4 EJIL - LAETA Young Researchers Meeting
AeroG, Covilhã, 9-10 November 2017

NUMERICAL VERIFICATION OF ADJOINT-BASED SENSITIVITY ANALYSIS FOR MULTISTAGE TURBOMACHINERY DESIGN

Simão S. Rodrigues[a,*] and André C. Marta[a]

[a]IDMEC - LAETA, Avenida Rovisco Pais 1, 1049-001 Lisboa
[*]e-mail: simao.rodrigues@tecnico.ulisboa.pt

Keywords: turbomachinery, discrete adjoint, multistage, gradient-based optimization, parallel processing, finite-differences.

Abstract

With the exponential growth in computational power as well as improvements in the accuracy of computational fluid dynamics (CFD) tools, their use in turbomachinery design and analysis has seen a great increase, particularly in optimization environments, where gradient-based optimization algorithms are often selected for their efficiency. These algorithms require the computation of the sensitivities of the functions of interest to the design variables. The number of design variables in an optimization problem may be in the order of thousands. As such, the use of the adjoint approach for calculating the gradients is highly advantageous as it produces function sensitivities with computational cost that is nearly independent of the number of design variables. In the analysis of turbomachinery, accounting for the interaction between the multiple blade passages is of paramount importance if one wishes to increase the accuracy of the simulation. Many computational methods exist to address this interactions. The mixing-plane treatment is one of the most widely used methods in the steady analysis of multiple rows of a turbomachine. This paper describes improvements to a discrete adjoint solver of a proprietary CFD solver for multistage turbomachinery applications, namely the adjoint counterpart of the mixing-plane formulation of the direct solver. The adjoint solver is developed using the ADjoint approach, where the partial derivatives required for the assembly of the adjoint system of equations are obtained using automatic differentiation tools. A verification of the implementation of the mixing-plane against the finite-difference approximations is presented. Sensitivities of selected surface functions of interest, such as mass flow, to other selected design parameters, such as surface nodes or inflow boundary conditions, calculated with both methods are presented. The results show good agreement of both derivatives and emphasise the benefits of the adjoint approach versus finite-differences in terms of accuracy and computational cost.

1 INTRODUCTION

With the growth in computational power, external and internal flow simulations using high-fidelity computational fluid dynamic (CFD) models have become a routine, with the emerging trend being to use optimization techniques as part of the design process, both in academia and industry.

Given the nature of the flow models, a numerical simulation may take hours or even days to complete a function evaluation, meaning that an optimization case, which may require hundreds of function evaluations to find an optimum, may lead to a prohibitive time requirement. For this reason, the most commonly used optimization methods are the gradient based (GB) ones, which are the most efficient. These GB methods, however, require the calculation of the derivatives, which, if using methods such as the commonly used finite difference method, may also lead to prohibitive computational and time requirements, in the case of a high number of design variables. This problem is overcome by the adjoint method, which produces exact derivatives with a cost that is independent of the number of design variables.

The adjoint method was first introduced to computational fluid dynamics by Pironneau [1] and further extended by Jameson to optimization of airfoil profiles [2] and wings [3]. More recently it as been used in solving multipoint aerodynamic shape [4, 5] and aero-structural [6] optimization problems, magneto-hydrodynamic flow control [7] and turbine blades [8]. Other developments on the application of the adjoint approach to gradient-based optimization in turbomachinery environments have also been made. However, most of these cases cannot account for the interaction between different blade passages, which has an important impact on the whole performance of a multistage turbomachine [9]. Its incorporation on the optimization environment would therefore provide an more realistic insight of the direction to which the optimization should proceed. Frey et al. [10], Wang et al. [12, 11] and Walther and Nadarajah [13, 14] present adjoint solvers which allow multi-row optimization.

While Frey et al. uses finite differences to obtain the derivatives to set-up the adjoint system of equations, in their work, Walther and Nadarajah manually differentiate the routines that compute those derivatives manually.

Following the work of Marta et al. [15] on the implementation of the adjoint solver of a proprietary turbomachinery CFD solver, the adjoint multistage interface was implemented onto the same adjoint solver using Automatic Differentiation (AD) to compute the partial derivatives for the adjoint equations. This allows for a much faster development than if differentiating by hand while still obtaining the computational benefits of avoiding the finite difference method.

