Computers and Fluids 176 (2018) 182–192

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Adjoint formulation of a steady multistage turbomachinery interface using automatic differentiation

S.S. Rodrigues*, A.C. Marta

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Portugal

ARTICLE INFO

Article history:
Received 17 June 2018
Revised 13 September 2018
Accepted 19 September 2018
Available online 21 September 2018

Keywords:
Discrete adjoint
Mixing-plane
Algorithmic differentiation
Sensitivity analysis
Shape optimization
Endwall contouring
Operating conditions

ABSTRACT

The use of high-fidelity computational fluid dynamics (CFD) tools in turbomachinery design has seen a continuous increase as a result of computational power growth and numerical methods improvement. These tools are often used in optimization environments, where gradient-based optimization algorithms are the most common due to their efficiency. In cases where the optimization contains a large number of design variables, the adjoint approach for calculating the gradients is beneficial, as it provides a way of obtaining function sensitivities with a computational cost that is nearly independent of the number of design variables. The interaction between adjacent blade rows is of utmost importance for the performance of multistage turbomachines. The most commonly used method to address these effects (i.e. coupling 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, the formulation and implementation of an adjoint 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. The sensitivity of several performance metrics relative to neighbor blade/hub row geometry and boundary conditions are shown to highlight the physical coupling in multi-row turbomachines. The verification of the adjoint multistage solver against the finite-difference approach is performed successfully with relative differences below 1 %.

1. Introduction

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

However, given the nature of the flow models, a numerical simulation may take hours or even days to evaluate the desired performance metric functions, 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 desired optimization methods are the gradient based (GB) because of their efficiency. The GB methods, however, require the calculation of the derivatives, which, if using methods such as finite-differences, also lead to prohibitive computational and time requirements, in the case of a large number of design variables. This problem is overcome by the adjoint method, which produces exact derivatives with a cost that is nearly 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 [2] to optimization of airfoil profiles [2] and wings [3]. Since then, it has been used in solving multi-point aerodynamic shape [4] and aerostructural [5] optimization problems, magneto-hydrodynamic flow control [6] and turbine blades [7]. 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 rows, which has a significant impact on the performance of the whole multistage turbomachine [8]. Its incorporation in the optimization environment provides a more realistic insight of the direction to which the overall design should evolve. Similarly to what is already an industrial standard in turbomachinery analysis, the multistage coupling adjoint sensitivity should also become a standard in multistage turbomachinery design.

The adjoint solver can be obtained either by linearizing the underlying flow governing equations followed by their discretization

* Corresponding author.
E-mail addresses: simao.rodrigues@tecnico.ulisboa.pt (S.S. Rodrigues), andre.marta@tecnico.ulisboa.pt (A.C. Marta).

https://doi.org/10.1016/j.compfluid.2018.09.015
0045-7930/

– *continuous approach* or by the linearization of the already discretized equations – *discrete approach*. The latter can also be implemented with different approaches, either by linearizing the discrete equations manually [9] or making use of Automatic Differentiation (AD) to obtain the linearization of the computation routine [10].

Previous works in implementing adjoint solvers with multistage capabilities were done by Frey et al. [11] using finite-difference approximation to set-up the discrete adjoint system of equations, Wang and Li [12,13] following the continuous approach, Walther and Nadarajah [9] which implemented an adjoint solver using the manual discrete approach and Backhaus et al. [14] using an operator-overloading AD tool to implement the adjoint solver.

This paper describes the adjoint formulation, development and implementation of a mixing-plane boundary interface. It follows the previous work of Marta and Shankaran [15] on the implementation of the discrete adjoint counterpart of a proprietary turbomachinery CFD solver, by using a source transformation AD tool on the direct routines. The improved adjoint solver is used to obtain sensitivity analysis of various functions of interest, such as pressure ratio, efficiency and mass flow, to both the hub and blade shapes and to the inlet and exit boundary conditions of a stator-rotor turbomachinery state. The sensitivities are compared to finite-difference approximations to verify the correct implementation of the mixing-plane boundary conditions.

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 input parameters 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 some constraints. The generic CFD design problem can be formulated as

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

where $\mathcal{F}$ 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

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

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

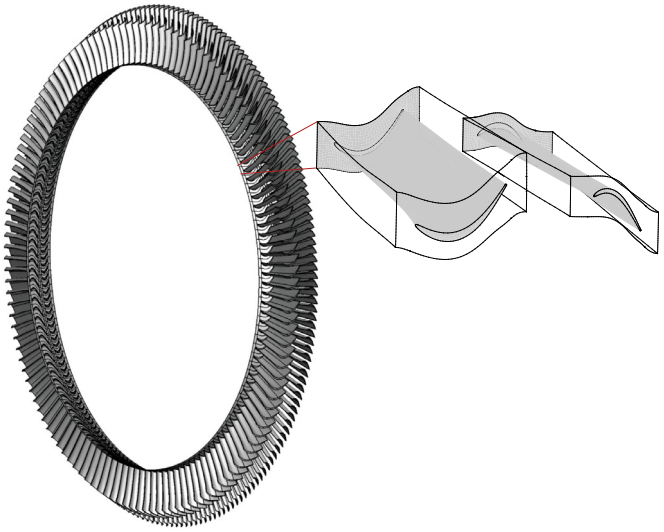


Fig. 1. Stator-rotor stage in study and computational domain using a single blade per row.

