies. The aerodynamic design variables are used to describe the wing jig-shape and are parameterised. A Free Form Deformation (FFD) technique [34] is employed: the position of the aerodynamic grid points ($\mathbf{x}_{F\alpha}$) of the wing surface are interpolated through Bezier's basis functions with reduced number of control points according to the following formula

$$\mathbf{x}_{F\alpha} = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n N_i(\mu) N_j(\nu) N_k(\xi) \mathbf{x}_{ijk}^{CP} \tag{8}$$

where \mathbf{x}_{ijk}^{CP} are the control points coordinates, identified by indexes i , j and k , whereas the functions N are Bernstein Polynomials that depend on the parametric coordinates μ , ν and ξ . These coordinates are evaluated with a point-inversion procedure as described in reference [35]. As a consequence, the control point positions become the actual aerodynamic design variables of the problem $\boldsymbol{\alpha}_a$. The chain rule allows to easily express the relationship between the derivatives of the objective function with respect to the aerodynamic grid points displacement and the control point displacements:

$$\frac{dJ}{d\mathbf{u}_{ijk}^{CP}} = \frac{dJ}{d\mathbf{u}_{F\alpha}} \frac{d\mathbf{u}_{F\alpha}}{d\mathbf{u}_{ijk}^{CP}} \tag{9}$$

A parameteric deescription may be introduced also for the structural design variables, e.g., by imposing a certain variation of the panels thickness along the wingspan.

2.3 The aerostructural adjoint problem

The method for computing the coupled aerostructural gradients closely follows the one proposed in reference [19]. In this preliminary work, small modifications have been done to account for the case of linear structures.

First, the Lagrangian function is defined,

$$\begin{aligned} \mathcal{L}(\mathbf{w}, \bar{\mathbf{w}}, \mathbf{z}, \bar{\mathbf{z}}, \mathbf{u}_s, \bar{\mathbf{u}}_s, \mathbf{u}_f, \bar{\mathbf{u}}_f, \mathbf{f}_s, \bar{\mathbf{f}}_s, \mathbf{f}_f, \bar{\mathbf{f}}_f, \mathbf{u}_{tot}, \bar{\mathbf{u}}_{tot}, \mathbf{u}_{F\alpha}, \boldsymbol{\alpha}_s) = \\ J + \bar{\mathbf{w}}^T [\mathbf{F}(\mathbf{w}, \mathbf{z}) - \mathbf{w}] + \bar{\mathbf{z}}^T [\mathbf{M}(\mathbf{u}_{tot}) - \mathbf{z}] + \bar{\mathbf{u}}_{tot}^T [\mathbf{u}_{tot} - \mathbf{u}_f - \mathbf{u}_{F\alpha}] \\ + \bar{\mathbf{u}}_f^T [\mathbf{H}_{MLS} \mathbf{u}_s - \mathbf{u}_f] + \bar{\mathbf{u}}_s^T [\mathbf{K}(\boldsymbol{\alpha}_s) \mathbf{u}_s - \mathbf{f}_s] + \bar{\mathbf{f}}_s^T [\mathbf{H}_{MLS}^T \mathbf{f}_f - \mathbf{f}_s] \\ + \bar{\mathbf{f}}_f^T [\mathbf{F}_f(\mathbf{w}, \mathbf{z}) - \mathbf{f}_f] \end{aligned} \quad (10)$$

where the adjoint variables related to the homologous state variables $\bar{\mathbf{w}}$, $\bar{\mathbf{z}}$, $\bar{\mathbf{u}}_{tot}$, $\bar{\mathbf{u}}_f$, $\bar{\mathbf{u}}_s$, $\bar{\mathbf{f}}_s$ and $\bar{\mathbf{f}}_f$ have the role of Lagrange multipliers. The adjoint state variables are gathered into the vector $\bar{\mathbf{y}}^T$.

The adjoint equations of the problem in Eq.(7) are obtained after differentiation with respect to the state variables, see Eq.(11a) to Eq.(11g). Even though the adjoint equations are linear about the primal solution \mathbf{y}^* , the adjoint to the flow solution equation can be written in iterative form (for iteration n), Eq.(11a). Among the reasons behind this choice, there is a more straightforward integration with the AD software.

$$\left. \frac{\partial J}{\partial \mathbf{w}} \right|_* + \bar{\mathbf{w}}_n^T \left. \frac{\partial \mathbf{F}}{\partial \mathbf{w}} \right|_* + \bar{\mathbf{f}}_f^T \left. \frac{\partial \mathbf{F}_f}{\partial \mathbf{w}} \right|_* = \bar{\mathbf{w}}_{n+1}^T \quad (11a)$$

$$\left. \frac{\partial J}{\partial \mathbf{z}} \right|_* + \bar{\mathbf{w}}^T \left. \frac{\partial \mathbf{F}}{\partial \mathbf{z}} \right|_* + \bar{\mathbf{f}}_f^T \left. \frac{\partial \mathbf{F}_f}{\partial \mathbf{z}} \right|_* + \bar{\mathbf{z}}^T = \mathbf{0} \quad (11b)$$

$$\left. \frac{\partial J}{\partial \mathbf{u}_{tot}} \right|_* + \bar{\mathbf{z}}^T \left. \frac{\partial \mathbf{M}}{\partial \mathbf{u}_{tot}} \right|_* + \bar{\mathbf{u}}_{tot}^T = \mathbf{0} \quad (11c)$$

$$\left. \frac{\partial J}{\partial \mathbf{u}_f} \right|_* - \bar{\mathbf{u}}_f^T - \bar{\mathbf{u}}_{tot}^T = \mathbf{0} \quad (11d)$$

$$\mathbf{K}^T \bar{\mathbf{u}}_s = - \left. \frac{\partial J}{\partial \mathbf{u}_s} \right|_* - \mathbf{H}_{MLS}^T \bar{\mathbf{u}}_f \quad (11e)$$

$$\left. \frac{\partial J}{\partial \mathbf{f}_s} \right|_* - \bar{\mathbf{u}}_s^T - \bar{\mathbf{f}}_s^T = \mathbf{0} \quad (11f)$$

$$\left. \frac{\partial J}{\partial \mathbf{f}_f} \right|_* - \bar{\mathbf{f}}_f^T + \bar{\mathbf{f}}_s^T \mathbf{H}_{MLS}^T = \mathbf{0} \quad (11g)$$

Finally, differentiation with respect to the design variables produces the sensitivities equations

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}_{F\alpha}} = \left. \frac{dJ}{d\mathbf{u}_{F\alpha}} \right|_* = \left. \frac{\partial J}{\partial \mathbf{u}_{F\alpha}} \right|_* - \bar{\mathbf{u}}_{tot}^T = \mathbf{0} \quad (12a)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_s} = \left. \frac{dJ}{d\boldsymbol{\alpha}_s} \right|_* = \left. \frac{\partial J}{\partial \boldsymbol{\alpha}_s} \right|_* + \bar{\mathbf{u}}_s^T \left. \frac{\partial \mathcal{S}}{\partial \boldsymbol{\alpha}_s} \right|_* = \mathbf{0} \quad (12b)$$

2.3.1 Solution of the adjoint problem

The adjoint variables are computed by iteratively solving the adjoint system in Eq.(11) from top to bottom. Each set of equations from Eq.(11) to Eq.(11g) receives as input a quantity from the previous set and provides input to the next block to be solved.

The first three equations are handled by *SU2* modules. In particular, the very first one, as a consequence of the non linearity of the aerodynamic problem, is solved in a fixed-point iteration fashion (for a certain number n of subiterations) and takes as input the adjoint variables $\bar{\mathbf{f}}_f$ from the previous adjoint FSI iteration.

The terms in the form $\bar{\mathbf{y}}^T \frac{\partial \mathbf{A}}{\partial \mathbf{x}} \Big|_*$ are computed using the AD library *CoDiPack*. Each module applies AD independently and exchanges information with the other modules via high-level functions. From the implementation perspective, after the solution of the primal problem, the AD software *CoDiPack* is initialized and one extra FSI iteration is performed to allow for the registration of the computational path, needed to evaluate the terms $\bar{\mathbf{y}}^T \frac{\partial \mathbf{A}}{\partial \mathbf{x}} \Big|_*$. The adjoint variables are then used to compute the sensitivities, with respect to the aerodynamic and structural design variables, as shown in Eq.(12). Specialization by chosen design variables and objective functions can lead to simplification of some of the equations.

3 DEVELOPMENT OF THE FEM MODULE

This section provides an overview of the in-house finite element code *AUGUSTO++* which is responsible for providing the solution of the primal and adjoint structural problems within the aerostructural optimisation.

3.1 Overview on the code

The finite element code *AUGUSTO++* is written in C++ according to the *Object-Oriented Programming* paradigm and features parallel computation capability via the *MPI* library [36]. The code is able to model the response of highly-flexible structures assuming small strains and elastic materials. The problem is solved considering an Updated Lagrangian approach. For large displacement fields, the strains are evaluating considering a co-rotational approach. A library of different finite elements is implemented, among which:

1. *4-nodes bilinear isoparametric shell element* with *Selective Reduced Integration*. It is based on *Nastran* CQUAD4 element [37].
2. *3-nodes CST and DKT element* based on [38].
3. *2-nodes linear isoparametric bar element* with constant cross section, *Selective Reduced Integration* and possibility to offset the section from the nodes line. It is based on *Nastran* CBAR formulation [39].
4. *Euler-Bernoulli beam element* with Hermite interpolation [38].
5. *rigid element RBE2* for modelling of stiff connections.

In this work, only linear static analysis are considered. The implementation of nonlinear capabilities to some of the FEs is currently under development.

3.2 Parallelisation strategy

As mention in section 2.1.2, the discretised linear finite element problem equilibrium equation is described as below:

$$\mathbf{K}\mathbf{u}_s = \mathbf{f}_s \quad (13)$$

The parallelisation of the code consists in distributing the mesh (finite elements and nodes) among different processes allowing for a parallel evaluation of the contributions to the global stiffness matrix. The same holds for the RHS.

At the implementation level, each process owns a portion of the overall number of finite elements and relative nodes. There are other two circumstances where *MPI* collective communication between processes is necessary:

1. *Application of external loads*: if a nodal force is applied on nodes owned by more than one process, only the process of lowest rank (among the ones owing that node) contains the load value. This check is clearly possible through a collective communication.
2. *Checks on RBE2 and RBE3 elements*: a series of checks are carried out to detect inconsistent situations, e.g., a node cannot be slave and master at the same time as a consequence of the presence of a circular pattern. These checks require collective communication between processes.

The distributed global stiffness matrix and external loads are fed into an instance of the linear system solver *MUMPS* [40] (which also relies on *MPI* and is instantiated on each working process) that handles the solution in parallel of the linear system. After solving the linear system, the entire solution vector \mathbf{u}_s is made available to all the processes via a broadcast operation.

3.3 Wrapping of the code with *SWIG*

To use *AUGUSTO++* capabilities within a *python* environment, the C++ code is wrapped with *SWIG* [41]. For this purpose, an additional higher-level class was implemented in the core code containing high-level member functions that execute the main operations of the entire analysis, such as initialisation of element/node/material/property objects, solution of linear system, extraction of a desired structural response, solution of the adjoint problem. Such functions are called by the orchestrator as well as by member functions belonging to other higher-level objects which communicate with the other wrapped modules of the framework, such as the swigged *SU2* and spline modules, or the functions of the numerical optimiser library, i.e. *SciPy* [22].

3.4 The adjoint of the structural solver

As far as the pure structural problem is concerned, the adjoint Eq.(11e) and the gradient Eq.(12b) can be considered, discarding the the term $\bar{\mathbf{u}}_f^T \mathbf{H}_{MLS}$ coming from the aerodynamic coupling:

$$\mathbf{K}^T \bar{\mathbf{u}}_s = - \left. \frac{\partial J}{\partial \mathbf{u}_s} \right|_* \quad (14a)$$

$$\frac{dJ}{d\boldsymbol{\alpha}_s} = \left. \frac{\partial J}{\partial \boldsymbol{\alpha}_s} \right|_* + \bar{\mathbf{u}}_s^T \left. \frac{\partial \mathcal{S}}{\partial \boldsymbol{\alpha}_s} \right|_* \quad (14b)$$

If the objective function is a structural response dependent only on the structural displacements and design variables (e.g., a stress component in an element), then $J = J(\mathbf{u}_s(\boldsymbol{\alpha}_s), \boldsymbol{\alpha}_s)$.

It is well known that in the linear case, the adjoint equation (14a) is a linear system featuring the transpose of the stiffness matrix, which has been already stored and factorised during the solution of the primal problem. The partial derivatives $\frac{\partial J}{\partial \mathbf{u}_s}$, $\frac{\partial J}{\partial \boldsymbol{\alpha}_s}$ and $\frac{\partial \mathcal{S}}{\partial \boldsymbol{\alpha}_s}$ are easily evaluated with *CoDiPack*. After solving the primal problem, an extra solution is carried out to perform the AD registration phase:

- *Input registration:* the design variables $\boldsymbol{\alpha}_s$ and the structural displacements \mathbf{u}_s are registered as inputs. Since at the current development stage of *AUGUSTO++* the allowed design variables refer to property or material parameters which are known to all the processes, all the processes register as input each one of the design variables. On the contrary, each process register as input only the components of \mathbf{u}_s that were allocated on it according to the mesh distribution, even if at the end of the linear system solution, the entire vector \mathbf{u}_s is made available to all the processes.
- *Output registration:* the response J and the residual \mathcal{S} are registered as outputs. The former is registered by a process only, the latter is distributed, therefore each process registers its own components.

After solving Eq.(14a) and Eq.(14b), the entire vectors $\bar{\mathbf{u}}_s$ and $\frac{dJ}{d\boldsymbol{\alpha}_s}$ are made available on all the processes.

The computational path from the inputs to the outputs involves the solution of a linear system. Recording all the passages related to the solution of the linear system is generally not possible (need for source code, and unbearable memory requirements). To circumvent this issue, the *CoDiPack* taping is interrupted right before the solution of the linear system and it is resumed after the end of it. The missing part is the adjoint of the linear system. According to reference [42], the reverse mode of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ problem is

$$\mathbf{A}^T \mathbf{s} = \bar{\mathbf{x}} \quad (15a)$$

$$\bar{\mathbf{A}} += -\mathbf{s} \cdot \mathbf{x}^T \quad (15b)$$

$$\bar{\mathbf{b}} += \mathbf{s} \quad (15c)$$

where \mathbf{s} is the solution of the adjoint of the linear system represented in Eq.(15a). The adjoints of RHS ($\bar{\mathbf{b}}$) and of the system matrix ($\bar{\mathbf{A}}$) constitute the previously mentioned lost information that could not be obtained during tape evaluation. The solution of the problem in Eq.(15) is handled in *CoDiPack* through the *External Function Helper* class.

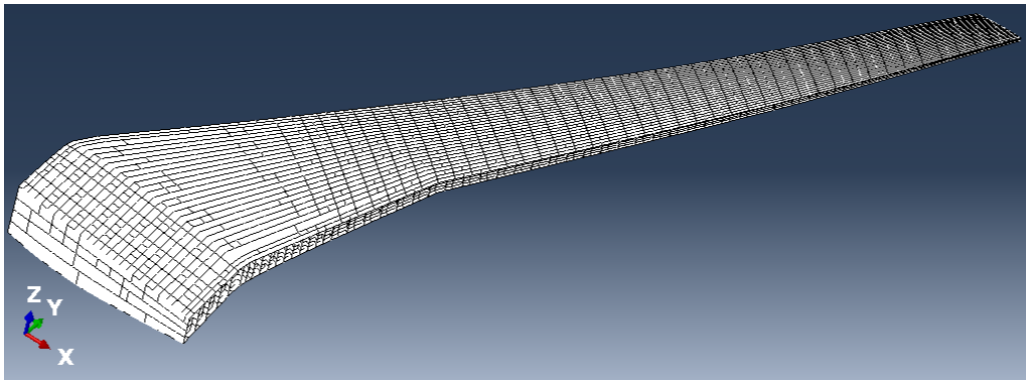
When considering the aerostructural coupling, and selecting an aerodynamic objective function and design variables the, Eq.(14b) must not be solved, and the RHS of Eq.(14a) becomes $\bar{\mathbf{u}}_f^T \mathbf{H}_{MLS}$. Moreover, it is not needed to perform the registration phase for the structural solver. This is the case of the aerostructural optimisation shown in section 4.

3.5 Preliminary verification

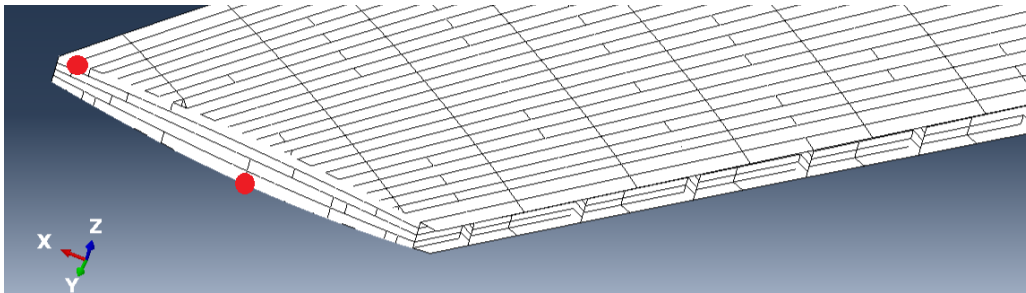
Verification of *AUGUSTO++* has been done on a plethora of beam, plate and stiffened plate models against *Nastran* results. Two selected verification tests are described in the following subsections .