This paper presents a description of the adjoint multistage interface followed by the numerical verification of the final derivatives computed by the improved adjoint solver, against finite-difference approximations, in a stator-rotor stage simulation.

2 BACKGROUND

In a turbomachinery design environment various parameters can be used to define its geometry and operating conditions, such as blade stagger, camber angle and thickness distributions and axial and radial stacking. All these inputs will influence one or more performance characteristics that are to be studied (and improved), such as efficiency, pressure ratio or mass flow. This can constitute an optimization problem, where the adjustable parameters are the design variables and the performance characteristics are the functions of interest, either the cost function or come constraints.

2.1 GRADIENT-BASED OPTIMIZATION

A generic CFD design problem can be formulated as

$$\begin{array}{ll} \text{Minimize} & I(\boldsymbol{\alpha}, \boldsymbol{q}(\boldsymbol{\alpha})) \\ \text{w.r.t} & \boldsymbol{\alpha}, \\ \text{subject to} & \mathcal{R}(\boldsymbol{\alpha}, \boldsymbol{q}(\boldsymbol{\alpha})) = 0 \\ & \mathcal{C}(\boldsymbol{\alpha}, \boldsymbol{q}(\boldsymbol{\alpha})) = 0, \end{array} \tag{1}$$

where I is the cost function, $\boldsymbol{\alpha}$ is the vector of design variables, $\boldsymbol{q}$ is the flow solution and $\mathcal{C}$ represents additional constraints that may or may not involve the flow solution. The flow governing equations are expressed in the form $\mathcal{R} = 0$ and appear as a constraint, as the solution $\boldsymbol{q}$ must always obey the flow physics.

Gradient Based (GB) optimization algorithms are the most efficient methods to minimize a function with regard to a set of design variables. They make use of the gradient of the function to determine the direction to follow at each step on the search space. This means, however, that these gradients must be computed somehow.

2.1.1 Sensitivity Analysis

There are many methods one can choose to compute the gradients required by a GB algorithm, such as finite-differences (FD), complex-step (CS), algorithmic differentiation (AD), adjoint methods, etc.

The simplest way to compute them is to use finite difference approximation, as it requires little to no change in the analysis code, which can be treated as a black box. This method however comes with some disadvantages. The approximation is very dependent of the perturbation step, prone to subtractive errors, and requires at least one extra solver run to compute the sensitivity to one design variable.

Complex-step approximations eliminate the subtractive error introduced by finite-differences, however, its computational cost is still proportional to the number of design variables and requires that the solver can accept complex numbers.

Automatic (or algorithmic) differentiation (AD) consists in applying the chain rule to computer programs in order to obtain derivatives of their outputs based on their inputs.

The adjoint method is a analythic method which allows the computation of the the sensitivities with a computational cost that is independent of the number of design variables. The implementation of the adjoint equations for a given system of PDE's can be achieved in two ways, the *continuous* and the *discrete adjoint approach.* While the first forms a continuous adjoint problem from the governing PDE's and then discretizes the problem to solve it numerically, the *discrete adjoint approach* first discretizes the governing equations and then derives the adjoint system for the discrete equations. These two approaches result in different systems of linear equations that, in theory, converge to the same result with mesh refinement.

The discrete approach has the advantage of being able to be applied to any set of governing equations and to treat arbitrary cost functions and provides sensitivities that are consistent with those produced by the discretized solver. It is also easier to obtain the appropriate BC's for the adjoint solver with the discrete approach.

2.1.2 Flow governing equations

The present work uses the Reynolds-Averaged Navier-Stokes equations (RANS) for describing the flow. The Navier-Stokes equations, in conservation form, can be written as

$$\frac{\partial \boldsymbol{q}}{\partial t} + \frac{\partial \boldsymbol{f}_i}{\partial x_i} - \frac{\partial \boldsymbol{f}_{\mathrm{v}i}}{\partial x_i} = 0, \tag{2}$$

where, $\boldsymbol{q}$, $\boldsymbol{f}_i$ and $\boldsymbol{f}_{\mathrm{v}_i}$ are the vectors of state variables, inviscid, and viscous fluxes, respectively, define as