$$\boldsymbol{q} = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{Bmatrix}, \; \boldsymbol{f}_i = \begin{Bmatrix} \rho u_i \\ \rho u_1 u_i + p\delta_{i1} \\ \rho u_2 u_i + p\delta_{i2} \\ \rho u_3 u_i + p\delta_{i3} \\ \rho E u_i + p u_i \end{Bmatrix}$$

$$\text{and} \quad \boldsymbol{f}_{\mathrm{v}i} = \begin{Bmatrix} 0 \\ \tau_{ij}\delta_{i1} \\ \tau_{ij}\delta_{i2} \\ \tau_{ij}\delta_{i3} \\ u_j \tau_{ij} + q_i \end{Bmatrix}, \tag{3}$$

where ρ is the flow 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. The source term $\boldsymbol{Q}$ represents all potential body forces, such Coriolis force and centrifugal force. 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 remaining of the paper.

2.2. Multistage mixing plane

The mixing-plane method was first introduced by Denton and Singh [17] and has since become the industry standard for multi-row simulations. It is used with steady state simulations and requires only a single blade per row, as illustrated in Fig. 1 for a stator-rotor stage. The description is made for the case of an axial turbomachine but it can be easily extended to radial and mixed configurations.

Between each of the blade passages, the flow properties are circumferentially averaged in the so-called mixing-plane interface. 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

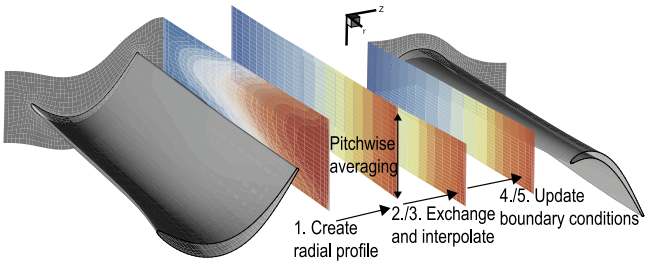


Fig. 2. Schematic representation of the multistage mixing-plane interface steps.

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 (schematically represented in Fig. 2) can be condensed in the following five steps:

1. Compute the fluxes from conserved quantities $\boldsymbol{q}$ at the mixing-plane face and create a local profile by averaging them at each spanwise position,
$$\boldsymbol{p}_{\text{local},j} = f(\tilde{\boldsymbol{q}}_j)\,; \tag{5}$$
2. Communicate the local radial profiles $\boldsymbol{p}_{\text{local}}$ of averaged quantities as donor profiles $\boldsymbol{p}_{\text{don}}$ between blade rows,
$$\boldsymbol{p}_{\text{local}} \rightarrow \boldsymbol{p}_{\text{don}}\,; \tag{6}$$
3. Interpolate the received $\boldsymbol{p}_{\text{rec}}$ profiles to match local cell distribution,
$$\boldsymbol{p}_{\text{rec}} = f(\boldsymbol{p}_{\text{don}}, \boldsymbol{p}_{\text{local}})\,; \tag{7}$$
4. Compute differences in fluxes and state variables $\boldsymbol{p}^*_{\text{rec}}$ between the interpolated and local profiles,
$$\boldsymbol{p}^*_{\text{rec}} = f(\boldsymbol{p}_{\text{rec}}, \boldsymbol{p}_{\text{local}})\,; \tag{8}$$
5. Compute the variation in the conserved variables $\boldsymbol{q}^*$ to be applied to the auxiliary cells, from the flux differences and update them,
$$\boldsymbol{q}^*_{\text{local}} = f(\boldsymbol{p}^*_{\text{rec}}, \boldsymbol{q}_{\text{local}})\,. \tag{9}$$

For simplicity, Fig. 2 only represents the transfer of information from one row to another, but this algorithm occurs in both directions for each interface.

2.3. Adjoint equations

Following the work by Giles and Pierce [20] in derivation of the adjoint equations for systems of partial differential equations (PDEs), the adjoint for the flow governing equations in Eq. (4) can be expressed as

$$\left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right]^T \boldsymbol{\psi} = \left[\frac{\partial \mathcal{F}}{\partial \boldsymbol{q}}\right]^T, \tag{10}$$

where $\boldsymbol{\psi}$ is the adjoint vector. The adjoint solution can be 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

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}\boldsymbol{\alpha}} = \frac{\partial \mathcal{F}}{\partial \boldsymbol{\alpha}} - \boldsymbol{\psi}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{\alpha}}\,. \tag{11}$$

The choice of variables in the previous equations is only limited to being able to describe the objective function $\mathcal{F}$ and residual $\boldsymbol{R}$ in terms of those variables. It should be noted that the general constrained problem Eq. (1) requires one additional adjoint system for each constraint $\mathcal{C}$ to compute the derivatives $\mathrm{d}\mathcal{C}/\mathrm{d}\boldsymbol{\alpha}$ required by the GB optimizer.

As stated, 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*. These two approaches result in different systems of linear equations that, in theory, converge to the same result with mesh refinement. However, the discrete approach brings some advantages, such as being applicable to any set of governing equations, treat arbitrary functions of interest and provide sensitivities consistent with those produced by the discretized solver. It is also easier to obtain the appropriate boundary conditions for the adjoint solver with the discrete approach, which is paramount for the purpose of this work.