3.5.1 Structural analysis testcase

This subsection shows the verification of results carried out considering the NASA Common Research Model (CRM). The finite element model Version 15 has been selected from reference [43]. The model is composed by 14134 CBAR elements and 21349 CQUAD4 elements (35483 in total), whereas the total number of nodes is 13858. The structure is completely made of aluminum. All the nodes at the wing root are fixed. A pictorial view of the FE model is given in Fig.2a. Figure 3 shows the good agreement of the deflected shape



(a) Global view of the finite element model.



(b) Monitored nodes (Tab.2).

Figure 2: Views of the Nasa CRM V.15

predicted by *Nastran* and *AUGUSTO++*. *AUGUSTO++* solution features a slightly stiffer response attributed to a larger stiffness of the beam element. From the quantitative point of view, Table 2 compares the vertical displacement U_z and rotation R_x for two nodes on the wing tip of the model.

3.5.2 Structural optimisation testcase

The finite element model is an aluminum plate of uniform thickness. The structure is clamped on one edge (red coloured in Fig.4a) and loaded with distributed shear loads on two adjacent edges (black segments in Fig.4a). Five distinct shell property cards are defined as design variables, resulting in a division of the panel into different sectors, as

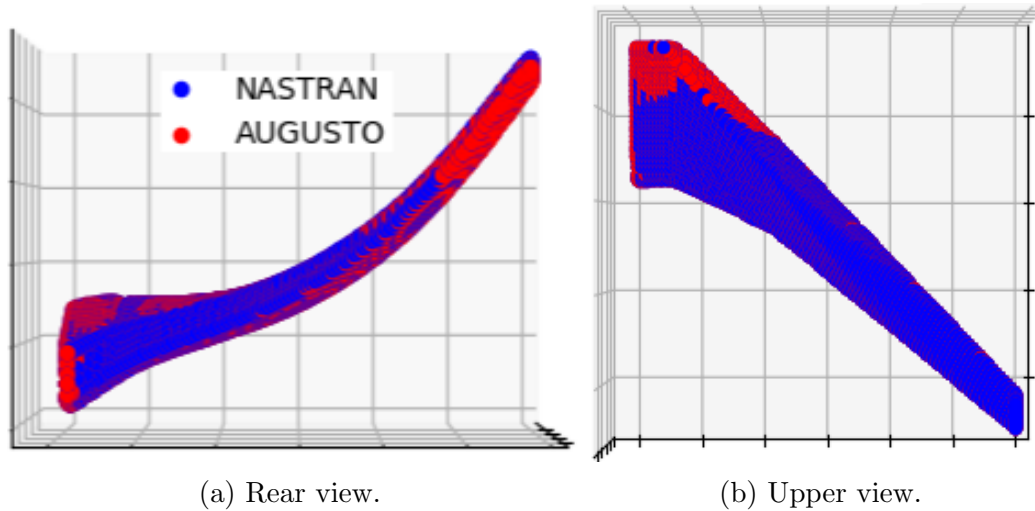


Figure 3: Nasa CRM: deflection comparison between *Nastran* and *AUGUSTO++*.

Table 2: Generalized nodal displacements (in *mm* or *rad*) and relative error between *Nastran* and *AUGUSTO++*.

node ID	component	<i>Nastran</i>	<i>AUGUSTO++</i>	$err_r(\%)$
7178	U_z	4143.044	4105.809	0.898735
	R_x	0.247399	0.245396	0.809785
13558	U_z	3972.113	3937.123	0.880891
	R_x	0.246879	0.244878	0.810277

shown in Fig.4b where areas with thickness from t_1 to t_5 are the ones with colors from red to fuchsia. The design variables initial values, lower and upper bounds are 12 *mm*,

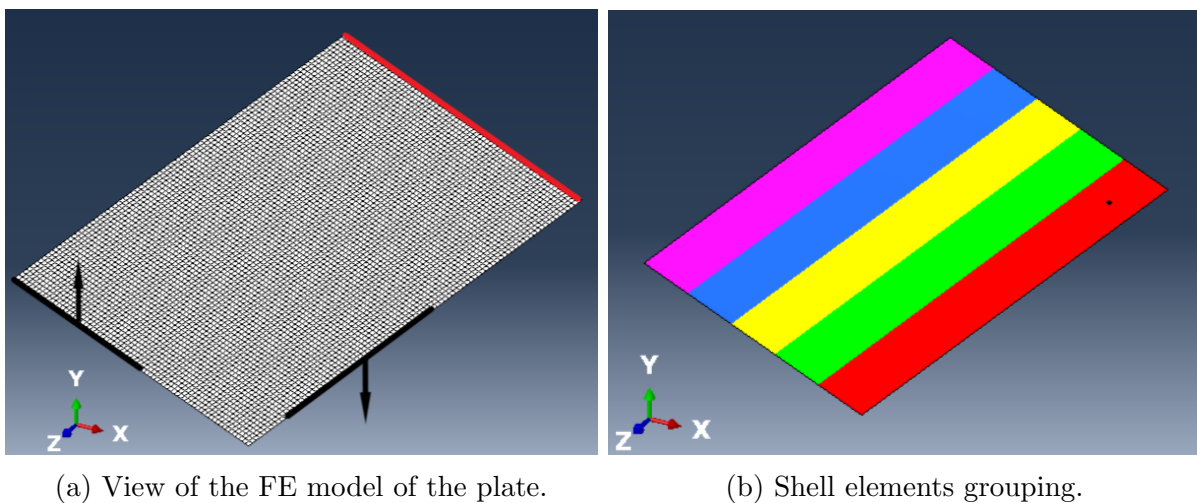


Figure 4: FE model for the structural optimisation.

1 *mm* and 20 *mm* respectively.

It is desired to reduce the weight of the structure keeping the Von Mises stress of a single

shell element (black-coloured in the red area in Fig.4b) below the threshold of $\sigma^{THS} = 270 \text{ MPa}$. The Von Mises stress is evaluated at the centroid and half-thickness of the element. The stress values computed by *Nastran* and *AUGUSTO++* for the initial design ($t = 12 \text{ mm}$) are 106.4144 MPa and 106.4311 MPa with a relative error of 0.0157%.

The optimisation history is depicted in Fig.5. It can be inferred that *Nastran* reaches the optimal solution in roughly one third of the evaluations needed by *AUGUSTO++*. This difference in performance is attributed to *Nastran*'s numerical optimisation software overperforming *SciPy* library SLSQP algorithm. Nevertheless, as apparent in Table 3, the

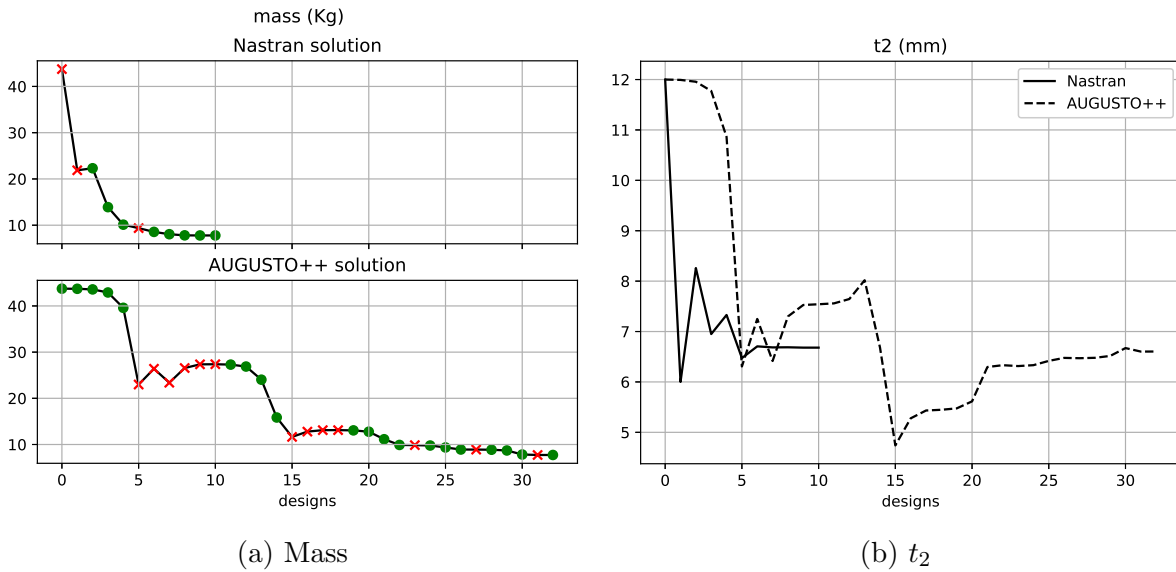


Figure 5: Optimisation history (green dot/ red cross: feasible/unfeasible design).

optimal designs feature similar values of mass and thickness t_2 (which is the only design variable that is not brought to the lower bound).

Table 3: Structural optimisation results summary (thicknesses in mm and mass in Kg).

	t_1	t_2	t_3	t_4	t_5	mass
<i>Nastran</i>	1.0	6.6804	1.0	1.0	1.0	7.786
<i>AUGUSTO++</i>	1.0	6.6036	1.0	1.0	1.0	7.730
$err_r(\%)$	0.0	1.150	0.0	0.0	0.0	0.720

4 PRELIMINARY RESULTS

As already anticipated, the framework described in [19] has been modified by replacing the structural solver *pyBeam* with *AUGUSTO++*. This section presents a comparison between the results of an aerostructural optimisation when considering the linear and nonlinear structural behaviour of the wing.

4.1 Aerostructural wing-shape model

The testcase is the one used in references [19]. It is based on the Nasa CRM testcase, however, the structure is modeled with equivalent beams ([44]) and the synthetic Young

Modulus has been tuned in order to obtain, at aeroelastic equilibrium, a wing-tip deflection of approximately the 6% of the semi-span. The FE model is represented in figure 6a. The wing is simply supported at the wing-fuselage intersection and symmetry condition is imposed at the root of the wing.

The flow model is based on Euler Equations. The fluid mesh is composed by 1.529.927 tetrahedral elements while the wing boundary is modelled with 71.998 triangular elements (figures 6b and 6c). Steady flight at sea level ($\rho = 1.2250 \text{ kg/m}^3$) at $M = 0.85$ is assumed. The interested reader is referred to the original works for further details on the testcase. The design variables for the aerostructural optimisation test are the vertical displacements

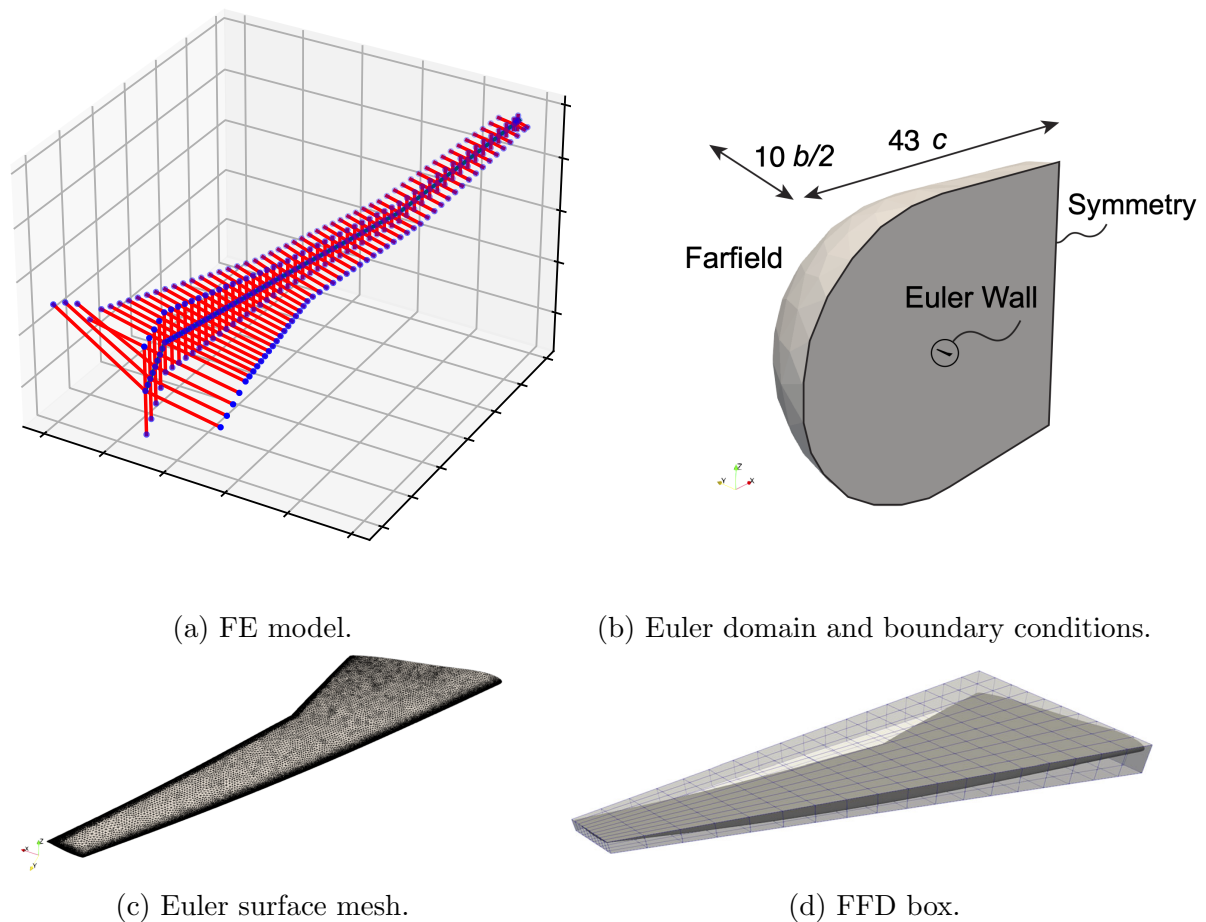


Figure 6: Testcase, from [19].

of 418 control points that define the FFD box (Fig.6d), based on Bezier functions of order 10, 18 and 1 along the chord, wing and thickness directions, respectively. The control points on the symmetry plane are kept fixed.

The constraints regulate the thickness ratio of the airfoils at different cross sections along the wingspan, as summarised in Tab.4.

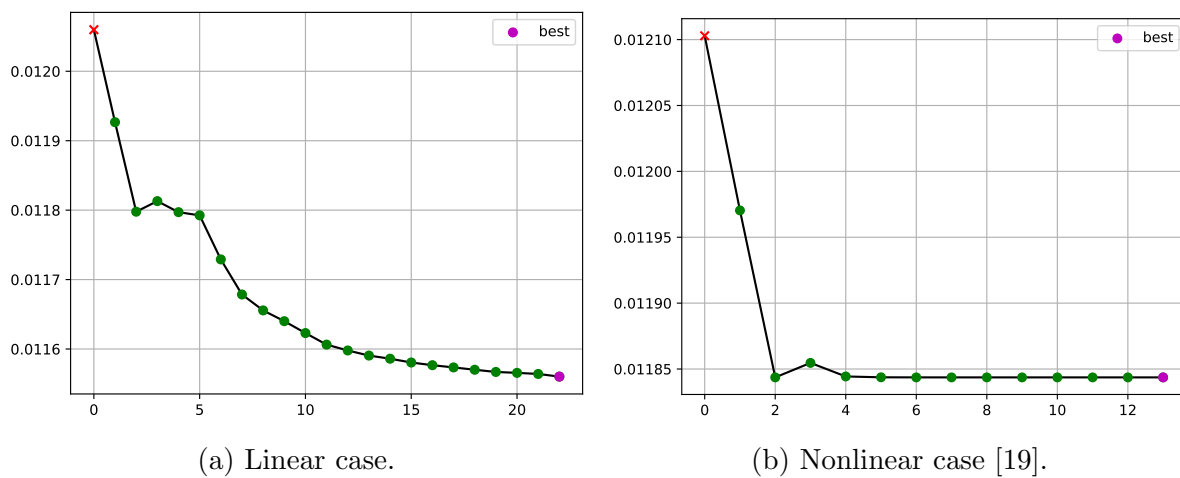
The objective function is the drag coefficient C_D . Summarizing, the jig-shape of the wing is changed to obtain, in reference cruise conditions and relative flying shape, the lowest C_D . Also, the angle of attack is automatically adjusted at the aerodynamic solver level to provide a lift coefficient $C_L = 0.5$ of the deformed wing.

Table 4: Geometric constraints. From [19].

spanwise pos. (%)	0.34	16.32	27.01	38.49	49.76	60.74	71.89	83.07	94.14
$(t/c) _{min}$ (%)	15.6	12.5	11.2	10.4	10.0	9.8	9.6	9.5	9.5

4.1.1 Results

The optimisation history considering the linear modelling of the wing structure is depicted in Fig.7 in terms of C_D . As apparent, the aerostructural optimisation considering the nonlinear structure converges faster to the optimal value. The pressure coefficient

Figure 7: C_D vs optimisation iterations (green dot/ red cross: feasible/unfeasible design).

distribution over the optimised semi-wings is depicted in Fig.8. The optimal C_D found in the linear case is slightly lower than the one found in the nonlinear case as summarised in Tab.5. For both cases, the optimised solutions present a smaller in-flight deflection than the non-optimised cases.