$$\boldsymbol{q} = \left\{ \begin{array}{c} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{array} \right\}, \ \boldsymbol{f}_i = \left\{ \begin{array}{c} \rho u_i \\ \rho u_1 u_i + p\delta_{i1} \\ \rho u_2 u_i + p\delta_{i2} \\ \rho u_3 u_i + p\delta_{i3} \\ \rho E u_i + p u_i \end{array} \right\}, \ \boldsymbol{f}_{\mathrm{v}i} = \left\{ \begin{array}{c} 0 \\ \tau_{ij}\delta_{i1} \\ \tau_{ij}\delta_{i2} \\ \tau_{ij}\delta_{i3} \\ u_j \tau_{ij} + q_i \end{array} \right\}, \tag{3}$$

where ρ is the density, u_i is the mean velocity in direction i, E is the total energy, τ_{ij} is the viscous stress and q_i is the heat flux. To model the Reynolds stresses, Wilcox's two-equation $k - \omega$ turbulence model [16] is used, resulting in a system with 7 equations.

The RANS equations can be expressed in their semi-discrete form, as

$$\frac{\mathrm{d}\boldsymbol{q}_{ijk}}{\mathrm{d}t} + \boldsymbol{R}_{ijk}(\boldsymbol{q}) = 0, \tag{4}$$

where $\boldsymbol{R}$ is the residual of the inviscid, viscous, turbulent fluxes, boundary conditions and artificial dissipation. The triad (i, j, k) represents the three computational directions. Since this work deals with the steady solutions of the RANS equations the unsteady term is dropped out for the rest of the paper.

2.1.3 Adjoint equations

Following the work by Giles and Pierce [17] in derivation of the adjoing equations for systems of PDEs, the adjoint for the flow equations in eq. (4) can be expressed as

$$\left[\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{q}}\right]^T \boldsymbol{\psi} = \left[\frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{q}}\right]^T, \tag{5}$$

where $\boldsymbol{\mathcal{F}}$ is the function of interest, $\boldsymbol{\psi}$ is the adjoint vector, which is used in the calculation of the total gradient of the function of interest with respect to a set of variables of interest $\boldsymbol{\alpha}$, given by

$$\frac{\mathrm{d}\boldsymbol{\mathcal{F}}}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{\alpha}} - \boldsymbol{\psi}^T \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{\alpha}}. \tag{6}$$

The choosing of the variables in the previous equations is only limited to being able to describe the objective function of residual in terms of those variables.

2.1.4 ADjoint method

This hybrid approach consists in computing the total derivative with the previously described adjoint method, where the partial derivatives of eqs. (5) and (6) are computed with automatically differentiated routines.

In this approach, the residual calculation is rearranged (if needed) into a routine that has as inputs the information of the stencil of influence and outputs the residual [18]. This routine is then differentiated using an AD tool, thus producing the necessary terms for the calculation of the adjoint solution.

2.2 MULTISTAGE TURBOMACHINERY

The use of mixing planes to permit a quasi-steady analysis of inherently unsteady multistage turbomachinery flows is a well-established idea. It requires only one stator blade per stage and, thus removing all transient rotor-stator interactions, it still gives fairly representative results [9]. The overall mixing-plane algorithm Holmes [19] (which is schematically represented in fig. 1) can be condensed in the following steps:

1. Compute the fluxes from conserved quantities and average them at each spanwise position;

$$\tilde{\boldsymbol{q}}_j = \frac{\sum_{i=1}^{n_i} \boldsymbol{q}_{ij} a_{ij}}{\sum_{i=1}^{n_i} a_{ij}}, \tag{7}$$

$$\boldsymbol{p}_{\text{local},j} = f(\tilde{\boldsymbol{q}}_j) \tag{8}$$

2. Communicate the radial profiles of averaged quantities between blade rows;

$$\boldsymbol{p}_{\text{local}} \xrightarrow{\text{MPI}} \boldsymbol{p}_{\text{don}} \tag{9}$$

3. Interpolate the received profiles to match local cell distribution;

$$\boldsymbol{p}_{\text{rec}} = f(\boldsymbol{p}_{\text{don}}) \tag{10}$$

4. Compute the variation in the conserved variables to be applied to the ghost cells, from the flux differences and update them.

$$\boldsymbol{q}^*_{\text{local}} = f(\boldsymbol{p}_{\text{rec}}, \boldsymbol{q}_{\text{local}}) \tag{11}$$

Although fig. 1 only represents the transfer of information from one row to another, for simplicity, this algorithm occurs in both directions for each interface. To assure maximum non-reflectivity in the interface, the method uses the two dimensional approach of Giles [20].