2.4. Automatic differentiation

Automatic differentiation – also known as algorithmic or computational differentiation – applies the chain rule to computer programs to obtain derivatives of their outputs based on their inputs [21].

Any computational algorithm consists in a sequence of operations that can be expressed in the form

$$t_i = f_i(t_1, t_2, \ldots, t_{i-1}) \qquad i = n+1, n+2, \ldots, m\,, \tag{12}$$

where each function f_i can be either a unary or binary operation, $t_1, t_2, \ldots, t_n$ are the independent variables and $t_{n+1}, t_{n+2}, \ldots, t_m$ are the dependent variables. By applying the chain rule, the derivative of t_i in respect of t_j is given by

$$\frac{\partial t_i}{\partial t_j} = \sum_{k=1}^{i-1} \frac{\partial f_i}{\partial t_k}\frac{\partial t_k}{\partial t_j} \qquad j = 1, 2, \ldots, n\,. \tag{13}$$

AD can work in two ways, the *forward mode* and the *reverse mode*. In the first j is kept fixed and i is advanced until achieving the desired derivative. The latter works by fixing i, and decreasing j down to the independent variables. The differentiation of the code can be achieved either by *operator overloading* or *source code transformation*. Depending on the programming language and structure of the code, the use of one might bring more advantages than the other. While AD is as accurate and much easier to implement than an analytic method, the run time of the differentiated version of the algorithm in forward mode takes approximately two times longer to run than the real-valued version, taking even longer in backward mode. Another limitation of this method is that more complex codes may be difficult, or even impossible to differentiate in one go, due to the limitations of the AD tool.

2.5. ADjoint method

This hybrid approach consists in computing the total derivative with the adjoint method but having the partial derivatives of Eqs. (10) and (11), $\partial \boldsymbol{R}/\partial \boldsymbol{q}$, $\partial \mathcal{F}/\partial \boldsymbol{q}$, $\partial \mathcal{F}/\partial \boldsymbol{\alpha}$ and $\partial \boldsymbol{R}/\partial \boldsymbol{\alpha}$, evaluated with automatically differentiated routines. To do so, 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 [10]. This routine is then differentiated using an AD tool, thus producing the necessary terms for the calculation of the adjoint solution.

2.6. Adjoint multistage interface

Assuming a simulation of a series of n blade rows, each blade will influence and be influenced by its neighbors. If no multistage interface were used, a system of equations Eq. (10) would be solved for each row independently. However, to consider the influence of the rows on each other the coupled systems of equations

must be solved,

$$\begin{bmatrix} \left[\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{q}_1}\right] \cdots \left[\frac{\partial \boldsymbol{R}_1}{\partial \boldsymbol{q}_n}\right] \\ \vdots \quad \ddots \quad \vdots \\ \left[\frac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{q}_1}\right] \cdots \left[\frac{\partial \boldsymbol{R}_n}{\partial \boldsymbol{q}_n}\right] \end{bmatrix} \begin{Bmatrix} \boldsymbol{\psi}_1 \\ \vdots \\ \boldsymbol{\psi}_n \end{Bmatrix} = \begin{Bmatrix} \frac{\partial \mathcal{F}}{\partial \boldsymbol{q}_1} \\ \vdots \\ \frac{\partial \mathcal{F}}{\partial \boldsymbol{q}_n} \end{Bmatrix} . \tag{14}$$

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 neighbors, therefore

$$\left[\frac{\partial \boldsymbol{R}_i}{\partial \boldsymbol{q}_j}\right] = 0 \,, \quad i-1 > j > i+1 \,. \tag{15}$$

The chain rule can be applied in the computation of the coupling non-zero off-diagonal terms to distinguish the *single-row* term $\partial \boldsymbol{R}_{\text{rec}} / \partial \boldsymbol{q}^*_{\text{local}}$ from a term that represents the influence of the state solution of the adjacent domain, $\boldsymbol{q}_{\text{don}}$ on the updated state solution $\boldsymbol{q}^*_{\text{local}}$, thus obtaining

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{don}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}^*_{\text{local}}} \frac{\mathrm{d}\boldsymbol{q}^*_{\text{local}}}{\mathrm{d}\boldsymbol{q}_{\text{don}}} \,. \tag{16}$$

Recalling the mixing-plane algorithm description, the subscript "rec" and "don" indicate, from the point of view of a single row, if the term belongs to the same row or the adjacent row, respectively.