Table 5: C_D and wingtip displacement (in m) for optimal configurations.

	C_D^{base}	C_D^{optim}	Δ (%)	δ_{tip}^{base}	δ_{tip}^{optim}	Δ (%)
linear	0.0121	0.0116	4.132	1.866	1.777	4.77
nonlinear	0.0121	0.0118	2.479	1.845	1.745	5.42

5 CONCLUSIONS

This paper discussed the development of an in-house finite element solver for adjoint-based aerostructural optimisation.

The software solving each multidisciplinary block of the overall framework is briefly described, and the formulation of the coupled problem and relative adjoint system are presented. A brief overview of the solution method and the use of algorithmic differentiation

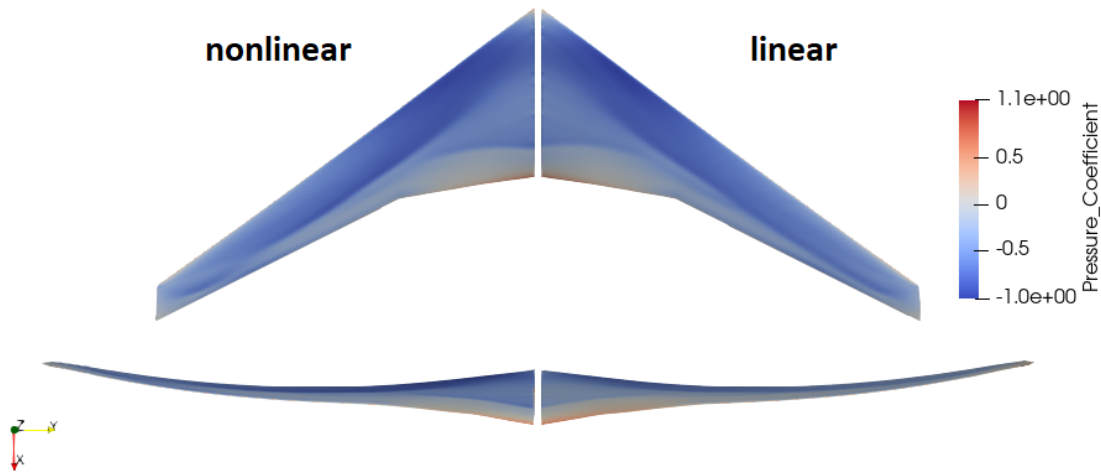


Figure 8: Pressure coefficient distribution and deflection at the relative optimal design in flight conditions.

software is also provided.

The development of the structural solver is then described a bit more in details, with practical choices at the implementation level, such as parallelisation, wrapping and integration of algorithmic differentiation library. A simple purely structural testcase is then shown comparing the in-house code predictions with the ones of the commercial software *Nastran*; a structural optimisation testcase follows, and, again, results of the present capability are checked against the ones provided by *Nastran*.

In the last section, a coupled aerostructural optimisation is performed considering as design variables the jig-shape of the wing, and minimizing the drag coefficient of the wing in its deflected shape. The optimisation is carried out considering linearly-behaving structures, as opposed to the nonlinear case found in the literature. Results are shown and briefly discussed.

5.1 Future works

The platform is currently under further development to consider concurrent mixed aerodynamic and structural design variables and responses. Moreover, the structural solver is currently being improved modelling geometric nonlinearities and composite materials.

ACKNOWLEDGEMENTS

The development of the FE software for structural analysis has been carried out within the research project OPERATIONAL (TED2021-131119A-I00) financed by MCIN/AEI/10.13039/501100011033 and European Union NextGenerationEU/PRTR.

REFERENCES

- [1] <https://www.iata.org/en/pressroom/pressroom-archive/2021-releases/>.
- [2] www.nasa.gov/aeroresearch/a-future-aircraft-design-supercomputed, .
- [3] www.nasa.gov/press-release/nasa-issues-award-for-greener-more-fuel-efficient-air

- [4] www.boeing.com/features/2019/01/spreading-our-wings-01-19.page.
- [5] Integration and digital demonstration of low-emission aircraft technologies and airport operations. <https://cordis.europa.eu/project/id/101096055>.
- [6] R. T. Haftka. Optimization of flexible wing structures subject to strength and induced drag constraints. *AIAA Journal*, 15:1101–1106, 1977. ISSN 0001-1452,1533-385X.
- [7] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA Journal*, 39(11):2051–2061, 2001.
- [8] J. Martins, J. Alonso, and J. Reuther. Aero-structural wing design optimization using high-fidelity sensitivity analysis. In *Proceedings - CEAS Conference on Multidisciplinary Aircraft Design Optimization*, Cologne, Germany,, 2001.
- [9] M. Barcelos and K. Maute. Aeroelastic design optimization for laminar and turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1813–1832, 2008.
- [10] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, Reno, US. 1992.
- [11] M. C. Sandford, D. A. Seidel, C. V. Eckstrom, and C. V. Spain. Geometrical and structural properties of an aeroelastic research wing (arw-2),. <https://ntrs.nasa.gov/api/citations/19890010728/downloads/19890010728.pdf>, April 1989. NASA-TM-4110.
- [12] G. Kenway, G. Kennedy, and J. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA Journal*, 52(5):935–951, 2014. ISSN 00011452.
- [13] E. van der Weide, G. Kalitzin, J. Schluter, and J. Alonso. Unsteady turbomachinery computations using massively parallel platforms. In *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno US*. 2006.
- [14] G. Kennedy and J. Martins. Parallel solution methods for aerostructural analysis and design optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, US. 2010.
- [15] L. Wang, B. Diskin, R. T. Biedron, E. J. Nielsen, and O. A. Bauchau. High-fidelity multidisciplinary sensitivity analysis and design optimization for rotorcraft applications. *AIAA Journal*, 01 2019.
- [16] NASA. Fun3d - fully unstructured navier-stokes. <https://fun3d.larc.nasa.gov/chapter-1.html>.
- [17] O. Bauchau, C. Bottasso, and Y. Nikishkov. Modeling rotorcraft dynamics with finite element multibody procedures. *Mathematical and Computer Modelling*, 33(10):1113–1137, 2001. ISSN 0895-7177. doi:[https://doi.org/10.1016/S0895-7177\(00\)00303-4](https://doi.org/10.1016/S0895-7177(00)00303-4).

- [18] R. Sanchez, T. Albring, R. Palacios, N. R. Gauger, T. D. Economon, and J. J. Alonso. Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation. International Journal for Numerical Methods in Engineering, 113(7):1081–1107, 2018.
- [19] R. Bombardieri, R. Cavallaro, R. Sanchez, and N. R. Gauger. Aerostructural wing shape optimization assisted by algorithmic differentiation. Structural and Multidisciplinary Optimization, 64:739–760, 2021.
- [20] N. G. M. Sagebaum, T. Albring. High-performance derivative computations using codipack. ACM Transactions on Mathematical Software (TOMS), 45(4), 2019. URL <https://dl.acm.org/doi/abs/10.1145/3356900>.
- [21] D. Kraft. A software package for sequential quadratic programming. Technical Report DFVLR-FB 88-28, DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany., 1988.
- [22] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL <http://www.scipy.org/>.
- [23] C. Farhat, P. Geuzaine, and G. Brown. Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter. Computers & Fluids, 32, 2003.
- [24] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. International Journal for Numerical Methods in Engineering, 56(6):911–933, 2003. ISSN 00295981.
- [25] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian Methods. John Wiley & Sons, Ltd, 2004. ISBN 9780470091357.
- [26] F. Palacios, T. D. Economon, A. Aranake, S. R. Copeland, A. K. Lonkar, T. W. Lukaczyk, D. E. Manosalvas, K. R. Naik, S. Padron, B. Tracey, A. Variyar, and J. J. Alonso. Stanford University Unstructured (SU2): Analysis and Design Technology for Turbulent Flows. In 52nd Aerospace Sciences Meeting. American Institute of Aeronautics and Astronautics, 2014.
- [27] F. Palacios, T. D. Economon, A. D. Wendorff, and J. J. Alonso. Large-scale aircraft design using su2. In 53rd AIAA Aerospace Sciences Meeting, 2015.
- [28] M. Pini, S. Vitale, P. Colonna, G. Gori, A. Guardone, T. Economon, J. Alonso, and F. Palacios. Su2: the open-source software for non-ideal compressible flows. Journal of Physics: Conference Series, 821(1):012013, 2017.
- [29] T. Albring, M. Sagebaum, and N. Gauger. Efficient aerodynamic design using the discrete adjoint method in su2. In 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2016.
- [30] R. Dwight. Robust mesh deformation using the linear elasticity equations. pages 401–406, Ghent, 2009. Springer Berlin.

- [31] R. Cavallaro, A. Iannelli, L. Demasi, and A. M. Razón. Phenomenology of nonlinear aeroelastic responses of highly deformable Joined Wings. Advances in Aircraft and Spacecraft Science, 2(2):125–168, April 2015.
- [32] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. web: <http://www.cs.umd.edu/~mount/ann>, 2010. URL <http://www.cs.umd.edu/~mount/ANN/>.
- [33] T. Albring, M. Sagebaum, and N. R. Gauger. Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2015. ISBN 978-1-62410-368-1.
- [34] J. Samareh. Aerodynamic shape optimization based on free-form deformation. Number AIAA 2004-4630. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Aug. 2014.
- [35] L. Pustina, R. Cavallaro, and G. Bernardini. Nerone: An open-source based tool for aerodynamic transonic optimization of nonplanar wings. Aerotec. Missili Spaz., 98: 85–104, 2019.
- [36] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, and T. S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In Proceedings, 11th European PVM/MPI Users’ Group Meeting, pages 97–104, Budapest, Hungary, September 2004.
- [37] R. H. Macneal. A simple quadrilateral shell element. Computers Structures, 8(2): 175–183, 1978. ISSN 0045-7949. doi:[https://doi.org/10.1016/0045-7949\(78\)90020-2](https://doi.org/10.1016/0045-7949(78)90020-2).
- [38] R. Levy and W. Spillers. Analysis of geometrically nonlinear structures. Number v. 1. Kluwer Academic Publishers, Dordrecht, Netherlands, 2003. ISBN 9781402016547.
- [39] Element Library Reference. Siemens Industry Software, Granite Park One 5800 Granite Parkway Suite 600 Plano, TX 75024 USA.
- [40] P. Amestoy, I. S. Duff, J. Koster, and J.-Y. L’Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM Journal on Matrix Analysis and Applications, 23(1):15–41, 2001.
- [41] D. M. Beazley. Swig: An easy to use tool for integrating scripting languages with c and c++. In Proceedings of the 4th Conference on USENIX Tcl/Tk Workshop, 1996 - Volume 4, TCLTK’96, pages 15–15, Berkeley, CA, USA, 1996. USENIX Association.
- [42] Codipack documentation. <https://www.scicomp.uni-kl.de/codi/index.html>, 2023.
- [43] . URL <https://commonresearchmodel.larc.nasa.gov/>.
- [44] J. Vela Peña. Aeroelastic calculations on an equivalent beam-based nasa crm. B.Sc. thesis uc3m, 2019.



AN EFFICIENT NUMERICAL MODELLING FOR AERODYNAMIC SHAPE OPTIMIZATION

Sara Pucciarelli^{1*}, Fernando Lau¹ and Afzal Suleman^{1,2}

1: IDMEC, Instituto Superior Técnico Lisboa.
Universidade de Lisboa

Av. Rovisco Pais, No. 1, 1049-001, Lisbon, Portugal
sara.pucciarelli, suleman, lau@tecnico.ulisboa.pt <https://tecnico.ulisboa.pt>

2: Department of Mechanical Engineering, Center for Aerospace Research
University of Victoria
3800 Finnerty Rd, Victoria, BC V8P 5C2, Canada suleman@uvic.ca, <https://www.uvic.ca>

Abstract. *The current trend in the aeronautical industry aims at an increased aircraft efficiency at all phases of the mission and a reduction on the environmental impact. Morphing wing architectures address these objectives by optimally adapting the airfoil shape during flight. The optimization of airfoil profiles is complex and computationally expensive since it requires a trade-off among different objectives, constraints and numerical solutions. The work-in-progress paper presents a data-driven approach to perform the aerodynamic analysis and optimization of a morphing airfoil over a wide range of Reynolds numbers, angles of attack (AOA) and Mach numbers. The methodology is based on iterative surrogate models implemented in the MACH-Aero framework. Surrogate models such as artificial neural networks, rely on a design dataset to compute the aerodynamic coefficients without solving the governing equations. A dataset has been created with thousands of high fidelity Computational Fluid Dynamics (CFD) simulations with the purpose of training a multi-layer perceptron network spanning the subsonic to subsonic ranges. Once the network is trained, it returns as output the lift, drag and pitch coefficients. Hundreds of airfoil shapes have been selected from the University of Illinois at Urbana-Champaign (UIUC) database and the supercritical airfoils database made available by NASA. Python scripts have been used such as PyHpy for the meshing, Prefoil for the space sampling, and PyOptSparse for the optimization algorithms.*

Keywords: Aerodynamic shape optimization, machine learning , morphing architectures, aircraft design, performance

1 INTRODUCTION

The current trend in the aeronautical industry is to increase the efficiency of the aircraft through all phases of the mission. Aerodynamic shape optimization (ASO) is an important design driver to increase efficiency and reduce environmental impact. Current designs are based on a fixed airfoil design with control surfaces, selected from a trade-off study based on a diverse set of requirements. Morphing wings present a potential solution to improve aerodynamic efficiency throughout the mission envelope. [1]

Surrogate based optimization (SBO) presents a low cost evaluation model which reduces time and computational cost. Once the surrogate is built, it is used in lieu of expensive full-order simulation. Ideally, SBO provides the required accuracy and convergence depending on model efficiency, design space size and infilling effectiveness [2].

This work-in-progress paper presents a surrogate based model to optimize morphing airfoils from a subsonic regime to a transonic one. The surrogate model has been built based on a dataset made up of a large number of candidate designs.

The paper is organised as follows. In Section 2, the state-of-art in AOS and SBO have been presented. In Section 3, a surrogate model based on Artificial Neural Network (ANN) algorithm is discussed. Finally, the key steps to develop the dataset used for training is explained in Section 4.

2 THEORETICAL BACKGROUND

2.1 Aerodynamic Shape Optimization: state-of-art

Aerodynamic Shape Optimization (ASO) has gained increasing attention by designers and engineers to improve aerodynamic performance of aircraft. With the development of computer science tools and processes and Computational Fluid Dynamics (CFD), ASO has become an important step in modern aircraft design [3].

The current design methodology is described in Fig.1.

The process starts by setting the optimization model with the definition of the design

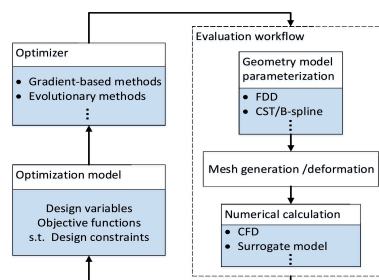


Figure 1: Flow chart of ASO - Yan et al (2019) [3]

variables and the objective function subject to the design constraints. Each objective function is evaluated for a set of design constraints. There are two main types of constraints: geometrical constraints and aerodynamic constraints. The design variables are defined using shape parameterization. This evaluation requires the following steps:

- **Geometry model parameterization** is required to define the designed variables. There are various approaches to parameterize the aerodynamic shape. The literature highlights the class shape function (CST) [4] and free-form deformation [5, 6] methods that provide more geometric freedom by increasing the number of design

variables. The Bézier-curve parameterization is used in this work based on the work by Samareh [7]. The Bezier form is a far better representation than the power basis, even though mathematically equivalent, because the computation of Bernstein polynomials is a recursive algorithm that minimizes the round-off error. Also, the control points are more closely related to the curve position and they approximate the curve.

- **Mesh generation.** The mesh is continuously updated with the geometry given by the previous iteration of the optimization process.
- **Numerical Analysis.** In this step, the objective functions are computed using CFD or surrogate models depending on the architecture of the algorithm. The results are passed to the optimization model block which evaluates the accuracy and decides if the process should continue.

Finally, the optimizer connects the optimization model and the evaluation workflow. It is basically the core of the process since it practically evaluates the updating of the design variables. The adjoint method pioneered by Jameson [8] enables the accurate and efficient computation of gradients in high dimensional space. The method consists in the computation of derivatives of interest based on the gradient in a numerical approach. It is commonly available in CFD-solvers. MACH-Aero [9] is an example of implementation of the adjoint optimization. The workflow of MACH-Aero has been considered as a reference optimization process. The computational tools included in MACH-Aero have been used to generate a reliable and representative training dataset for our model.

2.2 Surrogate Based Optimization

Surrogate models are mostly based on algebraic relations and evaluate the response of the system without solving the governing equations but estimating its response. Once the surrogate is built, it evaluates the objective functions at x locations defined in the design of experiments (DoE). SBO is composed by two phases[10]:

- Design space is processed by the DoE and representative location are defined and evaluated by the full-order aerodynamic model. The resulting dataset trains the initial surrogate.
- Refinement of the model by infilling new training samples, derived by solving a sub-optimal problem, the so called online training.

The DoE is the sampling strategy to adopt in order to maximize the amount of information extracted from a limited number of samples [3]. Therefore, the primary interest is finding the best trade-off between bias error and variance.

A model is defined as a function that maps input to an output. Therefore the surrogate model can be considered as a "cheap-to-evaluate" mathematical approximation that mimics the full-order expensive response of an original system over the design space [11]. The process is clearly sketched in Fig.2.