2.3 ADJOINT MULTISTAGE INTERFACE

Assuming a simulating of a series of n blade rows, each blade will influence and be influenced by its neighbours. If no multistage interface is used, a system of equations eq. (5) is solved for each row. However, to consider the influence of the rows on each other, a coupled coupled systems of equations must be solved (eq. (12)).

$$\begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{q}_1}\right] & \cdots & \left[\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{q}_n}\right] \\ \vdots & \ddots & \vdots \\ \left[\frac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{q}_1}\right] & \cdots & \left[\frac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{q}_n}\right] \end{bmatrix} \begin{Bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_n \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \mathcal{F}}{\partial \boldsymbol{q}_1} \\ \vdots \\ \frac{\partial \mathcal{F}}{\partial \boldsymbol{q}_n} \end{Bmatrix}. \tag{12}$$

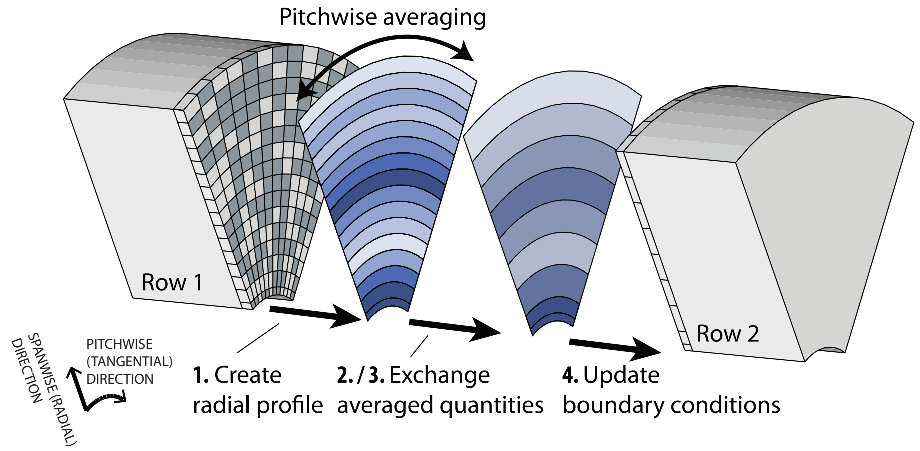
Pitchwise averaging
Row 1
Row 2
SPANWISE (RADIAL) DIRECTION
PITCHWISE (TANGENTIAL) DIRECTION
1. Create radial profile
2./3. Exchange averaged quantities
4. Update boundary conditions

Figure 1: Schematic representation of the multistage mixing-plane interface steps.

The term $\partial \boldsymbol{R}_i/\partial \boldsymbol{q}_j$ represents the influence of row j in the residual of row i. Each row only influences its neighbours, therefore

$$\left[\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{q}_j}\right] = 0\,, \quad i-1 > j > i+1. \tag{13}$$

Going back to the multistage interface described in section 2.2, the derivative of the residual of a certain cell in the receiver in order to the state variables of another cell in the donor (the terms outside of the diagonal of the matrix represented in eq. (12)) could be expressed as

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{don}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{rec}}}\frac{\partial \boldsymbol{q}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{don}}}. \tag{14}$$

Using the chain rule, a more detailed expression can be derived, taking into account all the various steps of the mixing-plane algorithm,

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{don}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}^*_{\text{local}}}\frac{\partial \boldsymbol{q}^*_{\text{local}}}{\partial \boldsymbol{p}_{\text{rec}}}\frac{\partial \boldsymbol{p}_{\text{rec}}}{\partial \boldsymbol{p}_{\text{don}}}\frac{\partial \boldsymbol{p}_{\text{don}}}{\partial \tilde{\boldsymbol{q}}_{\text{don}}}\frac{\partial \tilde{\boldsymbol{q}}_{\text{don}}}{\partial \boldsymbol{q}_{\text{don}}}, \tag{15}$$

With the previous terms it is then possible to compute the coupled multistage adjoint solution. To obtain the final derivative given by eq. (6) is is also necessary to compute $\partial \boldsymbol{R}/\partial \boldsymbol{\alpha}$ with the coupling taken into account.