Recalling the multistage interface described in Section 2.2, the updated state solution can be represented as a function of the various terms computed during the mixing-plane step,

$$\boldsymbol{q}^*_{\text{local}} = f\Big(\boldsymbol{p}^*_{\text{rec}}\big(\boldsymbol{p}_{\text{don}}(\boldsymbol{q}_{\text{don}}), \boldsymbol{p}_{\text{local}}(\boldsymbol{q}_{\text{local}})\big), \boldsymbol{q}_{\text{local}}\Big) \,. \tag{17}$$

An expression for the multistage coupling term can be obtained by differentiating each of the terms identified above and applying the chain rule, thus obtaining

$$\frac{\mathrm{d}\boldsymbol{q}^*_{\text{local}}}{\mathrm{d}\boldsymbol{q}_{\text{don}}} = \frac{\partial \boldsymbol{q}^*_{\text{local}}}{\partial \boldsymbol{p}^*_{\text{rec}}} \frac{\partial \boldsymbol{p}^*_{\text{rec}}}{\partial \boldsymbol{p}_{\text{don}}} \frac{\partial \boldsymbol{p}_{\text{don}}}{\partial \boldsymbol{q}_{\text{don}}} \,. \tag{18}$$

The previous expression regards only the dependence on the cells across the multistage interface. However, there is also a new dependency on the cells of the local face due to the multistage boundary condition

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}_{\text{local}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}^*_{\text{local}}} \frac{\mathrm{d}\boldsymbol{q}^*_{\text{local}}}{\mathrm{d}\boldsymbol{q}_{\text{local}}} \,, \tag{19}$$

where the coupling term is obtained from

$$\frac{\mathrm{d}\boldsymbol{q}^*_{\text{local}}}{\mathrm{d}\boldsymbol{q}_{\text{local}}} = \frac{\partial \boldsymbol{q}^*_{\text{local}}}{\partial \boldsymbol{p}^*_{\text{rec}}} \frac{\partial \boldsymbol{p}^*_{\text{rec}}}{\partial \boldsymbol{p}_{\text{local}}} \frac{\partial \boldsymbol{p}_{\text{local}}}{\partial \boldsymbol{q}_{\text{local}}} + \frac{\partial \boldsymbol{q}^*_{\text{local}}}{\partial \boldsymbol{q}_{\text{local}}} \,. \tag{20}$$

The first term of the RHS of Eq. (20) increases the stencil of the residual calculation to cover at least a whole row of cells in the radial position of each cell of the single stage stencil that belongs to the multistage interface. The second term comes from the non-reflectivity boundary conditions and also increases the stencil of influence to cover a certain number of radial rows of the mixing-plane face.

To obtain the total derivative given by Eq. (11) it is also necessary to compute $\partial \boldsymbol{R} / \partial \boldsymbol{\alpha}$ with the coupling taken into account. The present work assumes inlet and outlet boundary conditions $\boldsymbol{U}$ and computational grid coordinates $\boldsymbol{X}$ as possible design variables $\boldsymbol{\alpha}$, representing operating conditions and blade/hub shape, respectively. For the case of boundary conditions $\boldsymbol{U}$, nothing has to be done regarding the mixing-plane interface, since the inlet and outlet surfaces are either the first inlet or last outlet of the coupled domains.

There is, however, a dependence of the updated state on the grid coordinates $\boldsymbol{X}$ of the adjacent domain. Therefore, it is necessary to take the multistage coupling into account in its computation, if the grid coordinates are chosen as the design variables. In this case, the term $\partial \boldsymbol{R} / \partial \boldsymbol{X}$ is given by

$$\frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{X}_{\text{don}}} = \frac{\partial \boldsymbol{R}_{\text{rec}}}{\partial \boldsymbol{q}^*_{\text{local}}} \frac{\mathrm{d}\boldsymbol{q}^*_{\text{local}}}{\mathrm{d}\boldsymbol{X}_{\text{don}}} \,. \tag{21}$$

where the term $\mathrm{d}\boldsymbol{q}^*_{\text{local}} / \mathrm{d}\boldsymbol{X}_{\text{don}}$ reflects the dependency of the local state solution on the computational grid coordinates of the donor cells.

3. Implementation

Some details of the implementation of the coupling multistage interface previously described are presented next.

3.1. Flow solver

The legacy flow solver TACOMA, is capable of solving the steady or unsteady RANS using a finite-volume formulation [22]. 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. The solver is able to run in multiple processes via Message Passing Interface (MPI).

3.2. Adjoint solver

The discrete adjoint solver was previously implemented by using the ADjoint hybrid approach [10]. The AD tool chosen in the mentioned work, as well as in the present work, was Tapenade [23], 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) [24] is used to solve the adjoint system in Eq. (10), more specifically, the generalized minimum residual method (GMRES) with the incomplete factorization preconditioner with one level fill, ILU(1), are used. The turbulence equations were fully handled in the discrete adjoint formulation, despite having the option to run the adjoint solver with frozen turbulence [25].

Following the same ADjoint approach used to develop the adjoint solver, the adjoint mixing-plane interface is implemented by differentiating the rewritten subroutines responsible to compute the quantities described in Section 2.6 to obtain the terms $\mathrm{d}\boldsymbol{q}^*_{\text{local}} / \mathrm{d}\boldsymbol{q}_{\text{don}}$ and $\mathrm{d}\boldsymbol{q}^*_{\text{local}} / \mathrm{d}\boldsymbol{q}_{\text{local}}$.

The original main routine `ms_exchange` responsible for the mixing-plane interface, as schematically detailed in Fig. 3, could not be differentiated due to its complexity, namely dependency on data from Fortran modules that could not be used as explicit dependent variables for differentiation, use of MPI communications and use of dynamically allocated data structures that could not be differentiated. Instead, it was divided into rewritten subroutines represented by the previously described individual steps, as detailed in Fig. 4, and each of those subroutines (in green) was then differentiated using Tapenade. The general rule was to remove MPI communications from routines to be differentiated while trying to the data flow as close to the original as possible and rewrite the various subroutines to take de dependent and independent variables as explicit arguments. The MPI communications were then manually added in the merge of the rewritten subroutines into the main routine `adj_ms_exchange`.