The surrogate modelling deals with [2]:

- sampling effectively the design space in order to provide representative locations;

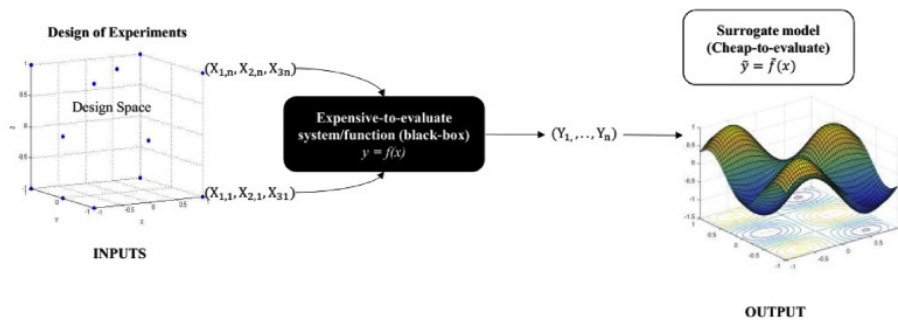


Figure 2: Surrogate model flowchart - Yondo et al (2017) [11]

- constructing a model from the available data;
- assessing the error attached to the model.

Lately the choice of the surrogate is not trivial but depends on the nature of the domain and on the characteristics of the problem. A useful classification can be found in Queipo et al [2]:

- Design variables are classified as parametric or not-parametric. The parametric approaches (such as polynomial regression and Kriging) assume that there is a global relation between the input data and the output desired variables. Non-parametric ones (such as Multi Expertise and Artificial Neural Network) use a combination of local information in different regions and finally an overall surrogate is obtained.
- The surrogate model can be built with the following methodologies: interpolation, regression or projection.

3 METHODOLOGY

3.1 Theoretical Background

A popular approach for ASO supported by machine learning (ML) is based on constructing a data-driven surrogate model in order to simplify the geometric design space, compute the aerodynamic derivatives without solving the governing equations and create an optimization algorithm characterized by a reasonable computational time.

The ANN is the core component of deep learning. Actually it is the most used algorithm for ML-base ASO. The simplest example of ANN is the Multilayer perceptron (MPL). The unit base of MLP is the perceptron or neuron which are clustered in multiple levels or layers. If inner layers are hidden, ANN became deep neural network (DNN). Neurons are deeply connected and the overall system is the network. Each neuron is characterized by its activation number. It takes as input the activation number of the first layer and return activations of output layer. The critical part is to understand how it obtains the results and how reliable they are. Generally a MLP is fully connected as it is shown in Fig.3.

The output value are the weighted sum of the previous activation's layer in which the outputs of the model may be fed back into itself. To increase the reliability of network, each neuron is associated with an unknown bias which aims to neglect the perceptron if it is meaningfully active in the current iteration. Within the i -th neuron of the $L+1$ -th

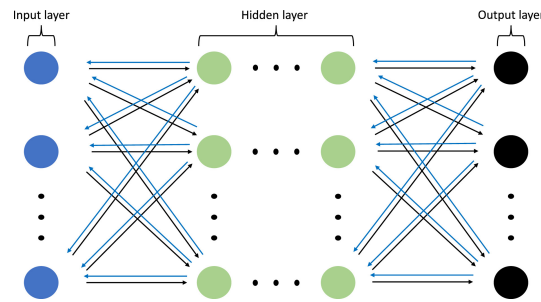


Figure 3: ANN-architecture- Li et al (2022) [10]

layer, the following operation is consequentially

$$a_i^{(L+1)} = \sigma * (w^T * a^{(L)} + b_i). \quad (1)$$

The σ function is a typical activation function called sigmoid function, which output are within $[0, 1]$ in order to restrict the activation values into that range. In DNN, activation functions are important as they help in learning and making sense of non-linear and complex mappings between the inputs and the corresponding outputs [12]. The main problem is computing the optimum values of weights and bias such that the network can provide the best results. Parameters are then defined in the training phase. There are several ways to train a network but the most diffused is certainly the back-propagation algorithm [13].

3.2 The aerodynamic surrogate

In this study, we have opted for the MLP implemented by Moin et al [14]. The MLP takes the airfoil coordinates and flight regime as inputs and predicts the aerodynamic coefficients without relying on CFD solvers. The authors of the neural network trained it using a wide database of NACA series airfoils in a low subsonic and incompressible regime, achieving very good results for the prediction of the aerodynamic polar.

The final architecture, depicted in Fig. 4, was designed to strike a balance between accuracy and costs. The authors utilized the Keras API in Python and employed a fully connected layer structure for the model implementation. The model consists of four layers, each with a decreasing number of neurons. The chosen model shape is [512, 256, 128, 3].

Based on their findings and after consulting with the authors, we have decided to train the same model for our specific application. For training, we employed the Adam optimizer [15]. The model's performance was evaluated using both the Root Mean Square Error (RMSE) and R-squared values, while the Mean Square Error (MSE) was used to control the loss during training. We trained the network for 50 epochs, feeding the data in multiple batches of 128 epochs.

4 DATASET CREATION

The creation of a high-quality dataset is indeed crucial for training and validating the model effectively. The key steps for the building of the dataset are:

1. Pre-processing of the University of Illinois at Urbana-Champaign (UIUC) Database [16] and NASA Super Critical Aerofoil Database;

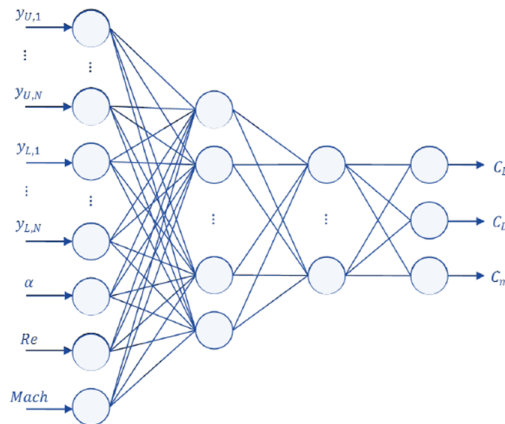


Figure 4: Artificial neural network- Moin et all (2022) [14]

2. Sampling of each shape in 20 points via cosine distribution to represent each shape in the dataset file;
3. Defining flight conditions and application regime in order to create the most representative database ;
4. Computing lift coefficient (C_l), drag coefficient (C_d) and pitch coefficient (C_m). To enhance the aerodynamic theory behind those coefficients we recommend to consult [17];
5. Studying an infilling criteria to augment the size of the dataset.
6. The final dataset is saved in a csv file.

In each step, the process has been automated by coding several pipelines to simulate the large number of aerofoils needed for the database. This approach is viable since the computational tools chosen, such as *pyHyp*[18] for mesh generator, *ADFlow*[19] for CFD simulations, and *prefoil* as sampler tool are all supported by Python API. All tools are an open source distribution by MDOLab.

4.1 Computational fluid dynamic solver set up

Both subsonic and transonic CFD simulations are conducted with ADflow software to solve the steady RANS equations [20]. The Spallart-Allmaras (SA)[21] is the recommended option for external flows.

In order to realistically optimize a morphing airfoil, hundreds of shapes have been collected. For each shape different flight conditions have been simulated since our objective is the optimization of the airfoil from the subsonic regime up to the transonic regime.

The range of flight condition is reported in Table 1

For each airfoil 126 simulations are collected. An infilling criteria has been tested in order to infill the sample with spare AoA and respect the budget cap imposed at the beginning of the work. The infilling criteria chosen is a polynomial interpolation, linear for the C_l and of the second order for C_d and C_m , according to the thin profile theory [22]. Infilling the samples is a strategy that allows to save time considerably. The dataset has been computed in 68% of the time necessary compared to the computation for each sample

Table 1: CFD set up

Parameter	Range	Total
Reynolds	1,4e7 – 8,7e6 – 5,1e7	3
Mach	0,35 - 0,45 - 0,55 - 0,65 - 0,75 - 0,85	6
AoA	[0 – 2 – 4 -6] (simulated) [1 – 3 – 5] (infilled)	7

via CFD simulation. The method seems to be efficient for subsonic regime but not so accurate for the transonic regime.

4.2 dataset preparation and pre-processing

The aerofoils used in this work are taken from the UIUC database which collects almost 1600 different shapes that have been used in real-world applications. The UIUC library covers mainly subsonic and transonic applications. To increase the number of shapes suitable for the transonic regime, 21 super-critical airfoils have been picked from the NASA database[23]. The resulting complete training dataset consists of 1433 geometries.

The UIUC Aerofoil Coordinate database provides a collection of real aerofoils defined

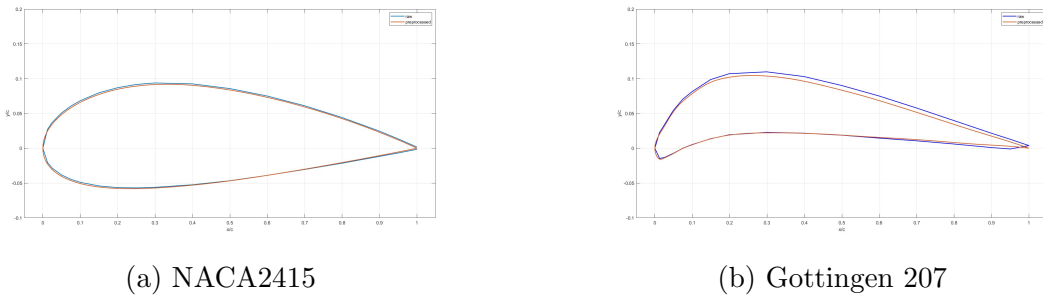


Figure 5: Preprocessed aerofoil example

by a set of coordinates. Unfortunately the aerofoil data are not in a uniform format and are not directly usable for the creation of the dataset. A preprocessing method is then necessary. In this work, the same procedure proposed by Li et al [24] has been followed where each shape has been interpolated by a B-spline curve. The number used for the interpolation returns to be significantly important for the characterization of the transonic shock.

Fig. 5 depicts the effect of the standardizing procedure on example airfoils.

4.3 Sampling

Each airfoil is represented in the dataset file by a finite and discrete number of points. According to the study by Moin et al, each airfoil has been sampled via cosine distribution by 20 points. The cosine distribution has been chosen despite of an uniform distribution, since the accurate representation of the leading edge (LE) is necessary to ensure good performance.

The shapes have pre-processed by normalizing the chord and rotating the chord in order to have zero AoA. The leading edge and the trailing edge (TE) are thus fixed respectively in $[0,0]$ and $[1,0]$. Since both points are not allowed to change, they do not contribute

to the training of the surrogate, therefore those positions have been discarded and 20 additional points have been considered. The sampling code used for this task is the open source code *prefoil*, a pySpline-based utility mode that allows to handle airfoil geometries and enable rapid, custom surface mesh generation.

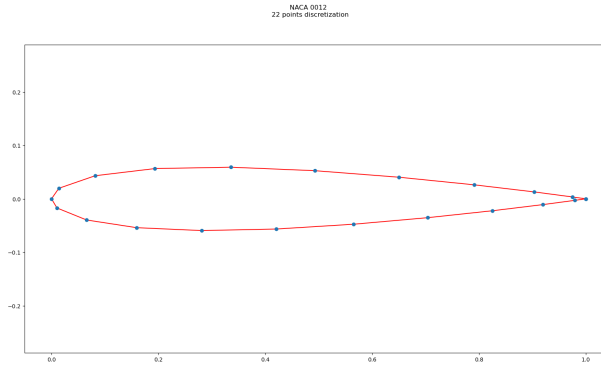


Figure 6: Example of airfoil sampled by 20 points via cosine distribution

4.4 Mesh convergence study

As we already mentioned before, *pyHyp* is the computational tool used to generate automatically the mesh of each shape. Since *ADflow* is a 3D finite volume solver, 3D meshes are needed even for 2D problems. This task has been automatically handled by *pyHyp* that:

- Extrudes a surface mesh from the provided geometry and saves it in a PLOT3D file with extension `.xyz`;
- Obtains a "O" volume mesh layer by layer until the prescribed far-field edge is reached.

Once the meshing process ends, the results are saved in a CFD General Notation System (CGNS) file format. Even though *pyHyp* has been useful in terms of the automation of the process, it imposes some restrictions in terms of the boundary conditions (BC) and the possible patterns of the mesh. The induced and only BCs chosen are viscous wall for the airfoil edge and the far-field condition for the external edge.

The parameters that can be modified are:

- the number of nodes/layer in the off-wall direction; (N)
- the thickness of the first off-wall cell layer (s_0);
- the distance of the far-field (marchdist).

The choice of the number N of nodes/layers is important for the dissipation of the turbulent wake. Moreover, for the purpose of consistency the first layer attached to the wall has to be thick enough such that the dimensionless wall distance defined as

$$y^+ = \frac{u_* y}{\nu} \quad (2)$$

is equal or lower than 1 [25]. In this equation $\nu = \mu/\rho$ is the kinematic viscosity and u_* is the friction velocity, which is estimated with the Schlichting skin friction correlation [26]. Considering the experimental results of the NACA0012, published on the NASA website [27] specific trade off between accuracy and time consuming has been made.

The mesh convergence study has been conducted for both the subsonic regime and the transonic regime, following the Richardson Extrapolation technique [28]. Indeed the discretisation error is estimated by extrapolating the solution of a grid with a fictitious grid spacing on the order of zero:

$$\phi_{exact} = \frac{r^p \phi_1 - \phi_2}{r^p - 1} \quad (3)$$

where ϕ_i is the grid solution value of the i_{th} most refined grid, r is the grid refinement ratio and s is the rate convergence index:

$$p = \frac{\ln \left| \frac{\phi_3 - \phi_2}{\phi_2 - \phi_1} \right|}{\ln r} \quad (4)$$

If the value of the grid refinement ratio is unsure, the procedure requires an estimation of the effective grid refinement ratio as:

$$r_{effective} = \left(\frac{N_1}{N_2} \right)^{(1/D)} \quad (5)$$

where N_i is the total number of grid points used for the i_{th} grid and D is the dimension of the flow domain. The former value is easily extrapolated from pyHpy output, that prints on the terminal the progressing layer for each mesh. The latter is known from the pre-processing phase where the chord has been normalised to the unit.

Table 2: Refinement set up

	N	s_0	y^+	N on walls	Total nodes
COURSE	70	$3,00 * 10^6$	0.64	133	9314
MEDIUM	129	$2,00 * 10^6$	0.43	201	25929
FINE	180	$1,00 * 10^6$	0.21	401	72180

The Reynolds number study has been conducted on 3 refined meshes reported in Table 2. The effective grid resolution computed is equal to $r = 2.7838$. The results of the grid convergence study conducted are presented in Tables 3-4 and Figure 7. The study reveals that the values of C_d converges both in the subsonic and transonic regimes as the number of cells in the grid increases. However, the finest refinement of the mesh yields a significant increase of the CPU time and therefore the medium mesh has been chosen for the computation of the dataset since the discretization error is lower than the threshold of 5%.

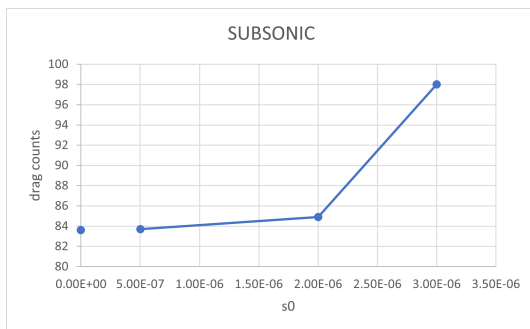
The final mesh used is the one represented in Fig.8

Table 3: Subsonic RE

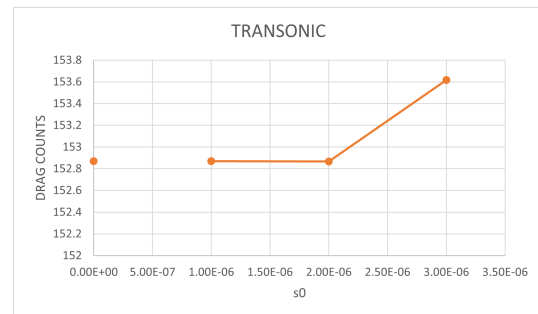
	C_d counts	C_d %error	CPU time [s]
COURSE	98.021	-19.65%	22.70
MEDIUM	84.936	3.65%	30.01
FINE	83.718	2.19%	1536.89
RE	83.599	-	-

Table 4: Transonic RE

	C_d counts	C_d %error	CPU time [s]
COURSE	153.61	-	5.20
MEDIUM	152.87	0.49%	14.93
FINE	152.86	0.001%	52.09
RE	152.85	-	-



(a) Subsonic Case



(b) Transonic Case

Figure 7: Richardson extrapolation plot

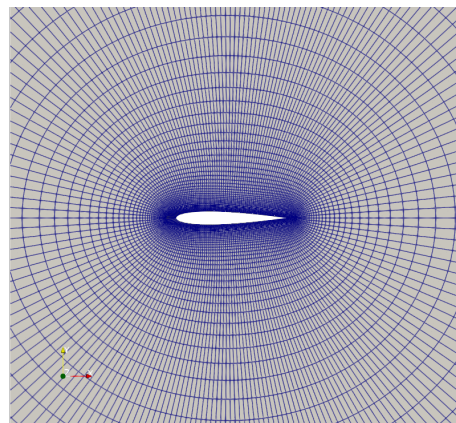
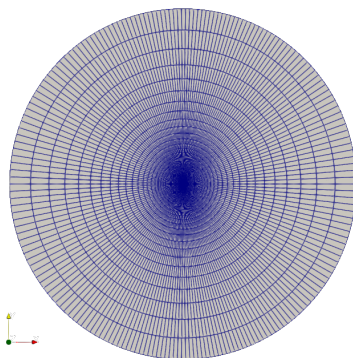


Figure 8: Mesh chosen for the dataset

5 THE INFILLING CRITERIA

A polynomial interpolation has been made to infill the spare AoA in our dataset. In order to verify the accuracy of the strategy, 10 random shapes have been selected and

simulated for a range of AoA [1 – 3 – 5] deg. For each airfoil of the batch, the high fidelity samples have been compared with the respective infilled samples. The first study of the C_d errors highlights that 9.83% of the infilled samples are characterised by a percentage error higher than 5%.