In the present work, the derivatives to be computed are $\mathrm{d}\boldsymbol{\mathcal{F}}/\mathrm{d}\boldsymbol{u}$ and $\mathrm{d}\boldsymbol{\mathcal{F}}/\mathrm{d}\boldsymbol{x}$. The first does not need the multistage coupling to be taken into account, as the boundary conditions used as variables of interest are located at the entry of the first row or at the exit of the last row. The latter, however, requires the computation of coupling terms, due to the averaging by area creating a dependence of mesh on the face of the adjacent row. It is therefore necessary to also include coupling terms on the last term of eq. (6), $\partial \boldsymbol{R}/\partial \boldsymbol{x}$, which can be obtained following the chain rule,

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{x}_{\text{don}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \tilde{\boldsymbol{q}}_{\text{don}}}\frac{\partial \tilde{\boldsymbol{q}}_{\text{don}}}{\partial \boldsymbol{x}_{\text{don}}}. \tag{16}$$

3 IMPLEMENTATION

In this section, some details of the implementation of the coupling multistage interface previously described are presented.

3.1 FLOW SOLVER

In the present work the steady RANS equations are solve with a proprietary solver [21]. It supports three-dimensional, multi-block and structure grids, as well as multiple turbulence models such as the $k - \omega$, $k - \epsilon$ and SST, having the option to use wall functions or wall integration for the boundary layer resolution. It is also capable of runing in multiple processors via Message Passing Interface (MPI). For the interaction between blade passages, when solving for steady state, the previously described mixing-plane formulation is used.

3.2 ADJOINT SOLVER

The discrete adjoint solver for the mentioned flow solver was previously implemented by using the ADjoint hybrid approach [18]. The AD tool chosen in the mentioned work, as well as in the present work, was Tapenade [22], as it supports Fortran 90, which is the programming language used in the flow solver implementation. The built-in Krylov subspace method of the Portable, Extensible Toolkit for Scientific Computation (PETSc) [23] is used to solve the system of equations, more specifically, the generalized minimum residual method with the incomplete factorization preconditioner with one level fill, ILU(1).

3.3 ADJOINT MULTISTAGE INTERFACE

Using the previously described ADjoint approach, the implementation of the adjoint multistage interface would require a routine that receives the state solution at two adjacent faces and would compute the updated local solution at the face in treatment. This routine would then be differentiated to obtain the term $\partial \boldsymbol{q}_{\text{rec}}/\partial \boldsymbol{q}_{\text{don}}$ of eq. (14).

Such routine proved very difficult to implement, due to the complexity of the direct solver routines, as well as the use of features of the Fortran language that proved to be dificult to differentiate by the selected AD tool. With that in mind, a series of requirements needed to be complied by the rewritten routines to be differentiated to obtain the terms of eqs. (15) and (16):

1. Routines inputs and outputs must passed as arguments in order to be properly differentiated.
2. The input and output variables should not be structures with dinamically allocated arrays, as the AD tool in use does not deal with it easily;
3. The routine to be differentiated should not have any MPI calls.

The first item of the previous list had to be addressed in most of the subroutines of the direct solver, as the code was programmed to manage memory stored in Fortran modules. This implied having to define new derived storage structures.

As the original solver was structured in the structure with many arrays way. One approach to this requirement would be to use the arrays of the variables inside the structures directly as arguments. However, as the number of variables was quite big (particularly for the profile computation), the use of an array of structures, each structure containing a value for each variable was chosen instead.

The direct mixing-plane interface is inherently parallel (e.g. the accumulation of quantities in 1D profile, passage between rows, etc), as such, the

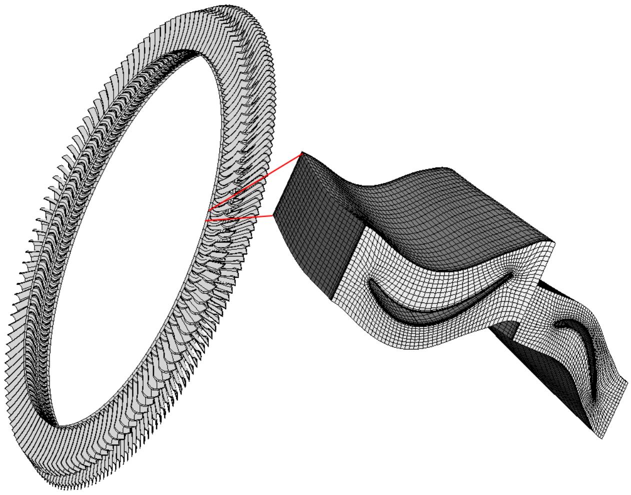

Figure 2: Representation of stator-rotor stage in study and respective computational mesh.

number of MPI communication calls in the various subroutines was substantial. To comply with this item, the choice was either store all the solution across the multiple processes in one master process that would perform all the computations or maintain the parallel structure of the code. The latter approach was chosen.