Subroutine `adj_ms_preprocess` creates all the required data structures to be used in the computation of the profiles and boundary condition updates; `adj_ms_comp_prof_1`, `adj_ms_accumulate` and `adj_ms_comp_prof_2` result from splitting `ms_compute_profile` into the local accumulation on each process (producing `stAdjLocal`), accumulation to the profile owner (producing `stAdjGlobal`) and normalization of the profiles (producing `prAdj_loc`, or $\boldsymbol{p}_{\text{local}}$),

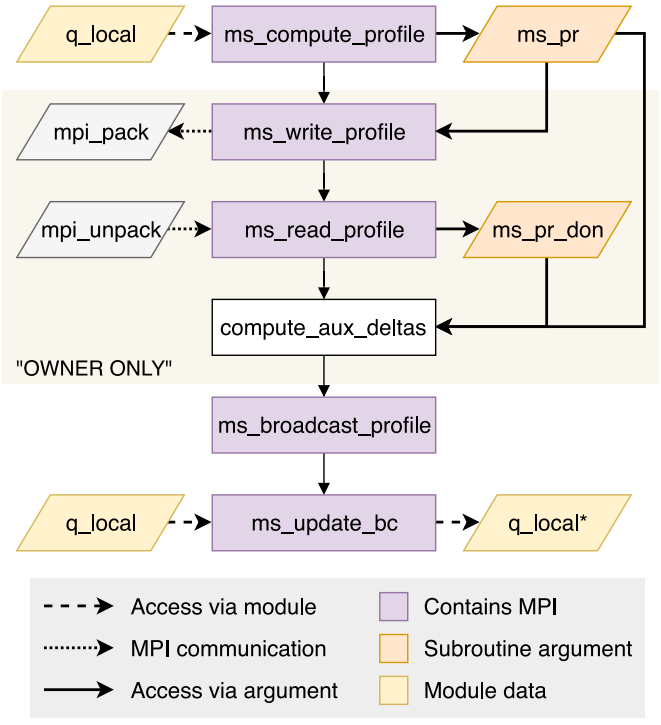


Fig. 3. Schematic of the original mixing-plane interface routine `ms_exchange` in flow solver.

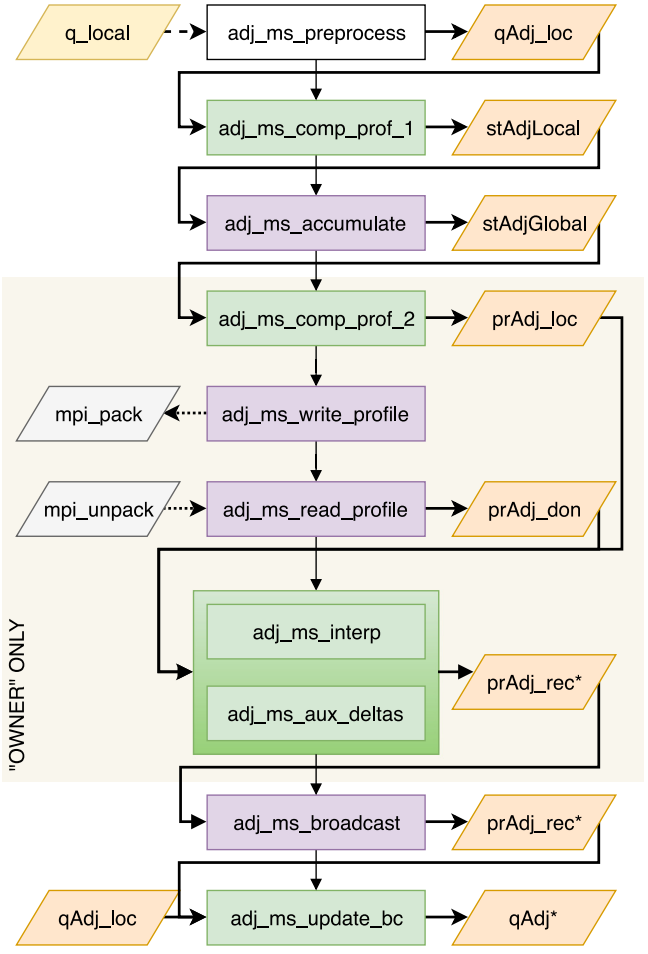


Fig. 4. Schematic of the re-written mixing-plane interface routine `adj_ms_exchange`.