On the one hand, the second detailed study of the errors shows that 76% of infilled samples with $\epsilon_{C_d} > 5\%$ is found in the transonic regime, particularly for Mach number in the range [0.65 – 0.75 – 0.85]. On the other hand, the subsonic regime is well captured by the interpolation. Those results lead to conclude that the strategy adopted is not accurate for the transonic regime. As future work, the transonic regime will be fully simulated and the infilling samples will be kept only for the Mach number equal or lower than 0.55.

6 TRAINING

The aerodynamic surrogate model has been trained with a quarter of the dataset to validate the methodology adopted. Here, only 358/1433 shapes have been simulated collecting a dataset of 42620 samples.

The training losses are depicted in Fig.9 and the accuracy is plotted in Fig.10.

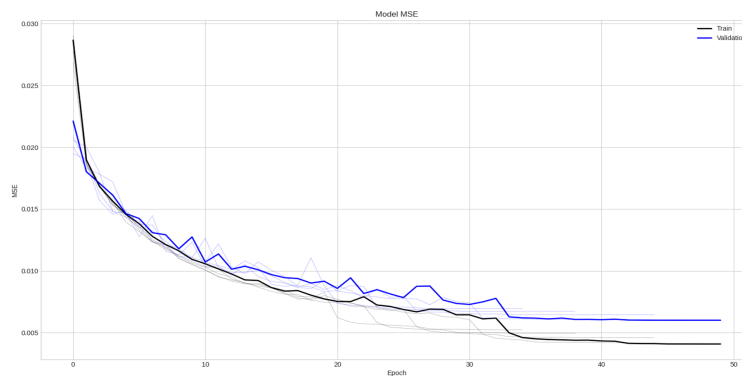


Figure 9: Model loss function

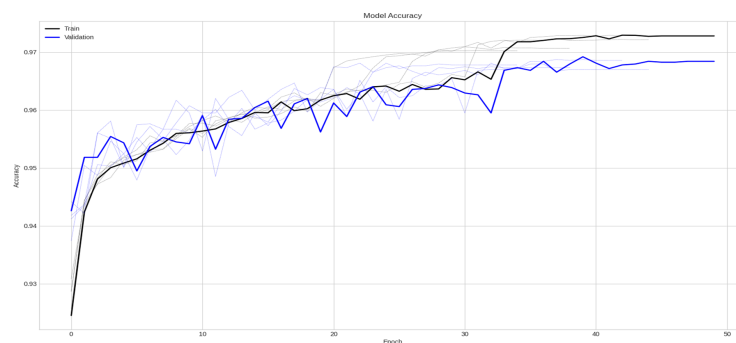


Figure 10: Model accuracy

The losses plots are characterized by an asymptotic trend with a value of 0.005 while the accuracy asymptotic value is equal to 0.97. The performance is justified by the temporary roughness of the dataset.

7 PRELIMINARY REMARKS

The first evaluations of the aerodynamic surrogate model are not as accurate as expected. The following preliminary observations are summarized:

- The MLP depends on the accuracy of the data. Even though the MLP is easily built and adaptable for every application, if the dataset does not adequately represent the design space, the results are not accurate.
- Morphing architectures cover a wide range of flight conditions. Therefore the dataset may require the generation of a prohibitive number of samples in terms of computational cost and time.
- The MLP logic does not search for patterns, which may be useful for the intended purpose.

Current work under development regarding the aerodynamic model include:

- Augment the size of the dataset, infilling more samples for the subsonic regime and discarding the infilled samples for the transonic range.
- Build a convolutional neural network in order to compare the aerodynamic performances of both aerodynamic surrogate models.

Currently, the training of a generative shape surrogate model coupled with the optimizer and the aerodynamic model is being considered. The generative surrogate model chosen is an informed generative adversarial network (InfoGAN) proposed by Chen et al [29] that will be trained based on the airfoils collected in the first phase. The GAN generates realistic shapes and returns each new airfoil parameterized by a Bézier curve.

ACKNOWLEDGEMENTS

The authors acknowledge Fundação para a Ciência e a Tecnologia (FCT), through IDMEC, under LAETA, project UIDB/50022/2020. A. Suleman acknowledges also the NSERC Canada Research Chair Program.

REFERENCES

- [1] Ö. Cevdet, E. ÖZBEK, and S. Ekici. A review on applications and effects of morphing wing technology on uavs. *International Journal of Aviation Science and Technology*, 1(01):30–40, 2021.
- [2] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in aerospace sciences*, 41(1): 1–28, 2005.
- [3] X. Yan, J. Zhu, M. Kuang, and X. Wang. Aerodynamic shape optimization using a novel optimizer based on machine learning techniques. *Aerospace Science and Technology*, 86:826–835, 2019.
- [4] B. M. Kulfan. Universal parametric geometry representation method. *Journal of aircraft*, 45(1):142–158, 2008.

- [5] J. Samareh. Aerodynamic shape optimization based on free-form deformation. In *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 4630, 2004.
- [6] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, 1986.
- [7] J. A. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA journal*, 39(5):877–884, 2001.
- [8] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the navier–stokes equations. *Theoretical and computational fluid dynamics*, 10(1): 213–237, 1998.
- [9] M. Lab. Oerview of mach-aero. <https://coderwall.com/p/wntyia/how-to-cite-a-website-in-latex>.
- [10] J. Li, X. Du, and J. R. Martins. Machine learning in aerodynamic shape optimization. *Progress in Aerospace Sciences*, 134:100849, 2022.
- [11] R. Yondo, E. Andrés, and E. Valero. A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Progress in aerospace sciences*, 96:23–61, 2018.
- [12] S. Sharma, S. Sharma, and A. Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.
- [13] H. Leung and S. Haykin. The complex backpropagation algorithm. *IEEE Transactions on signal processing*, 39(9):2101–2104, 1991.
- [14] H. Moin, H. Z. I. Khan, S. Mobeen, and J. Riaz. Airfoil’s aerodynamic coefficients prediction using artificial neural network. In *2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 175–182. IEEE, 2022.
- [15] Z. Zhang. Improved adam optimizer for deep neural networks. In *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pages 1–2. Ieee, 2018.
- [16] M. S. Selig. Uiuc airfoil data site. 1996.
- [17] H. Kussner. General airfoil theory. Technical report, 1941.
- [18] N. R. Secco, G. K. Kenway, P. He, C. Mader, and J. R. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4):1151–1168, 2021.
- [19] C. A. Mader, G. K. Kenway, A. Yildirim, and J. R. Martins. Adflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 17(9):508–527, 2020.

- [20] A. Yildirim, G. K. Kenway, C. A. Mader, and J. R. Martins. A jacobian-free approximate newton–krylov startup strategy for rans simulations. *Journal of Computational Physics*, 397:108741, 2019.
- [21] P. Spalart and S. Allmaras. Proc. aiaa 30th aerospace sciences meeting and exhibit, 1992.
- [22] L. M. Milne-Thomson. *Theoretical aerodynamics*. Courier Corporation, 1973.
- [23] C. D. Harris. Nasa supercritical airfoils: A matrix of family-related airfoils. Technical report, 1990.
- [24] J. Li, M. A. Bouhlel, and J. R. Martins. Data-based approach for fast airfoil analysis and optimization. *AIAA Journal*, 57(2):581–596, 2019.
- [25] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.
- [26] L. Prandtl. The resistance law for rough plates. *David W. Taylor Model Basin, Navy Department, Washington DC, Translation 258*, 1955.
- [27] C. Rumpsey. 2dn00: 2d naca 0012 airfoil validation case. URL https://turbmodels.larc.nasa.gov/naca0012_val.html.
- [28] Z. Zlatev, I. Dimov, I. Faragó, and Á. Havasi. *Richardson extrapolation: Practical aspects and applications*, volume 2. Walter de Gruyter GmbH & Co KG, 2017.
- [29] W. Chen, K. Chiu, and M. D. Fuge. Airfoil design parameterization and optimization using bézier generative adversarial networks. *AIAA journal*, 58(11):4723–4735, 2020.



TOPOLOGY OPTIMIZATION OF ORIGAMI STRUCTURES BASED ON CREASE PATTERN AND AXIAL RIGIDITY

Vincenzo Cretella^{1,2*}, Abdolrasoul Sohoul³, Alfonso Pagani¹ and Afzal Suleman^{2,3}

1: Department of Mechanical and Aerospace Engineering
Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
s294797@studenti.polito.it (V.C.), alfonso.pagani@polito.it, <https://www.polito.it>

2: IDMEC, Institute of Mechanical Engineering
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais, No. 1, 1049-001, Lisbon, Portugal
vincenzo.cretella@tecnico.ulisboa.pt, suleman@tecnico.ulisboa.pt, <https://tecnico.ulisboa.pt>

3: Department of Mechanical Engineering, Center for Aerospace Research
University of Victoria
3800 Finnerty Rd, Victoria, BC V8P 5C2, Canada
sohouli@uvic.ca, suleman@uvic.ca, <https://www.uvic.ca>

Abstract. *Origami structures present desirable stowage properties for application in deployable space structures. The present work proposes a design method for origami structures using topology optimization to find the axial rigidity distribution of the trusses and the crease pattern that maximizes the displacement at set locations under prescribed forces and boundary conditions. In the first part, a linear truss method is used to determine small strain and small rotation mechanics of flexible origami, functional to study their behavior at the initiation of folding. Subsequently, a modified nonlinear truss method is implemented to consider large displacement and large rotation mechanics. To carry out the optimization process, constraints on the number of fold lines and on the material distribution are applied. Previous studies on topology optimization of origami structures have considered only folding and bending in their analyses. Here, it is shown that including also the axial rigidity as a design variable leads to new promising origami designs.*

Keywords: Topology optimization; Origami structures; Linear analysis; Nonlinear analysis; Truss method.

1 INTRODUCTION

Space structures must be lightweight, compact and small during transportation to space, and of large size when deployed in orbit. This makes origami the best choice for certain applications, thanks to their capability to be deployed from a $2D$ to a $3D$ configuration; to be changed in scale through folding and unfolding; to fold and unfold without an external actuation in the case of active origami; to dynamically change the shape according to specific design constraints; to be tailored for a specific task and to be stowed in a relatively small volume.

Nevertheless, the application of origami structures in space missions is still limited. This is due to the complexity of the nonlinear dynamics of deployment, of the mechanical systems and the impossibility to be tested in orbit. For this reason, it is often mandatory to use simplified design processes, but that still provide reliable results.

Various kinematic approaches have been applied [1, 2], with the assumption of rigid origami. To improve on this analysis, [3] introduced a truss model to allow for facet deformations using a linear method, hence considering only small strains and small deformations. This method was applied to topology optimization by [4], with the objective of discovering origami crease patterns that maximize displacements at set locations.

In order to take into account large deformations and large rotations, nonlinear models like the ones used in [5, 6] are required. A nonlinear topology optimization approach was proposed by [7], where a procedure to find an origami crease pattern that achieves desired large deformation through folding for a given input force was provided, using the fold stiffness as a design variable and applying a constraint on the number of fold lines.

This work will be divided into two parts. In the first part, a linear topology optimization will be carried out with the objective of improving the optimization method in [4], introducing axial rigidity as a design variable thus allowing each truss to stretch in an optimized way. Subsequently, the same process will be applied to nonlinear topology optimization, referring to [7]. Although the complexity of the problem is slightly increased with respect to these works, this modification enables to obtain promising results for topology optimization of origami structures, since new configurations and greater actuation than using only the fold stiffness as a design variable are obtained.

Two gradient-based methods are used to perform the optimization: the Method of Moving Asymptotes (MMA) [8] and the Sequential Quadratic Programming (SQP) [9]. Also, the axial rigidity distribution is optimized through a modified Solid Isotropic Material with Penalization (SIMP) method used in [10].

Lastly, it must be noted that the thickness of the origami structure is not considered in this work, but it has been studied in [11].

2 LINEAR TOPOLOGY OPTIMIZATION

2.1 Linear truss model

In this Section, a linear truss method is used to study small displacement and small rotations in flexible origami structures. Fig. 1 shows a general starting configuration of an origami with this method. Each vertex in the folded sheet, i.e. each node in the starting configuration (black points in Fig. 1) is represented by a pin-joint, while each fold line and external segment (grey lines in Fig. 1) is represented by a truss element.

Following the method in [12], modeling the folding patterns as a pin-jointed framework

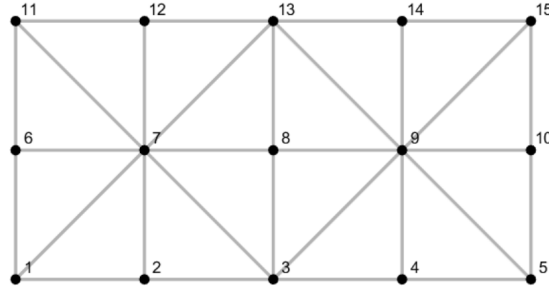


Figure 1: Reference grid of an origami sheet in the linear truss method.

allows the use of established structural engineering methods to study the mechanics of the folded sheet. The mechanical properties of the pin-joint framework should be described by introducing a stiffness formulation, that relates the nodal displacements \mathbf{u} with the applied nodal forces \mathbf{F} through the material stiffness matrix \mathbf{K} :

$$\mathbf{K}\mathbf{u} = \mathbf{F}. \quad (1)$$

The matrix \mathbf{K} can be obtained from:

$$\mathbf{K} = \mathbf{K}_J + \mathbf{K}_T, \quad (2)$$

where \mathbf{K}_J and \mathbf{K}_T are the stiffness matrices obtained from the fold constraint [13] and the truss model respectively.

In a *linear analysis* the stiffness matrix remains constant and independent of \mathbf{u} . This has the advantage of having a low computational cost, but the drawback is that the applied loads are required to be small, leading to small deformations. Even though origami folds usually generate large deformation, this analysis can determine the optimal topology for the initiation of folding.

2.2 Optimization Methods

2.2.1 Fold Constraint

The folding mechanism is derived from constraints on the relationship between the dihedral angles (ϕ) and the nodal coordinates (\mathbf{X}):

$$\mathbf{J} = \frac{d\boldsymbol{\theta}}{d\mathbf{X}}. \quad (3)$$

This Jacobian can be evaluated considering each dihedral angle ϕ relative to adjacent faces, like in Fig. 2, that is:

$$\phi = \sin^{-1} \left[\frac{\mathbf{v}_{12} \times (\mathbf{v}_{14} \times \mathbf{v}_{12}) \cdot (\mathbf{v}_{12} \times \mathbf{v}_{13'})}{\sin \gamma \sin \beta \|\mathbf{v}_{12}\|^3 \|\mathbf{v}_{13'}\| \|\mathbf{v}_{14}\|} \right] \quad (4)$$

where \mathbf{v}_{ij} is the vector from node i to j .

For each fold line k , the fold stiffness K_J^k can be computed as:

$$K_J^k = (J^k)^T G_k J^k \quad k = 1, \dots, N_f \quad (5)$$

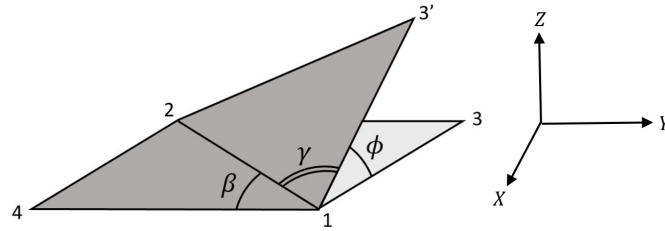


Figure 2: Scheme of the origami element in the linear truss model.

where N_f is the number of fold lines and G_k is the fold stiffness relative to folding, hence it is equal to zero in the trusses at the boundaries.

[4] proposed a design method for origami structures using topology optimization considering the fold stiffness G as a design element, to discover the *optimized crease pattern*, i.e. the configuration of fold lines in the origami that allows to achieve the greatest actuation. Only the fold lines required to obtain this actuation are revealed after the analysis, considering a constraint on the maximum allowable number of fold lines that can be active (i.e. soft folds with small values of G that allow folding), hiding the remaining inactive fold lines (i.e. stiff folds with large values of G that do not allow folding). Even if the inactive fold lines are not shown, they indicate that the two adjacent facets are considered as one, therefore, if their angle ϕ is not zero, it is possible to observe a coarse representation of facet bending. The fold stiffness is represented as the following function

$$G_k = 10^{\alpha_0 + \alpha_k(\alpha_1 - \alpha_0)} \quad (6)$$

which is continuous and differentiable to allow to use gradient-based optimization algorithms. The design variable $\alpha_k \in [0, 1]$, i.e. the fold stiffness exponent associated with folding, is considered, while α_0 and α_1 are constants and G_k that takes values from $G_{soft} = 10^{\alpha_0}$ to $G_{stiff} = 10^{\alpha_1}$.