4 VERIFICATION OF THE MULTISTAGE SENSITIVITY ANALYSIS

This section presents the verification of the implemented adjoint mixing-plane interface with the simulation of a stator-rotor stage of General Electric GenX LP Turbine. The computational domain is made of two single blade passages, with a total of 90750 cells (see fig. 2).

The adjoint multistage interface was verified with a second order finite-difference approximation, with the two methods showing good agreement. The detailed results are to be presented in a paper to be submitted.

This paper will present the verification of the total derivatives computed using the improved adjoint solver, with the multistage interface. The selected quantity of interest is the mass flow on the outlet of the computational domain of the rotor ($\dot{m}_2^{\text{exit}}$) and shall be used for the rest of the paper. The direct and adjoint solutions for the equation of continuity (ρ) are presented in fig. 3.

The influence of the inlet total pressure (p_1^{in}) on $\dot{m}_2^{\text{exit}}$, given by $\dot{m}_2^{\text{exit}}/p_1^{\text{in}}$, is presented in fig. 4.

The same derivative was computed with finite-differences for five control cells on the inlet face and are presented in fig. 5, along with the derivatives computed with the adjoint method and the relative error between the two values. The perturbation step used in the finite-difference approximation was converged until the relative error was below 1%. While the adjoint method was able to compute the derivatives for the whole domain with one extra computation (with a similar computational cost of one direct run), for the finite-difference approximation, a minimum of one direct run was needed for each control cell. In reality, it took many direct runs in order to converge the perturbation step for each of the control cells.

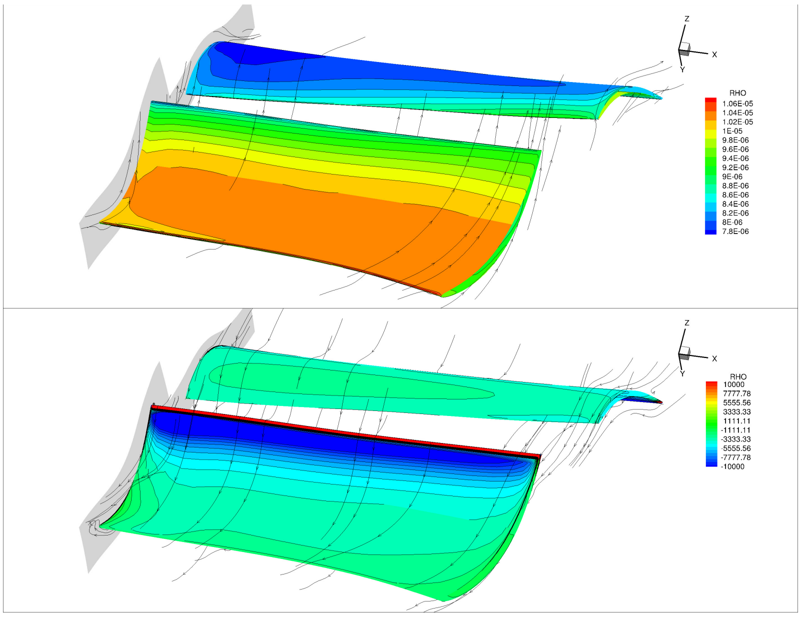

Figure 3: Direct (q_1) and adjoint (ψ_1) solutions.

