Congresso de Métodos Numéricos em Engenharia 2015
Lisboa, 29 de Junho a 2 de Julho 2015

DISCRETE ADJOINT MIXING-PLANE FORMULATION FOR MULTI-STAGE TURBOMACHINERY DESIGN

Simão S. Rodrigues*, André C. Marta

CCTAE, IDMEC,
Instituto Superior Técnico, Universidade de Lisboa
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal
e-mail: simao.rodrigues,andre.marta@tecnico.ulisboa.pt

Keywords: turbomachinery, discrete adjoint, multi-stage, gradient-based optimization, parallel processing, design framework

Abstract. *The use of computational fluid dynamics (CFD) tools in turbomachinery has seen an increase as a result of the exponential growth of computational power, as well as of improvements of the accuracy of numerical simulations. These tools are often used in optimization environments, where gradient-based optimization algorithms are the most common due to its efficiency. The optimization may contain a large number of design variables (typically in the order of thousands), such in cases of shape optimization. In these cases, the adjoint approach for calculating the gradients is beneficial, as it provides a way of obtaining function sensitivities with a computational cost that is independent of the number of design variables, as opposed to what happens with the well known finite-difference approach. The interaction between adjacent blade rows has an important impact on the whole performance of a multistage turbomachine. The most commonly used method to address these effects in the simulation of multiple rows is the mixing-plane treatment, that has become a standard industrial tool in the design environment. In this paper, improvements on the adjoint solver of a proprietary CFD solver for multistage turbomachinery applications are presented, namely the adjoint counterpart of the mixing-plane formulation used in the direct solver. The solver is developed using the discrete ADjoint approach, where the partial derivatives required for the assembly of the adjoint system of equations are obtained using automatic differentiation tools.*

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 multi-point 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, this paper describes the work developed on the implementation of the adjoint multistage interface onto the same adjoint solver using Automatic Differentiation (AD) tools to obtain the 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.

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. The 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.

2.1 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

$$\frac{dq_{ijk}}{dt} + \mathcal{R}_{ijk}(\boldsymbol{q}) = 0, \tag{4}$$

where $\mathcal{R}$ 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.

2.2 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

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

where I is the function of interest and $\boldsymbol{\psi}$ is the adjoint vector, which is used in the calculation of the total gradient of the function of interest with respect ot the computational grid coordinates of each node $\boldsymbol{x}$, given by

$$\frac{dI}{d\boldsymbol{x}} = \frac{\partial I}{\partial \boldsymbol{x}} - \boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{x}}. \tag{6}$$

Since typically the design variables $\boldsymbol{\alpha}$ are not geometric parameters handled directly by the CFD solver, it is necessary to apply the chain rule of differentiation to express the gradient of I with respect to the desired design parameters as

$$\frac{dI}{d\boldsymbol{\alpha}} = \frac{dI}{d\boldsymbol{x}} \frac{d\boldsymbol{x}}{d\boldsymbol{\alpha}}. \tag{7}$$

The last term in eq. (7), $\partial \boldsymbol{x}/\partial \boldsymbol{\alpha}$, implies the sensitivity analysis of the grid generation routine, which implicitly defines the function $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\alpha})$.

2.3 Multistage mixing plane

The use of mixing planes to permit a quasi-steady analysis of inherently unsteady multi-stage turbomachinery flows is a well-established idea [9]. Holmes [18] describes a mixing plane algorithm that achieves several key goals, including complete flux conservation at the interface, robustness, indifference to local flow direction and non-reflectivity. It consists in using a control-theory based flux balance algorithm to drive the differences between the fluxes in the two faces to zero, by updating the conserved variables in the ghost cells with a value based on the flux differences. To assure maximum non-reflectivity in the interface, the method uses the two dimensional approach of Giles [19].

The overall mixing-plane algorithm (which is schematically represented in fig. 1) consists in the following:

1. Compute the fluxes from conserved quantities and average them at each spanwise position;
2. Communicate the radial profiles of averaged quantities between blade rows;
3. Interpolate the received profiles to match local cell distribution;
4. Compute the variation in the conserved variables to be applied to the ghost cells, from the flux differences, and update them.

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.

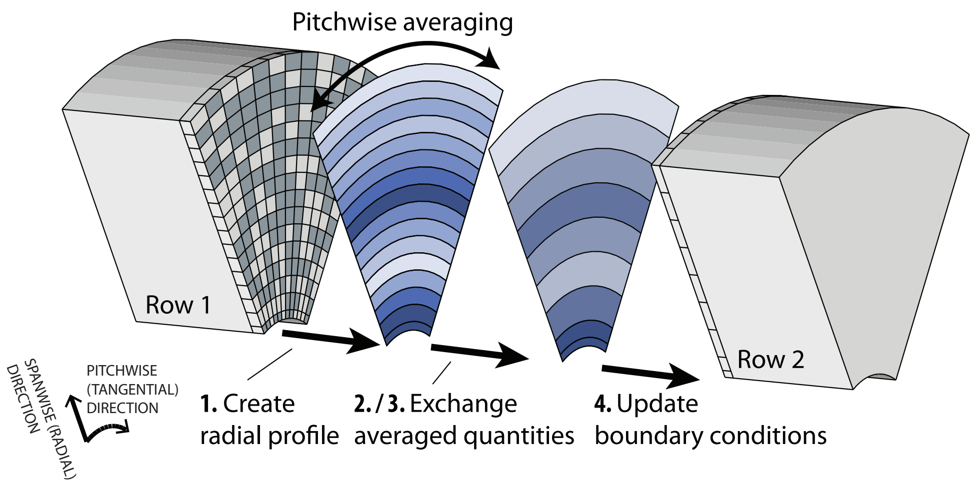

Figure 1: Schematic representation of the multistage mixing-plane interface steps.