2.2.2 Truss Model

The stiffness matrix from the truss model is the sum of every elemental stiffness matrix of the truss elements k_e^j , as follows:

$$\mathbf{K}_T = \sum_{j=1}^{N_e} k_e^j \quad (7)$$

where k_e^j is a function of the axial rigidity EA_j .

In this work, the optimization method in [4] is improved to consider also the axial rigidity EA as a design element, discovering both the optimized crease pattern and the *axial rigidity distribution*, i.e. the origami configuration in terms of the axial rigidity of each truss that allows to achieve the greatest actuation. To obtain the axial rigidity, a density-based approach to topology optimization can be considered, similar to the modified SIMP approach used in [10]. The design domain is discretized by finite elements with an assigned axial rigidity density β :

$$EA_j = EA_{min} + \beta_j^p (EA_0 - EA_{min}) \quad j = 1, \dots, N_e \quad (8)$$

where N_e is the number of truss elements, EA_0 the axial solid rigidity of the material, EA_{min} the axial void rigidity assigned to avoid singularity and p a penalization factor that ensures black-and-white solutions. Considering the axial rigidity instead of the Young's Modulus E (like in [10]) has manufacturing advantages for the trusses in the structure since it is possible to decide whether to act on their Young's Modulus (changing material) or on their area (keeping the same material).

2.3 Optimization Framework

The aim of this optimization is to find the axial rigidity distribution of the trusses and the origami crease pattern that maximizes the displacement at set locations, achieving the desired deformations through folding and stretching for a given input force. The *optimization problem* is the following:

<p>Find $\mathbf{x} = \alpha_1, \dots, \alpha_{N_f}, \beta_1, \dots, \beta_{N_e}$ that</p> <p>Minimize $f = -\mathbf{c}^T \mathbf{u}$</p> <p>Subject to</p> $g_1 = v_{0,1} - \frac{1}{N_f} \sum_{i=1}^{N_f} x_i \leq 0; \quad g_2 = v_{0,2} - \frac{1}{N_e} \sum_{i=N_f+1}^{N_f+N_e} x_i \leq 0; \quad (9)$ $0 \leq x_i \leq 1; \quad \forall i = 1, \dots, N_f + N_e;$ $\mathbf{K}\mathbf{u} = \mathbf{F}.$
--

f is the objective function, hence minimizing f means maximizing the displacements along a direction, where \mathbf{c} is a vector that selects the displacements relative to the problem, taking on values of either 1, -1 or 0 to indicate the associated direction of optimal actuation. $g_1 \leq 0$ and $g_2 \leq 0$ are the two inequality constraints:

- g_1 limits the number of foldlines to meet design requirements given a total of N_f available lines, while $v_{0,1}$ limits the fold stiffness used in the structure. The parameter $l_0 = 1 - v_{0,1}$ is considered as the allowable foldline fraction.
- g_2 limits the number of trusses that are allowed to have an axial solid rigidity EA_0 to meet design requirements given a total of N_e available lines, while $v_{0,2}$ limits the axial rigidity used in the structure. The parameter $m_0 = 1 - v_{0,2}$ is considered as the allowable material fraction of the structure.

The value of the design variables is limited to take on values between 0 and 1 from the second constraint. The final constraint is the governing equation of the system. An overview of the algorithm is shown in Fig. 3.

To use a gradient-based optimization algorithm, a *sensitivity analysis* must be carried out. Therefore, the derivative of the stiffness matrix is found as:

$$\frac{\partial \mathbf{K}}{\partial \mathbf{x}} = \frac{\partial \mathbf{K}_J}{\partial \boldsymbol{\alpha}} + \frac{\partial \mathbf{K}_T}{\partial \boldsymbol{\beta}} \quad (10)$$

where $\boldsymbol{\alpha} = x_1, \dots, x_{N_f}$ and $\boldsymbol{\beta} = x_{N_f+1}, \dots, x_{N_f+N_e}$. The first term in Eq. (10) comes from

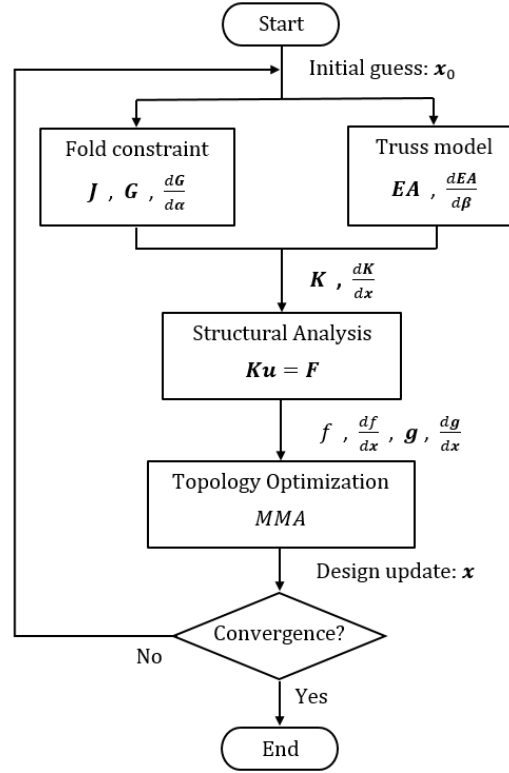


Figure 3: Flow chart of the linear optimization process.

the fold constraint

$$\frac{\partial \mathbf{K}_J}{\partial \alpha_k} = \mathbf{J}^T \frac{\partial G_k}{\partial \alpha_k} \mathbf{J} = \mathbf{J}^T [(\alpha_1 - \alpha_0) 10^{\alpha_0 + \alpha_k (\alpha_1 - \alpha_0)} \log(10)] \mathbf{J} \quad (11)$$

while the second one from the truss model, using a chain rule

$$\frac{\partial \mathbf{K}_T}{\partial \beta_j} = \frac{\partial \mathbf{K}_T}{\partial EA_j} \frac{\partial EA_j}{\partial \beta_j} \quad (12)$$

In Eq. (12), $\partial \mathbf{K}_T / \partial EA_j$ is obtained from the gradient of the elemental stiffness matrix of the truss elements and

$$\frac{\partial EA_j}{\partial \beta_j} = p \beta_j^{(p-1)} (EA_0 - EA_{min}) \quad (13)$$

The gradient is obtained from the following adjoint method:

$$\mathbf{K} \left(\frac{\partial \mathbf{u}}{\partial x_i} \right) = - \left(\frac{\partial \mathbf{K}}{\partial x_i} \right) \mathbf{u} \quad (14)$$

Solving Eq. (14) with \mathbf{u} and \mathbf{K} obtained from Eq.s (1) and (2) respectively, $\partial \mathbf{u} / \partial x_i$ is found, and the gradients df / dx_i can be computed as:

$$\frac{df}{dx_i} = - \left(\frac{\partial \mathbf{u}}{\partial x_i} \right)^T \mathbf{c} \quad (15)$$

Lastly, the optimization problem can be solved using the Method of Moving Asymptotes (MMA).

2.4 Numerical Example

In the following example, the stiffness coefficients are set to $\alpha_0 = 2$ and $\alpha_1 = 6$. Moreover, the applied forces must be small enough to remain in a linear regime, thus obtaining displacements within a 10% range of the length of the structure.

The problem is studied first using the Origami Mechanism Topology Optimizer (OMTO) in [14] that uses only the fold stiffness exponent (α) as design variable and the allowable fold line fraction (l_0) as constraint, then with a modified OMTO, which uses the method described in this Section, that also considers the axial rigidity density (β) as design variable and the allowable material fraction (m_0) as constraint. The objective function varies according to two parameters, introduced in this Section:

- l_0 : constraint on the number of allowable fold lines. If $l_0 = 0$ no fold line of the starting configuration is active, while if $l_0 = 1$ all the fold lines are active;
- m_0 : constraint on the allowable material fraction. If $m_0 = 0$ every truss member has $EA = EA_0$, while if $m_0 = 1$ every truss member has $EA = EA_{min}$.

In [4], l_0 alone is sufficient to carry out the analyses, while in this method also m_0 needs to be used since there are now two sets of design variables. However, when $m_0 = 0$ (or sufficiently close to 0 to avoid matrix singularities), the axial rigidity is kept equal among each truss, hence the second set of design variables is not considered and the same results as using only one set of design variables are obtained.

Lastly, the dashed and dotted-dashed lines in the optimized crease pattern indicate the mountain and valley folds of the origami, while the black and magenta lines in the optimized axial rigidity distribution indicate that the truss members have an axial rigidity $EA = EA_0 = 10^8 Pa \cdot m^2$ or $EA = EA_{min} = 10^4 Pa \cdot m^2$ respectively.

2.4.1 Miura-Ori fold pattern

In this example, the trusses in the starting configuration are displayed in Fig. 4a, and the sheet is folded like in Fig. 4b.

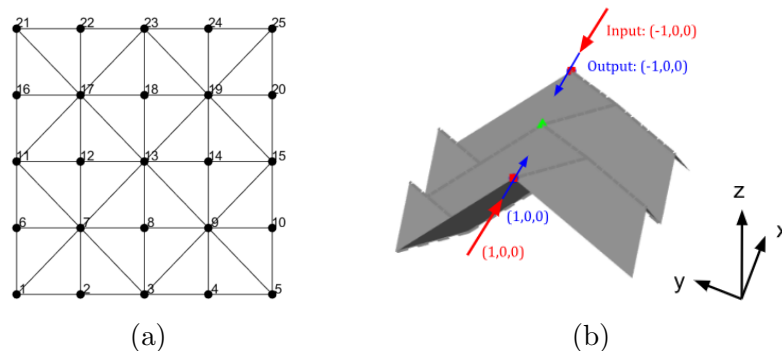


Figure 4: Starting configuration of the Miura-Ori pattern. (a) Reference grid; (b) Load and boundary conditions ($L_x = 1.0 m$, $L_y = 1.0 m$, $F = 600 N$).

This configuration is called *Miura-Ori*, a famous fold pattern designed for deployable structures in space, used in the *Space Flyer Unit (SFU)* [15] in 1995. The green triangles are the fixed nodes, the red squares are the applied loads \mathbf{F} and the blue dots are the nodes

where the displacements \mathbf{u} need to be evaluated. This problem has 96 design variables ($N_f = 40$ relative to α , $N_e = 56$ relative to β).

Tab. 1 displays that, maintaining l_0 constant and equal to 0.50 and increasing m_0 (from 0.30 to 0.50), more trusses are allowed to have a minimum axial rigidity. Therefore, the structure becomes more flexible and can sustain the targeted larger displacement, for the cost of higher design complexity and iteration steps (Fig. 5).

l_0 [-]	m_0 [-]	Optimized Crease Pattern	Axial Rigidity distribution	f [mm]
1 design variable (α)				
0.50	0			-38.00
2 design variables (α, β)				
0.50	0.30			-76.80
0.50	0.40			-87.20
0.50	0.50			-94.00

Table 1: Objective function and final configurations of the Miura-Ori pattern.

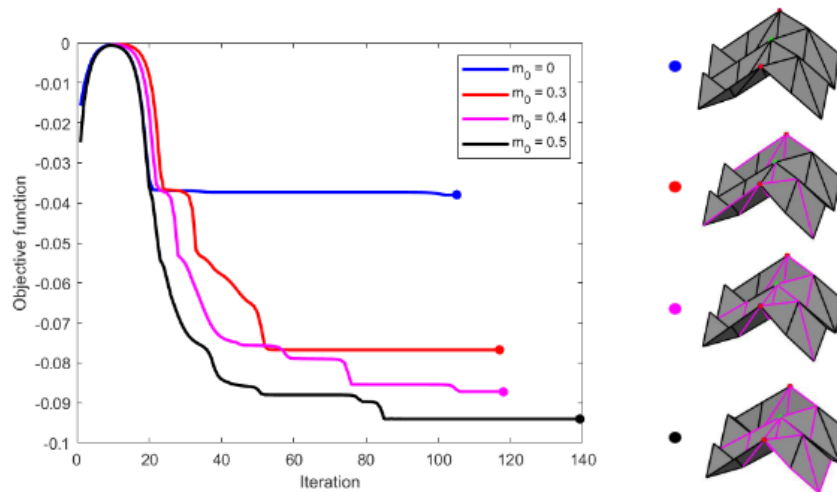


Figure 5: Objective function over iteration for different m_0 ($l_0 = 0.50$).

3 NONLINEAR TOPOLOGY OPTIMIZATION

3.1 Nonlinear truss model

In this Section, the modified nonlinear truss model introduced in [5] based on [16] is considered to study large displacements and large rotations, while optimizing efficiency and accuracy to face the increased difficulty given by the nonlinearity of the problem. This model presents a torsional spring around the truss element so that the fold stiffness between two adjacent faces can be considered.

A scheme of the origami element of the nonlinear truss model is shown in Fig. 6a, where $\mathbf{F}_i = (F_{X_i}, F_{Y_i}, F_{Z_i})$ ($i = 1, \dots, 4$) is the nodal force applied in one of the nodes, ζ is the nondimensional integration length dimension along the axial direction of the truss and $\mathbf{X}_i = (X_i, Y_i, Z_i)$ is the global position of the i -th node. $\mathbf{X}_{tri} = \{\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4\}$ is the set of global coordinates of the local nodes required to define the fold angle ϕ . Lastly, the black line that connects nodes 1 and 2 is the fold line, a truss element that contains an axial strain term with axial rigidity EA and a bending energy term with fold stiffness G . The reference grid of the described model is shown in Fig. 6b, where the black lines represent the trusses, enumerated in blue at the boundaries and in red in the interior (fold lines), while the nodes are enumerated in black.

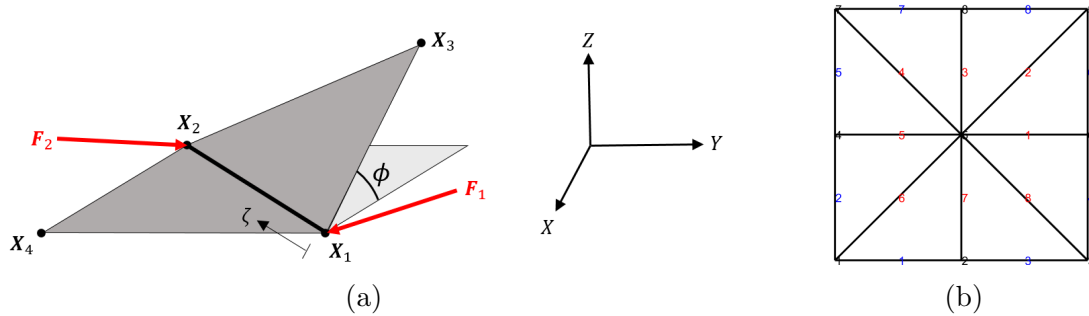


Figure 6: Nonlinear truss model. (a) Scheme of the origami element; (b) Reference grid.

3.2 Principle of Minimum Energy

To study the nonlinear problem, the principle of minimum energy associated with the single element in Fig. 6a is presented, and it can be extended to a generic origami truss structure through assembly. The total energy (Π) is defined as

$$\Pi = U_t - P, \quad (16)$$

where U_t is the potential energy and P is the external energy. The potential energy is

$$U_t = t_0 \int_0^1 \left[\frac{EA}{2} \varepsilon(\mathbf{X}_1, \mathbf{X}_2)^2 + \frac{G}{2} \tilde{\phi}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4)^2 \right] d\zeta, \quad (17)$$

The axial strain ε quantifies the axial deformation of the truss element

$$\varepsilon = \frac{1}{t_0} (|\mathbf{X}_2 - \mathbf{X}_1| - t_0) = \frac{1}{t_0} (\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2} - t_0), \quad (18)$$

while $\tilde{\phi}$ quantifies the torsional deformation of the spring

$$\tilde{\phi}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4) = \phi(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4) - \phi_0, \quad (19)$$

where t_0 and ϕ_0 are the length and the angle of the structure in their undeformed state, while ϕ is a nonlinear function that represents the current fold angle. A linear constitutive model is assumed in Eq.s (18) and (19), however ε and $\tilde{\phi}$ depend on the global position of the nodes, hence they are subjected to the geometric nonlinearities of the motion. Therefore, a penalty function is introduced to avoid singularities and to enforce that the two facets are kept in contact. The external energy is

$$P = F_{X_1}(X_1 - X_1^0) + F_{X_2}(X_2 - X_2^0) + F_{Y_1}(Y_1 - Y_1^0) + F_{Y_2}(Y_2 - Y_2^0) + F_{Z_1}(Z_1 - Z_1^0) + F_{Z_2}(Z_2 - Z_2^0) \quad (20)$$

where for example X_1^0 is the node location along the X axis of node 1 in the undeformed state. The displacements of the nodes are $\mathbf{u}_1 = \mathbf{X}_1 - \mathbf{X}_1^0$ and $\mathbf{u}_2 = \mathbf{X}_2 - \mathbf{X}_2^0$.