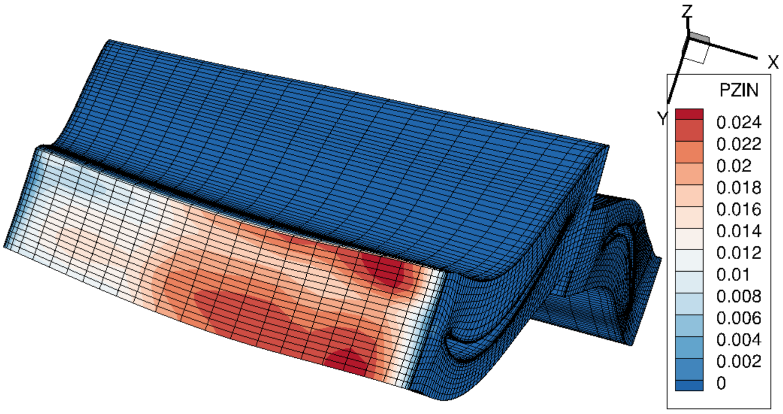

Figure 4: Sensitivity of exit mass flow $\dot{m}_2^{\text{exit}}$ to the inlet total pressure p_1^{in} $(\dot{m}_2^{\text{exit}}/p_1^{\text{in}})$.

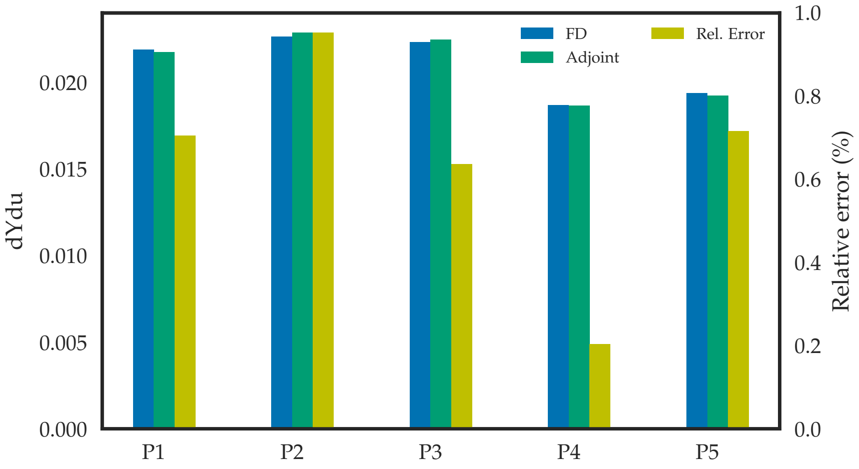

Figure 5: Relative errors and derivatives of verification cells.

5 CONCLUSIONS

The formulation of the discrete adjoint mixing-plane was developed and implemented on a proprietary CFD code, using AD tools. The adjoint mixing-plane implementation was verified against finite-differences. The multistage adjoint solution was computed for a stator-rotor case and the final derivative of various functions of interest were presented, along with comparison with finite differences. The derivatives computed with the two approaches showed good agreement, with relative errors below 1%, thus confirming the correct implementation of the adjoint multistage interface. While the adjoint method required only one extra solver run with a computational cost similar to one direct run, the finite difference approach required many direct solver runs in order to obtain a converged value, thus emphasizing the benefits of using the adjoint method for sensitivity analysis.

6 ACKNOWLEDGEMENTS

The first author thanks Fundação para a Ciência e Tecnologia (FCT), Portugal for the Ph.D. Grant SFRH/BD/97521/2013. The second author thanks the support by FCT through IDMEC, under LAETA, project Pest-OE/EME/LA0022.

REFERENCES

[1] O. Pironneau. “On optimum design in fluid mechanics”. In: *Journal of Fluid Mechanics* 64.01 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.

[2] A. Jameson. “Aerodynamic design via control theory”. In: *Journal of Scientific Computing* 3.3 (1988), pp. 233–260.

[3] A. Jameson, L. Martinelli, and N. A. Pierce. “Optimum aerodynamic design using the Navier–Stokes equations”. In: *Theoretical and Computational Fluid Dynamics* 10.1-4 (1998), pp. 213–237.

[4] J. J. Reuther et al. “Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1”. In: *Journal of Aircraft* 36.1 (1999), pp. 51–60.

[5] J. J. Reuther et al. “Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 2”. In: *Journal of Aircraft* 36.1 (1999), pp. 61–74.

[6] J. R. Martins, J. J. Alonso, and J. J. Reuther. “High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet”. In: *Journal of Aircraft* 41.3 (2004), pp. 523–530. ISSN: 0021-8669. DOI: 10.2514/1.11478.

[7] A. C. Marta and J. J. Alonso. “Toward optimally seeded airflow on hypersonic vehicles using control theory”. In: *Computers & Fluids* 39.9 (Oct. 2010), pp. 1562–1574. ISSN: 0045-7930. DOI: 10.1016/j.compfluid.2010.05.009.