3 IMPLEMENTATION

In this section the flow solver and the implementation approach of its adjoint are briefly described.

3.1 Flow solver

The proprietary flow solver is capable of solving the steady, Reynolds-averaged Navier-Stokes equations (RANS), as well as the unsteady Reynolds-aveaged Navier-Stokes equations (URANS) [20]. It supports three-dimensional, multi-block and structured grids. Available turbulence models include the $k-\omega$, $k-\epsilon$ and SST, having the option to use wall functions or wall integration for the boundary layer resolution.

3.2 Adjoint solver

A discrete adjoint solver for the mentioned flow solver was previously implemented by using the so called ADjoint hybrid approach [21]. In this approach, the solver is derived with the aid of an automatic differentiation (AD) tool, which is selectively applied to the flow solver source code to produce the routines that evaluate the partial derivative matrices $\partial\mathcal{R}/\partial\boldsymbol{q}$, $\partial I/\partial\boldsymbol{q}$, $\partial\mathcal{R}/\partial\boldsymbol{x}$ and $\partial I/\partial\boldsymbol{x}$ of eqs. (5) and (6). 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. Once the adjoint linear system of equations is assembled, the Portable, Extensible Toolkit for Scientific Computation (PETSc) [23] is used to solve it.

3.3 Adjoint multistage interface

Assuming a simulating of a turbomachine stage consisting in a rotor-stator pair of blade rows (row 1 and row 2), 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, two coupled systems of equations must be solved, as schematically represented in fig. 2. There will be two horizontal strips of non-zero values on the non diagonal part of

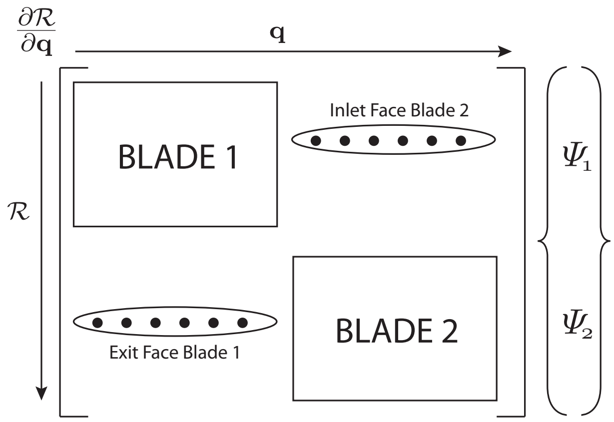

Figure 2: Schematic representation of the adjoint system of equations for two coupled turbomachinery rows.

the coupled system matrix, relative to the influence of the cells in the face of one row to the residual of some of the cells on the other row.

Going back to the multistage interface described in section 2.3, 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 fig. 2) could be expressed as

$$\frac{\partial \mathcal{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{don}}} = \frac{\partial \mathcal{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{rec}}} \frac{\partial \boldsymbol{q}_{\text{rec}}}{\partial \boldsymbol{p}_{\text{rec}}} \frac{\partial \boldsymbol{p}_{\text{rec}}}{\partial \boldsymbol{p}_{\text{don}}} \frac{\partial \boldsymbol{p}_{\text{don}}}{\partial \tilde{\boldsymbol{w}}_{\text{don}}} \frac{\partial \tilde{\boldsymbol{w}}_{\text{don}}}{\partial \boldsymbol{q}_{\text{don}}}, \tag{8}$$

where $\boldsymbol{p}$ is a profile containing the radial distribution of certain quantities and $\tilde{\boldsymbol{w}}$ is pitchwise average of the quantities of interest, given, for example, by

$$\tilde{\boldsymbol{w}}_j = \frac{\sum_{i=1}^{n_i} \boldsymbol{w}_{ij} a_{ij}}{\sum_{i=1}^{n_i} a_{ij}}, \tag{9}$$

with a_{ij} being the area of the cell and i, j the pitchwise and spanwise directions, respectively.

Following the ADjoint approach, the original flow solver code was first rewritten so that it consisted in routines that could be differentiated by Tapenade to obtain the partial derivatives of the RHS of eq. (8). Currently, only the terms $\partial \tilde{\boldsymbol{w}}_{\text{don}}/\partial \boldsymbol{q}_{\text{don}}$, $\partial \boldsymbol{p}_{\text{don}}/\partial \tilde{\boldsymbol{w}}_{\text{don}}$ and $\partial \boldsymbol{p}_{\text{rec}}/\partial \boldsymbol{p}_{\text{don}}$ have been fully developed and verified. The term $\partial \boldsymbol{q}_{\text{rec}}/\partial \boldsymbol{p}_{\text{rec}}$, which represents the treatment of the boundary conditions is still a work-in-progress, as is the adaptation of the already developed routines that compute the term $\partial\mathcal{R}_{\text{rec}}/\partial\boldsymbol{q}_{\text{rec}}$.

4 CONCLUSIONS

This paper presented a description of the implementation of the mixing-plane interface on the adjoint solver of a proprietary CFD turbomachinery solver. The partial derivatives required for the assembly of the adjoint system of equations were obtained using the AD tool Tapenade. It produced routines that evaluate the derivatives, in a shorter development time than would be necessary if differentiating by hand. The adjoint mixing plane interface is still a work-in-progress at the time of writing of this paper, as the treatment of the boundary conditions as not been yet fully differentiated.

5 ACKNOWLEDGMENTS

The first author thanks Fundação para a Ciência e Tecnologia (FCT), Portugal for the Ph.D. Grant SFRH/BD/97521/2013.

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] 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.

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

[20] 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.

[21] 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.

[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.