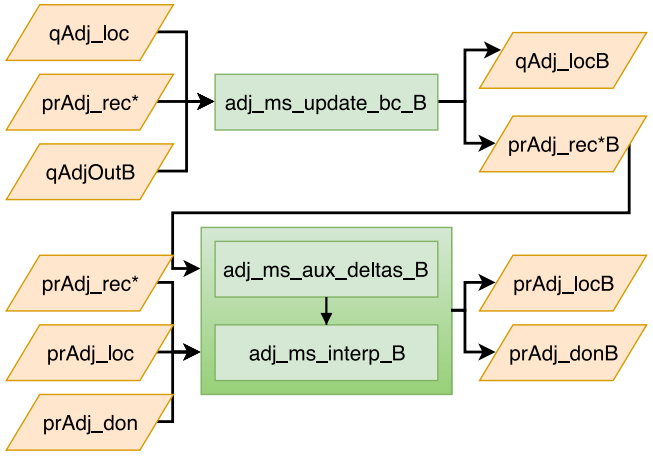


Fig. 5. Partial schematic of the manually assembled subroutine `adj_ms_exchange_b`.

respectively. The interpolation of the profile received from the neighbor, originally performed in the `ms_read_profile` was removed to the routine `adj_ms_interp` and merged with the subroutine `adj_ms_aux_deltas` into a subroutine than computes all the required deltas between the local and received profiles (`prAdj_loc` and `prAdj_don`), producing `prAdj_rec*` (or $\boldsymbol{p}^{)}_{\text{rec}*}$ Recalling the mixing-plane algorithm steps mentioned in Section 2, step 1 is performed by `adj_ms_comp_prof_1`, `adj_ms_accumulate` and `adj_ms_comp_prof_2`, step 2 is performed by `adj_ms_write_profile` and `adj_ms_read_profile`, step 3 and 4 by `adj_ms_interp` and `adj_ms_aux_deltas`, respectively, and step 5 by `adj_ms_update_bc`.

The data structure `stAdj` contains the radial local accumulation of the quantities needed to create the profile and `prAdj` is the modified structure containing the profile. This modified structure was implemented due to Tapenade not being able to differentiate structures containing Fortran's `allocatable` arrays. With the modified structures, instead of having one structure element with various arrays, `pr%array1(:), ..., pr%arrayN(:)`, we have an array of structures containing fixed size elements, `prAdj(:)%value1, ..., prAdj(:)%valueN)`.

This rewritten routine `adj_ms_exchange` is only required for the differentiation procedure and does not replace the original subroutine in the direct solver. An extensive verification was performed to ensure it delivered the same results as the original.

The final assembly of the differentiated main routine `adj_ms_exchange_b` was performed manually, once again following the chain rule of differentiation and by using the derivative obtained from one subroutine as seed for the next differentiated subroutine call, as partially represented in Fig. 5. The MPI communications were dealt similarly to the routine `adj_ms_exchange`. This routine `adj_ms_exchange_b` takes as inputs the state at the mixing-plane surface `qAdj`, and the seed `qAdjB`, both $N_i \times N_j \times N_v$ sized arrays, where i and j represent the two dimensions of the face and N_v the number of state variables, which selects the cell(s) to which the sensitivity to the adjacent row face cells is to be computed. Its output is the sensitivity to the neighbor face quantities. It is worth mentioning that further enhancements/modifications to the original code would have to be manually propagated to the rewritten code, which would then be differentiated and propagated to the adjoint solver. For modifications that do not introduce large changes to the structure of the code, this is a rather straightforward process.

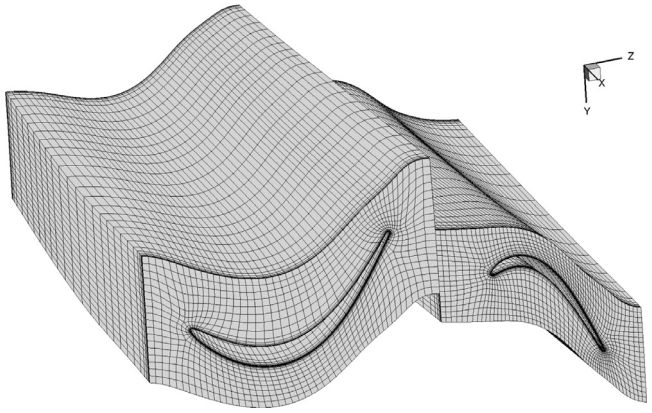


Fig. 6. Computational mesh used in the present work.

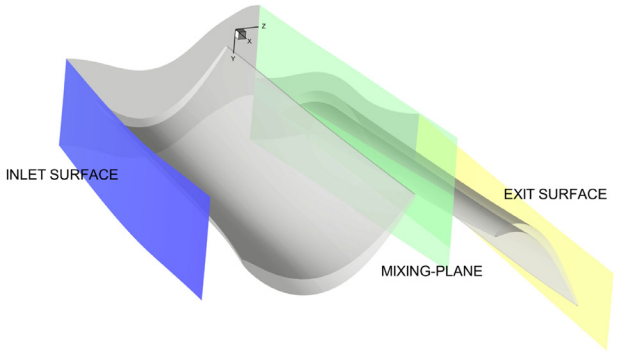


Fig. 7. Schematic representation of the inlet and exit boundary surfaces of the stator-rotor stage domain.