[8] A. C. Marta and S. Shankaran. “Exploring the use of Adjoint Methods for Detailed Sensitivity Analysis on Turbomachinery”. In: International Conference on Engineering Optimization. Rio de Janeiro, Brasil, 2012.

[9] J. D. Denton. "The Calculation of Three-Dimensional Viscous Flow Through Multistage Turbomachines". In: *Journal of Turbomachinery* 114.1 (Jan. 1, 1992), pp. 18–26. ISSN: 0889-504X. DOI: 10.1115/1.2927983.

[10] C. Frey, H.-P. Kersken, and D. Nurnberger. "The discrete adjoint of a turbomachinery RANS solver". In: *ASME Turbo Expo 2009: Power for Land, Sea, and Air*. American Society of Mechanical Engineers, 2009, pp. 345–354.

[11] D. X. Wang et al. "Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines—Part II: Validation and Application". In: *Journal of Turbomachinery* 132.2 (2010), p. 021012. ISSN: 0889504X. DOI: 10.1115/1.3103928.

[12] D. X. Wang and L. He. "Adjoint Aerodynamic Design Optimization for Blades in Multistage Turbomachines—Part I: Methodology and Verification". In: *Journal of Turbomachinery* 132.2 (2010), p. 021011. ISSN: 0889504X. DOI: 10.1115/1.3072498.

[13] B. Walther and S. Nadarajah. "Constrained Adjoint-Based Aerodynamic Shape Optimization of a Single-Stage Transonic Compressor". In: *Journal of Turbomachinery* 135.021017 (2013), p. 021017.

[14] B. Walther and S. Nadarajah. "An Adjoint-Based Optimization Method for Constrained Aerodynamic Shape Design of Three-Dimensional Blades in Multi-Row Turbomachinery Configurations". In: (June 16, 2014), V02BT39A031. DOI: 10.1115/GT2014-26604.

[15] A. C. Marta, S. Shankaran, and A. Stein. "Blade Shape Optimization using a RANS Discrete Adjoint Solver". In: *Proceedings of the 2nd International Conference on Engineering Optimization, number ENGOPT2010-1410, Lisbon, Portugal.* 2010.

[16] D. C. Wilcox. "Reassessment of the scale-determining equation for advanced turbulence models". In: *AIAA Journal* 26.11 (1988), pp. 1299–1310. ISSN: 0001-1452. DOI: 10.2514/3.10041.

[17] M. B. Giles and N. A. Pierce. "An introduction to the adjoint approach to design". In: *Flow, turbulence and combustion* 65.3-4 (2000), pp. 393–415.

[18] A. C. Marta et al. "A methodology for the development of discrete adjoint solvers using automatic differentiation tools". In: *International Journal of Computational Fluid Dynamics* 21.9-10 (Oct. 2007), pp. 307–327. ISSN: 1061-8562, 1029-0257. DOI: 10.1080/10618560701678647.

[19] D. G. Holmes. "Mixing Planes Revisited: A Steady Mixing Plane Approach Designed to Combine High Levels of Conservation and Robustness". In: *Proceedings of ASME Turbo Expo 2008: Power for Land, Sea and Air*. Berlin, Germany, Jan. 1, 2008, pp. 2649–2658. DOI: 10.1115/GT2008-51296.

[20] M. B. Giles. *UNSFLO: a Numerical Method for Unsteady Inviscid Flow in Turbomachinery.* Gas Turbine Laboratory, Massachusetts Institute of Technology, 1988.

[21] D. G. Holmes, B. J. Moore, and S. D. Connell. "Unsteady vs. steady turbomachinery flow analysis: exploiting large-scale computations to deepen our understanding of turbomachinery flows". In: SciDAC Conference. Denver, CO, 2011.

[22] L. Hascoët and V. Pascual. "The Tapenade Automatic Differentiation tool: Principles, Model, and Specification". In: *ACM Transactions On Mathematical Software* 39.3 (2013).

[23] S. Balay et al. *PETSc Users Manual Revision 3.5*. 2014.

[24] A. C. Marta et al. "Interpretation of Adjoint Solutions for Turbomachinery Flows". In: *AIAA Journal* 51.7 (July 2013), pp. 1733–1744. ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J052177.