Lastly, the principle of minimum energy states that the equilibrium state of the structure is reached when the following equation is satisfied

$$\frac{\partial \Pi}{\partial X_l} = t_0 \int_0^1 \left[\frac{EA}{2} \frac{d\varepsilon}{dX_l} + \left(Gp(\phi)\tilde{\phi} + G \frac{\tilde{\phi}^2}{2} \frac{\partial p(\phi)}{\partial \phi} \right) \frac{\partial \phi}{\partial X_l} \right] d\zeta - F_{X_l} = 0. \quad (21)$$

To numerically solve the system of nonlinear equations introduced by the principle of minimum energy, the *Newton-Rapson method* is applied, to linearize Eq. (21) through a Taylor's series expansion

$$R_l(X_{tri}) = \frac{\partial \Pi}{\partial X_l} \approx R_l(X_{tri}^0) + \nabla R_l(X_{tri}^0) \Delta X_{tri} = 0, \quad (22)$$

where $R_l(X_{tri})$ is the residual and the term $\nabla R_l(X_{tri}^0)$ is the tangent stiffness K_{lm} , where the indices l and m iterate through all the components of \mathbf{X}_{tri} . Eq. (22) is solved iteratively until the equilibrium is reached (within a prescribed tolerance).

Furthermore, at each iteration, the residual R_l and the tangent stiffness K_{lm} are computed both for the trusses at the interior (fold lines) and for the trusses at the boundary (with $G = 0$). The global residual and tangent stiffness are then computed:

$$R_l = R_l^F + R_l^B; \quad (23)$$

$$K_{lm} = K_{lm}^F + K_{lm}^B. \quad (24)$$

Lastly, increment loads are considered in this work in order to follow the complex nonlinear loading behavior of origami structures subjected to large displacements.

3.3 Optimization Framework

Following the linear method described in the previous Section, the optimization method introduced by [7] is improved including also the axial rigidity EA as a design element through the SIMP method. However, due to the nonlinearity of the problems, this method does not provide only black-and-white results, leading to some in-between values. The

optimization problem is the following:

<p>Find $\mathbf{x} = \alpha_1, \dots, \alpha_{N_f}, \beta_1, \dots, \beta_{N_e}$ that</p> <p>Minimize $f = -\mathbf{c}^T \mathbf{u}$</p> <p>Subject to</p> $g_1 = v_{0,1} - \frac{1}{N_f} \sum_{i=1}^{N_f} x_i \leq 0; \quad g_2 = v_{0,2} - \frac{1}{N_e} \sum_{i=N_f+1}^{N_f+N_e} x_i \leq 0; \quad (25)$ $0 \leq x_i \leq 1; \quad \forall i = 1, \dots, N_f + N_e;$ $R_l(\mathbf{X}) = 0; \quad u_l = X_l - X_l^0; \quad l = 1, \dots, 3N_n.$
--

The difference with the linear optimization in the previous Section lies in the governing equation of the system, i.e. the linearization of the residual to solve the system of nonlinear equations introduced in Eq. (21). Lastly, N_n is the number of nodes and $3N_n$ is the number of degrees of freedom. An overview of the algorithm is shown in Fig. 7.

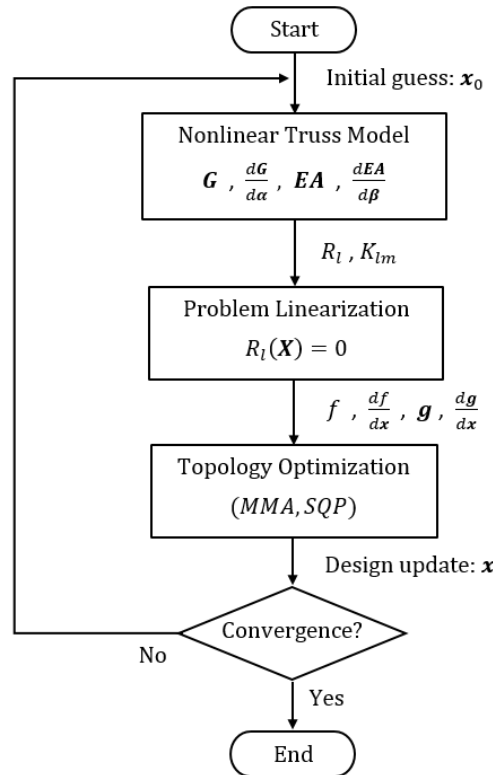


Figure 7: Flow chart of the linear optimization process.

Two gradient-based optimization algorithms are used to perform the analyses: the Method of Moving Asymptotes (MMA) and the Sequential Quadratic Programming (SQP).

To carry out the sensitivity analysis, the derivative of the objective function with respect to the design variables is obtained as follows:

$$\frac{df}{dx_i} = \frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial X_m} \frac{dX_m}{dx_i}, \quad (26)$$

where $\partial f/\partial x_i = 0$ since f does not explicitly depend on x_i , $\partial f/\partial X_m$ is found by directly differentiating the objective function f , while dX_m/dx_i is determined solving the following system of equations:

$$K_{lm} \frac{dX_m}{dx_i} = -\frac{\partial R_l}{\partial x_i}. \quad (27)$$

The vector $\partial R_l/\partial x_i \in [3N_n, N_f + N_e]$ has the first N_f columns equal to

$$\frac{\partial R_l}{\partial \alpha_k} = l_0 \int_0^1 \left[\frac{\partial G_k}{\partial \alpha_k} p(\phi) \tilde{\phi} + \frac{\partial G_k}{\partial \alpha_k} \frac{\tilde{\phi}^2}{2} \frac{\partial p(\phi)}{\partial \phi} \right] \frac{\partial \phi}{\partial X_l} d\zeta, \quad (28)$$

where

$$\frac{\partial G_k}{\partial \alpha_k} = (\alpha_1 - \alpha_0) 10^{\alpha_0 + \alpha_k(\alpha_1 - \alpha_0)} \log(10), \quad (29)$$

while the remaining N_e columns are equal to

$$\frac{\partial R_l}{\partial \beta_j} = l_0 \int_0^1 \frac{1}{2} \frac{\partial EA_j}{\partial \beta_j} \frac{d\varepsilon}{dX_l} d\zeta, \quad (30)$$

where

$$\frac{\partial EA_j}{\partial \beta_j} = p\beta_j^{(p-1)}(EA_0 - EA_{min}). \quad (31)$$

3.4 Numerical Examples

To assess the capability of the nonlinear truss model with the introduction of axial rigidity as a design variable, two different starting configurations of well-known origami actuator designs (“Chomper” and “Square Twist” patterns) are studied. The optimization is evaluated through comparisons with the optimizer in [17], where the design variable is only the fold stiffness.

Furthermore, the dashed lines in the optimized crease pattern indicate the folds that remain after the optimization (soft folds with $G = G_{soft}$). Lastly, the color of the trusses in the axial rigidity distribution is given in a scale of gray, where white and black indicate an axial rigidity $EA = EA_{min}$ or $EA = EA_0$ respectively. However, a “projected” axial rigidity distribution is presented, where the black and magenta lines indicate that the truss members have an axial rigidity with a relative error within 10% close to $EA = EA_0$ or $EA = EA_{min}$ respectively, while greater than 10% for the gray lines.

3.4.1 Chomper fold pattern

The first example considers the *Chomper* fold pattern, which has been applied as a gripping mechanism for surgical tools [18], miniature robots [19], soft robotics [20] and deployable structures. The problem has 48 design variables ($N_f = 18$ relative to α , $N_e = 30$ relative to β). Fig. 8a shows the reference grid of the structure, while Fig. 8b shows the three-dimensional representation of the starting configuration with the load and boundary conditions.

In this example, the following parameters will be considered: $EA = 10^7 Pa \cdot m^2$ for one design variable; $EA_{min} = 10^5 Pa \cdot m^2$ and $EA_0 = 10^7 Pa \cdot m^2$ for two design variables; $l_0 = 0.44$ (equivalent to 6 active fold lines allowed); $G_{stiff}/G_{soft} = 10^3$. To demonstrate

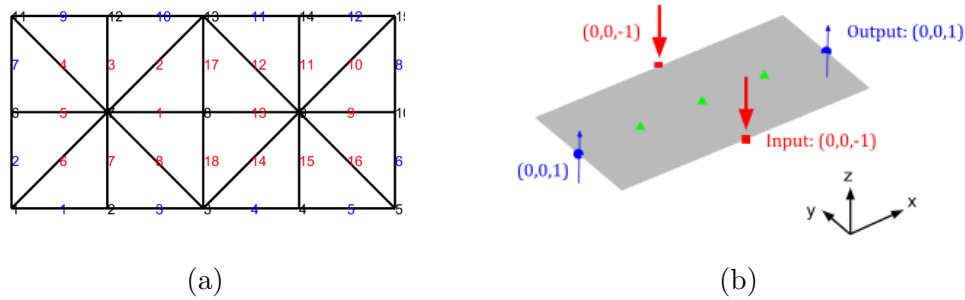


Figure 8: Starting configuration of the Chomper pattern. (a) Reference grid; (b) Load and boundary conditions ($L_x = 0.2 m$, $L_y = 0.2 m$, $F = 10\ 000 N$).

the effect that material properties have on the optimal actuation motion, two sets of material properties will be considered, varying G_{soft} and G_{stiff} :

1. Case 1: $EA/G_{stiff} = 10^1$ (or $EA_0/G_{stiff} = 10^1$ for two design variables);
2. Case 2: $EA/G_{stiff} = 10^3$ (or $EA_0/G_{stiff} = 10^3$ for two design variables).

For each case, different values of the constraint m_0 will be used to study the effect of the addition of the second set of design variables.

Case 1

Tab. 2 shows the values of the objective functions and the final configurations of the Chomper problem for Case 1 obtained with the MMA method.

l_0 [-]	m_0 [-]	Optimized Crease Pattern	Projected Axial Rigidity distribution	f [mm]
1 design variable (α)				
0.44	0			-52.92
2 design variables (α, β)				
0.44	0.20			-88.30
0.44	0.50			-92.84
0.44	0.80			-92.92

Table 2: Objective function and final configurations of the Chomper pattern, Case 1.

All the configurations in Tab. 2 discover the Chomper fold pattern. Like in the linear case, if m_0 is increased the structure becomes more flexible and can sustain a larger actuation than the one in [7], but with more iteration steps, as shown in Fig. 9.

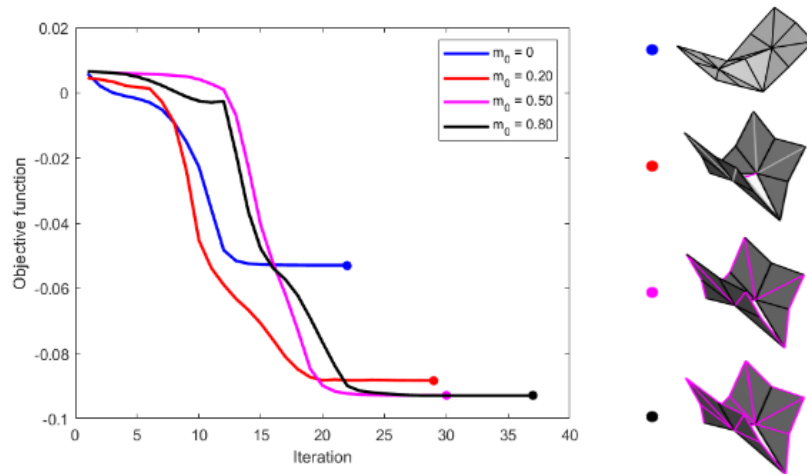


Figure 9: Case 1, objective function over iteration for different m_0 ($l_0 = 0.44$).

The small ratio considered in Case 1 between the axial rigidity of the truss element and a stiff fold ($EA/G_{stiff} = 10^1$) favors facet stretching over facet bending. Therefore, the structure appears to be slightly stretched or compressed along some trusses.

Lastly, since the load condition and l_0 are kept constant during this analysis, the optimized crease pattern is the same for the considered material fractions.

Case 2

Tab. 3 shows the values of the objective functions and the final configurations of the Chomper problem for Case 2 obtained with the MMA method.

l_0 [-]	m_0 [-]	Optimized Crease Pattern	Projected Axial Rigidity distribution	f [mm]
1 design variable (α)				
0.44	0			-9.0e-5
2 design variables (α, β)				
0.44	0.20			-14.23
0.44	0.50			-18.79
0.44	0.80			-54.42

Table 3: Objective function and final configurations of the Chomper pattern, Case 2.

The ratio considered in Case 2 between the axial rigidity of the truss element and a stiff fold ($EA/G_{stiff} = 10^3$) is larger than the one considered in Case 1, hence facet bending is favored over facet stretching. However, the gradient-based optimization using one set

of design variables fails to discover an origami configuration that leads to positive vertical actuation in the output nodes. Fig. 10 shows that in this case the problem does not converge and the iterative process is stopped at 100 iterations, resulting in an objective function value close to zero.

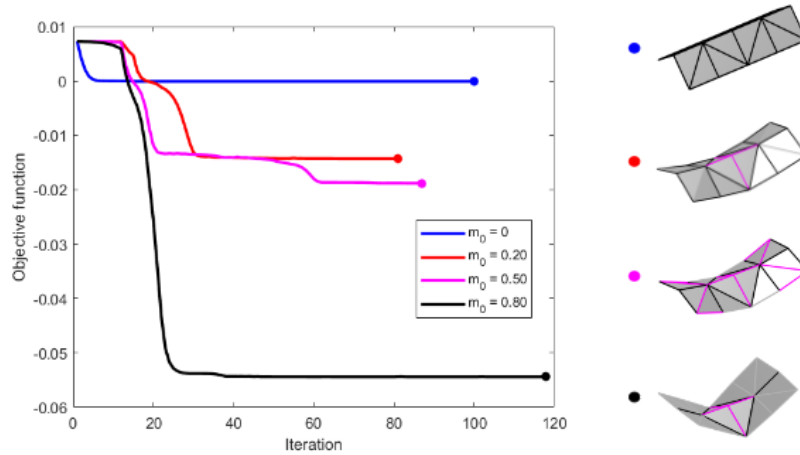


Figure 10: Case 2, objective function over iteration for different m_0 ($l_0 = 0.44$).

Nevertheless, as shown in Tab. 3, using two sets of design variables it is possible to observe a vertical actuation of the output nodes and the bending of the structure. Furthermore, if m_0 is increased, more trusses are allowed to have a minimum axial rigidity, thus the final configuration tends toward that of Case 1. Therefore, for $m_0 = 0.8$, the Chomper fold pattern is discovered, obtaining a larger actuation than with lower m_0 values, but for the cost of more iteration steps, as shown in Fig. 10.

Tab. 4 shows the real axial rigidity distribution of the structure for Case 1 and 2.

l_0 [-]	m_0 [-]	Axial rigidity distribution		Colorbar
		Case 1	Case 2	
0.44	0.20			
0.44	0.50			
0.44	0.80			

Table 4: Axial rigidity distribution of Case 1 and Case 2 of the Chomper pattern.

Lastly, the analysis is repeated with the SQP method, obtaining the same final configurations. As it can be observed from Fig. 11, both optimization methods converge to the same results, but the SQP method needs more iterations to converge due to its higher computational cost, as described in [21].

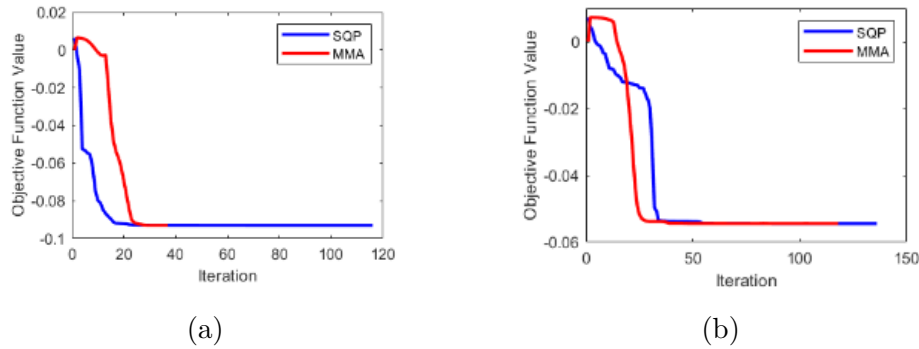


Figure 11: Objective function over iteration with SQP and MMA methods of the Chomper pattern, for $l_0 = 0.44$, $m_0 = 0.80$. (a) Case 1; (b) Case 2.

3.4.2 Square Twist fold pattern

The second example considers the *Square Twist* fold pattern, with 384 design variables (number of fold lines $N_f = 176$, total number of trusses $N_e = 208$). Fig. 12a shows the reference grid of the structure, while Fig. 12b shows the three-dimensional representation of the starting configuration with the load and boundary conditions.

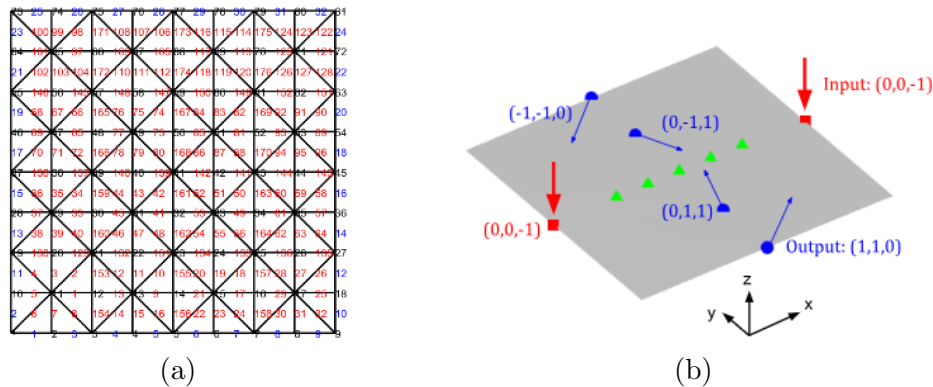


Figure 12: Starting configuration of the Square Twist pattern. (a) Reference grid; (b) Load and boundary conditions ($L_x = 0.4\text{ m}$, $L_y = 0.4\text{ m}$, $F = 10\,000\text{ N}$).