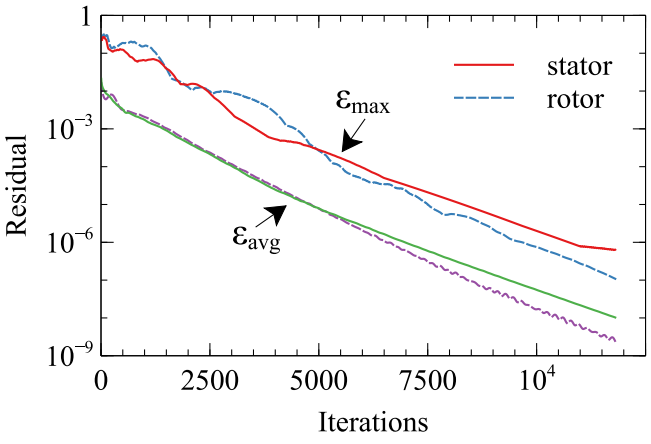


Fig. 8. History of residual of flow solution during convergence.

With the differentiated mixing-plane subroutine producing the mixing-plane coupling terms $\mathrm{d}\boldsymbol{q}^*_{\mathrm{local}}/\mathrm{d}\boldsymbol{q}_{\mathrm{don}}$ and $\mathrm{d}\boldsymbol{q}^*_{\mathrm{local}}/\mathrm{d}\boldsymbol{q}_{\mathrm{local}}$, their integration with the single stage solver is performed in an additional step of the assembly of the Jacobian $\partial\boldsymbol{R}/\partial\boldsymbol{q}$ in Eq. (10). In this additional step, the sensitivity of the state variable of each cell whose dependency stencil contains any cell in a shared boundary is used as seed to the differentiated mixing-plane routine, according to the chain rule of differentiation in Eq. (16). The resulting partial derivatives are then inserted into the respective row of the Jacobian matrix.

4. Results

This section presents various results of the analysis of a multi-row stator-rotor stage of a low pressure turbine and their comparison against finite-difference approximations to verify the correct implementation of the adjoint multistage interface.

4.1. Description of the test case

The test case consists in a stator-rotor stage of a low pressure turbine. Both the stator and rotor are modeled with a single blade passage, using periodic boundary conditions. Each domain is discretized with an OH-grid topology, with a total of 90,750 cells amongst the two domains. The computational mesh used in the present work is presented in Fig. 6.

Absolute total enthalpy h_{Ta}^{inlet} and pressure p_{Ta}^{inlet}, tangential velocity v_t^{inlet} and flow angle ϕ^{inlet} are imposed at the inlet of the stator, with extrapolation of the pressure to the interior. Static pressure p_s^{exit} is held fixed at the exit of the rotor. Between the two domains, the boundary conditions are updated with the mixing-plane algorithm with exchange of boundary fluxes. The inlet, outlet and mixing-plane surfaces are represented in Fig. 7. For the boundary layer, wall-functions were used to keep the mesh relatively coarse.

Four different functions of interest $\mathcal{F}$ are considered: exit mass flow ($\dot{m}^{\mathrm{out}}$), pressure ratio of stator (π_1), pressure ratio of rotor (π_2) and rotor efficiency (η_2). The mass flow is computed at the exit of the last (rotor) blade passage. The pressure ratio computed at the stator is given by

$$\pi_1 = \frac{p_{Ta}^{\mathrm{inlet},1} - p_{Ta}^{\mathrm{exit},1}}{p_{Ta}^{\mathrm{inlet},1}} \times 100\,, \tag{22}$$

where the total pressure is area averaged. The pressure ratio computed at the rotor is given by

$$\pi_2 = \frac{p_{Ta}^{\mathrm{exit},2}}{p_{Ta}^{\mathrm{inlet},2}}\,, \tag{23}$$

with the total pressure, in this case, being enthalpy averaged. Finally, the rotor efficiency given by

$$\eta_2 = \frac{\left(p_{Ta}^{\mathrm{exit},2}/p_{Ta}^{\mathrm{inlet},2}\right)^{(\gamma-1)/\gamma} - 1}{\left(T_{Ta}^{\mathrm{exit},2}/T_{Ta}^{\mathrm{inlet},2}\right) - 1}\,, \tag{24}$$

where the total pressure is enthalpy averaged and the total temperature is mass averaged. The subscript *Ta* refers to total absolute quantity. The inlet and exit surfaces are respective to the respective blade passage, meaning that for π_1 the exit surface is at the mixing-plane interface identified in Fig. 7, as is the inlet surface of π_2 and η_2.

4.2. Flow and adjoint solutions

The flow solution was converged to an averaged residual of the continuity equation of 10^{-9}, as observed from the history of the residual, presented in Fig. 8. For the adjoint solutions, the convergence criterium was a relative difference in the magnitude of the residual between iterations of 10^{-9}. The history of the residual during the GMRES iterations is presented in Fig. 9. The restart of the GMRES was set at 75 iterations, which is visible in the residual history. All the computations were performed in double precision.