The loads and boundary conditions in Fig. 12b should result in an inward folding followed by a twisting motion to achieve a flat folded configuration. In this example, the following parameters will be considered: $EA = 10^8\text{ Pa} \cdot \text{m}^2$ for one design variable; $EA_{min} = 10^6\text{ Pa} \cdot \text{m}^2$ and $EA_0 = 10^8\text{ Pa} \cdot \text{m}^2$ for two design variables; $l_0 = 0.20$ (equivalent to 32 active fold lines allowed); $m_0 = 0.50$ (104 trusses are allowed to have axial rigidity $EA = EA_{min}$); $EA/G_{stiff} = 10^4$ (or $EA_0/G_{stiff} = 10^4$ for two design variables); $G_{stiff}/G_{soft} = 10^3$. The number of design variables is increased with respect to the previous examples, thus the problem is now more complex and the convergence rate of the algorithm is significantly slowed.

Again, the optimization was carried out both with MMA and SQP methods. However, the objective function obtained with MMA oscillates without reaching convergence, probably due to the inability of the method to handle complex designs. Therefore, only the results obtained with the SQP method will be shown as follows.

Tab. 5 shows the objective functions and the final configurations obtained for one and two design variables.

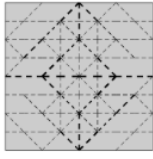
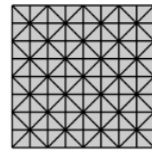
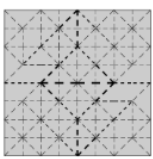
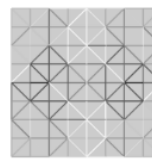
l_0 [-]	m_0 [-]	Optimized Crease Pattern	Axial Rigidity distribution	f [mm]
1 design variable (α)				
0.20	0			- 468.6
2 design variables (α, β)				
0.20	0.50			- 489.9

Table 5: Objective function and final configurations of the Square Twist pattern.

As it can be noted, the optimized crease patterns in Tab. 5 do not satisfy the constraint on the allowable number of fold lines, likely due to numerical instability phenomena during the optimization process related to the nonlinearity of the problem. Therefore, thresholding of the results is required and implemented as follows:

- for one design variable, the 32 lowest values of α (corresponding to $l_0 = 0.2$) are set equal to 0 (soft active folds) and the remaining values to 1 (stiff inactive folds);
- for two design variables, the 32 lowest values of α and the 104 lowest values of β (corresponding to $m_0 = 0.5$) are set equal to 0 and the remaining values to 1.

The results obtained after the thresholding are displayed in Tab. 6.

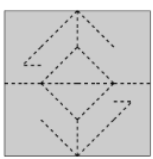
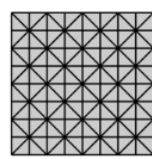
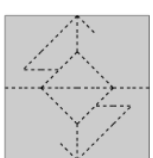
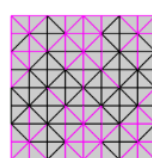
l_0 [-]	m_0 [-]	Thresholded Optimized Crease Pattern	Thresholded Axial Rigidity distribution	f [mm]
1 design variable (α)				
0.20	0			- 280.8
2 design variables (α, β)				
0.20	0.50			- 336.3

Table 6: Thresholded results of the Square Twist pattern.

The actuation has decreased since a smaller number of active fold lines is registered, hence the structure is allowed to fold along fewer fold lines. Also, the two optimized crease patterns do not match like in the previous example. However, under the same conditions,

adding the second set of design variables, the new configuration leads to an increase of about 20% in the displacement. Therefore, also this example shows how adding axial rigidity as a design variable can lead to new promising configurations capable of greater actuation.

Nevertheless, a flat configuration is not achieved using gradient-based optimizations. Therefore, further analyses are required to study the effect of the described method on a topology optimization based on non-gradient-based optimization, like the Genetic Algorithm (GA).

4 CONCLUSIONS

The present work describes a design method for origami structures using topology optimization. A truss model is utilized for linear and nonlinear analysis, proving to be a simple yet effective method to find new promising origami designs.

In the first part, a linear analysis is carried out, providing a low computational cost model effective in analyzing origami structures at the initiation of folding. Subsequently, a nonlinear analysis model is utilized to consider large displacements and large rotations while optimizing the efficiency and accuracy of the results.

Previous works on topology optimization of origami structures have only considered fold stiffness relative to the bending energy as a design variable, discovering origami fold patterns that maximize the displacement at set locations. In this work, also axial rigidity is considered a design variable, improving the optimization by allowing each truss to stretch in an optimized way.

The capability of the method is assessed through three test cases considering well-known origami designs, one for the linear and two for the nonlinear analysis. The results obtained using only fold stiffness as a design variable are compared to the ones obtained using also axial rigidity with the described method.

The study demonstrates that, if the axial rigidity is allowed to vary among the trusses of the structure, new configurations can be discovered. Moreover, with a variable axial rigidity, the structure is more flexible and can sustain a targeted larger actuation.

In this study, only gradient-based optimizations are applied. However, as shown in the Square Twist example, this kind of optimization does not always lead to optimal solutions for complex origami configurations involving highly nonlinear motions. Therefore, for future studies, the application of the described method to non-gradient-based optimizations needs to be assessed.

Lastly, origami structures usually present repeating patterns and exhibit equilibrium bifurcations off the flat state. Hence, the method can be generalized including periodic boundary conditions and modal analysis in the optimization process.

ACKNOWLEDGEMENTS

We are grateful to Dr. Kazuko Fuchi from the University of Dayton for her advice and for providing the MATLAB code in [14] and [17].

REFERENCES

- [1] T. Tachi. Simulation of rigid origami. *Origami*, 4(08):175–187, 2009.

- [2] E. A. Peraza Hernandez, D. J. Hartl, and D. C. Lagoudas. Active origami: modeling, design, and applications. *Cham: Springer*, 2018.
- [3] M. Schenk, S. D. Guest, et al. Origami folding: A structural engineering approach. *Origami*, 5:291–304, 2011.
- [4] K. Fuchi, P. R. Buskohl, G. Bazzan, M. F. Durstock, G. W. Reich, R. A. Vaia, and J. J. Joo. Origami actuator design and networking through crease topology optimization. *Journal of Mechanical Design*, 137(9):091401, 2015.
- [5] A. Gillman, K. Fuchi, and P. Buskohl. Truss-based nonlinear mechanical analysis for origami structures exhibiting bifurcation and limit point instabilities. *International Journal of Solids and Structures*, 147:80–93, 2018.
- [6] K. Liu and G. H. Paulino. Nonlinear mechanics of non-rigid origami: an efficient computational approach. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2206):20170348, 2017.
- [7] A. S. Gillman, K. Fuchi, and P. R. Buskohl. Discovering sequenced origami folding through nonlinear mechanics and topology optimization. *Journal of Mechanical Design*, 141(4):041401, 2019.
- [8] K. Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.
- [9] P. E. Gill, W. Murray, and M. H. Wright. *Numerical linear algebra and optimization*. SIAM, 2021.
- [10] E. Andreassen, A. Clausen, M. Schevenels, B. S. Lazarov, and O. Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16, 2011.
- [11] S. A. Zirbel, R. J. Lang, M. W. Thomson, D. A. Sigel, P. E. Walkemeyer, B. P. Trease, S. P. Magleby, and L. L. Howell. Accommodating thickness in origami-based deployable arrays. *Journal of Mechanical Design*, 135(11), 2013.
- [12] M. Schenk and S. D. Guest. Origami folding: A structural engineering approach. In *5OSME, 5th international conference on Origami in Science, Mathematics and Education*. Retrieved from: <http://www.markschenk.com/research/#papers>, 2010.
- [13] T. C. Hull et al. Modelling the folding of paper into three dimensions using affine transformations. *Linear Algebra and its applications*, 348(1-3):273–282, 2002.
- [14] K. Fuchi. Origami mechanism topology optimizer (omto) ver 1.1n, 2015. URL <https://www.mathworks.com/matlabcentral/fileexchange/53037-origami-mechanism-topology-optimizer-omto-ver-1-1n>. MATLAB Central File Exchange, Accessed on May 8, 2023.

- [15] Y. Shibayama, H. Arai, K. Matsui, K. Hama, A. Ushirokawa, M. Natori, K. Takahashi, N. Wakasugi, and T. Anzai. Sfu solar array. In *European Space Power Conference*, volume 2, pages 557–562, 1989.
- [16] M. Greco, F. A. R. Gesualdo, W. S. Venturini, and H. B. Coda. Nonlinear positional formulation for space truss analysis. *Finite elements in analysis and design*, 42(12): 1079–1086, 2006.
- [17] A. Gillman, K. Fuchi, and P. Buskohl. Origami topology optimization w/ nonlinear truss model, 2018. URL <https://www.mathworks.com/matlabcentral/fileexchange/69612-origami-topology-optimization-w-nonlinear-truss-model>. MATLAB Central File Exchange, Retrieved December 5, 2018, Accessed on July 6, 2023.
- [18] B. J. Edmondson, L. A. Bowen, C. L. Grames, S. P. Magleby, L. L. Howell, and T. C. Bateman. Oriceps: Origami-inspired forceps. In *Smart Materials, Adaptive Structures and Intelligent Systems*, volume 56031, page V001T01A027. American Society of Mechanical Engineers, 2013.
- [19] S. Miyashita, S. Guitron, M. Ludersdorfer, C. R. Sung, and D. Rus. An untethered miniature origami robot that self-folds, walks, swims, and degrades. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1490–1496. IEEE, 2015.
- [20] D.-Y. Lee, S.-R. Kim, J.-S. Kim, J.-J. Park, and K.-J. Cho. Origami wheel transformer: A variable-diameter wheel drive robot using an origami structure. *Soft robotics*, 4(2):163–180, 2017.
- [21] M. Fanni, M. Shabara, and M. Alkalla. A comparison between different topology optimization methods. *MEJ. Mansoura Engineering Journal*, 38(4):13–24, 2020.



BUILDING BLOCKS TOWARDS ADVANCED THERMOPLASTIC COMPOSITES FOR SUSTAINABLE AVIATION: INTEGRATION OF MATERIAL, PROCESS AND JOINING, DESIGN, VALIDATION, AND MONITORING

Mehmet Yildiz

Professor and Vice-President Research
Integrated Manufacturing Technologies Research and Application Center
Sabanci University
Orhanli-Tuzla, 34956, Istanbul, Turkey
mehmet.yildiz@sabanciuniv.edu, <https://suimc.sabanciuniv.edu/>

Abstract. *Automated Fiber Placement (AFP) is an automated manufacturing technique that can produce large-scale structural composite parts in a repeatable manner with a high production rate. The AFP technique has been gaining notable attention and importance due to i) the need for producing large-scale/high-volume composite structures such as fuselage and wing skins and ii) enabling simultaneous lay-up and consolidation, thereby leading to a significant reduction in material and labor costs. An AFP machine can have a robotic arm with a fiber placement head or end effector that lays up thermoset, thermoplastic, or dry fiber tows in different orientations, which can be determined in accordance with the loading configurations. Today, mostly carbon fiber-reinforced thermoset-based composite materials are used for manufacturing aerospace structural parts. However, the well-proven shortcomings of thermoset polymers such as limited shelf life, long processing times, relatively low fracture toughness, and poor absorption ability against water and chemicals force the aviation industry to seek for alternative composite material solutions, particularly for new-generation aircraft components. Carbon fiber-reinforced high-performance thermoplastic composites can be regarded as a very competitive alternative to their thermoset-based counterparts due to their following important attributes; high toughness, damage tolerance, unlimited storage life without any need for cold storage, recyclability, high chemical/solvent resistance, and relatively easier processability as they do not entail complex chemical reaction or curing process. Moreover, thermoplastics can readily lend themselves to automated manufacturing processes such as AFP. However, the high-performance thermoplastic composite structures manufactured with the AFP process can only be realized by successful implementations of material selection, process design, joining practices, design, and validation with prototyping for any specific application. Additionally, the fundamental steps of the composite value chain should be followed with continuous inspection considering the service conditions. In this framework, this talk will highlight the steps of the composite value chain for future practices with advanced thermoplastics by addressing the integrated and complementary building blocks of i. material characterization, ii. process design, iii. joining challenges, iv. mechanical characterization with inspections, and v. service life performances. The composite value chain will be evaluated with different case studies. First, the importance of accurate material input to the process design will be pro-*

vided with an improved thermal model considering the degree of intimate contact, degree of bonding, and fiber orientation. Next, the process design practices will be introduced considering the energy efficiency and digital transformation challenges by promoting in-situ consolidation. Moreover, the new opportunities with thermoplastic resin systems for joining technologies will be emphasized in a competitive manner with different welding mechanisms and adhesive joints which become ground for novel multifunctional material solutions to facilitate the joining performance. Then, the new challenges with new materials and processing techniques will be discussed from destructive and non-destructive mechanical characterization perspectives. Finally, the durability of the future material will be reported under harsh service conditions at different processing conditions.

Keywords: composite materials, manufacturing technique, automated fiber placement, large-scale structures, high-volume structures, high-performance thermoplastic

Index of Authors

--/ **A** /--
 Afonso, F 505, 525
 Aksoy, E 88
 Arteiro, A 142

--/ **B** /--
 Badia, S 284
 Barbosa, F R 439
 Bartoli, N 2, 368, 452
 Battaglia, L 34
 Bellier, T 102
 Benard, E 22
 Bento, D 439
 Bil, C 102
 Bookwala, M 495
 Bouzouiki, M E 310
 Brenner, F 439
 Bronz, M 368
 Budinger, V 478
 Bussemaker, J H 452

--/ **C** /--
 Canfield, R A 316
 Cardoso, I 2
 Carlini, F 34
 Cavallaro, R 545
 Cea, A 406
 Chen, Y E 158
 Cini, A 545
 Clarich, A 34
 Corrado, G 142
 Costa, A da 239
 Cretella, V 579
 Cronk, A 495

--/ **D** /--
 Delbecq, S 478
 Della Vecchia, P 452
 Diouane, Y 452
 Donelli, G 452
 Dubreuil, S 2

--/ **E** /--
 Eisenhut, D 439

--/ **F** /--
 Faraj, R 173
 Farhat, C 1
 Fereidooni, A 157, 158
 Fernandez, L F T 368
 Fossati, M 406

--/ **G** /--
 Gameiro, R S 260
 Gandolfi, R 439
 Gil, P J S 209
 Giuseppe, R di 478
 Gogu, C 2, 74
 Graczykowski, C 173, 379
 Grewal, A 157
 Guibert, A T R 495
 Gupta, K 388

--/ **H** /--
 He, P 22
 Hill, M R 284
 Holnicki-Szulc, J 173, 379

--/ **K** /--
 Kim, H A 495, 534
 Kipouros, T 222
 Knap, L 173, 379
 Krupa, G 222

--/ **L** /--
 Lafage, R 2, 452
 Laliberte, R 158
 Lau, F 565
 Le Lamer, Y 22
 Lefebvre, T 368, 452
 Lue, G 157
 Lunghitano, A 525

--/ **M** /--
 Mallon, C N 284
 Mandorino, M 452
 Mangold, J 439
 Mantegh, I 388
 Marquier, S 74
 Marta, A C 120, 184, 240, 260
 Matos, N M B 120, 240, 260
 Melis, M 505
 Mello, J M G de 452
 Meng, Y S 495
 Mikulowski, G 173
 Moebis, N 439
 Monteiro, J P L 209
 Morlier, J 22, 74, 102, 452
 Muradas, D 74

--/ **N** /--
 Nagy, P 406
 Najjaran H 388
 Neofytou, A 534

--/ **O** /--
 Odaguil, F I K 439

--/ **P** /--
 Pagani, A 579
 Palacios, R 406
 Pauvert, V 478
 Pawlowski, P 173
 Portugal, M 184
 Pucciarelli, S 565
 Pudsey, A 102

--/ **R** /--
 Reinoso, J 142
 Ricci, S 533
 Rocha, R M 209
 Roussouly, N 406
 Russo, R 34

--/ **S** /--
 Sahin, M 88
 Salaün, M 2
 Saves, P 452
 Scalia, L 545
 Sedaghati, R 310
 Sekula, K 173
 Silva, V M T 240
 Sohoulis, A 579
 Sotouni, S 388
 Souza, A 505
 Spinelli, A 222
 Stiharu, I 310
 Suleman, A 525, 565, 579
 Świercz, A 173, 379

--/ **T** /--
 Teza, H F 439
 Thornton, A W 284
 Tucsok, M 388

--/ **U** /--
 Urbano, A 102

--/ **W** /--
 Wiacek, D 173
 Wickramasinghe, V 158
 Wolejsza, Z 173, 379
 Wu, S 54

--/ **Y** /--
 Yano, M 157
 Yildiz, M 599
 Yokozeki, T 54