The converged flow solution and corresponding adjoint solution (computed with $\mathcal{F} = \dot{m}^{\mathrm{out}}$) are presented, for the case of the density, in Fig. 10 and Fig. 11, respectively. The next subsections presents the solutions of the adjoint system of equations for the four different functions of interest described in Section 4.1. These functions are computed either at the exit surface alone (for the case of exit mass flow) or at the exit/inlet surfaces together with the mixing-plane surface (refer to Fig. 7). From the many sensitivities possible to be evaluated by the multistage adjoint solver, five

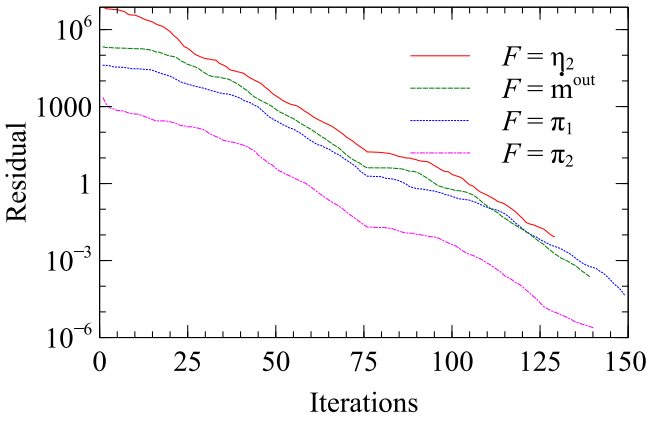


Fig. 9. History of residuals of adjoint solutions during GMRES iterations.

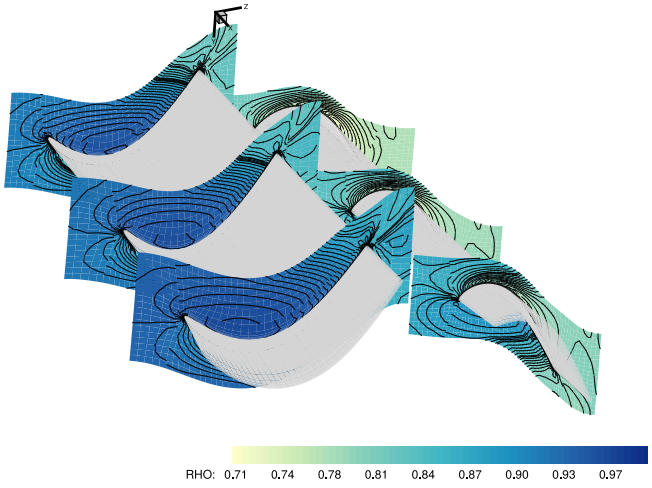


Fig. 10. Normalized flow solution of the continuity equation.

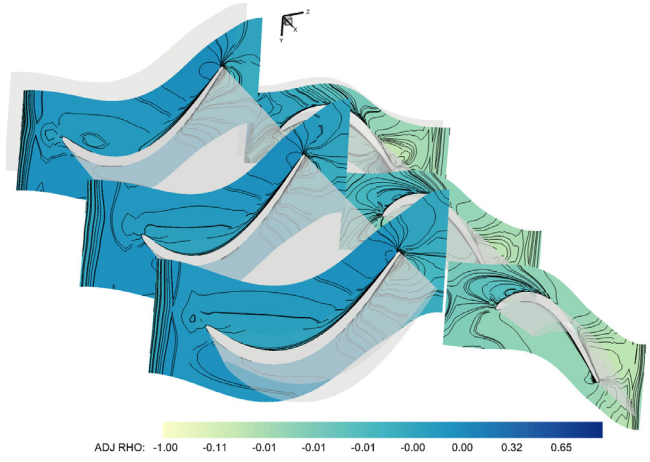


Fig. 11. Normalized adjoint solution of the continuity equation for exit mass flow as the objective function.

combinations of function of interest $\mathcal{F}$ and design parameter $\boldsymbol{\alpha}$ were selected, as summarized in Table 1, where the superscripts in the variables denote inlet of stator or exit of rotor and the subscripts in the functions denote stator (row 1) and rotor (row 2). The variables r_{hub} and n_{blade} correspond to the radial coordinate of the hub geometry and to the local normal coordinate of the blade geometry, respectively.

Table 1
Selection of test cases included.

$\frac{d\mathcal{F}}{d\boldsymbol{\alpha}}$	π_1	π_2	η_2	$\dot{m}^{out}$
h_{Tq}^{inlet}	–	–	–	–
p_{Tq}^{inlet}	–	–	–	✓
ν_t^{inlet}	–	–	✓	–
ϕ^{inlet}	–	–	–	–
p_s^{exit}	✓	–	–	–
r_{hub}	–	✓	–	–
n_{blade}	–	–	✓	–

Table 2
Comparison of computational requirements of direct and adjoint solvers (normalized by the direct solver).

	CPU time	Memory
Direct	1	1
Adjoint	∼1	∼10
Preprocess (1)	0.3%	–
Assemble $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}$, $\frac{\partial \mathcal{F}}{\partial \boldsymbol{q}}$ (2)	51.1%	∼50%
GMRES Solver (3)	42.2%	∼50%
Assemble $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{U}}$, $\frac{\partial \mathcal{F}}{\partial \boldsymbol{U}}$ (4)	6.2%	–
Compute Sensitivity (5)	0.1%	–
I/O (6)	0.1%	–

The selected adjoint-based sensitivities, presented in the next subsection, were verified against first-order forward finite-difference approximations. For these verifications, four nodes were randomly selected at the inlet and exit boundary faces of the stator-rotor stage, as well as four at the hub of the stator and four at the surface of the stator blade (two on the pressure and two on the suction side). The flow solver was run to convergence and for every design variables $\boldsymbol{\alpha}$ perturbation imposed on each control node. This would lead in theory to 20 runs (five variables times four control points) but, in practice, the number of runs was much higher, as the finite-difference approximation is highly sensitive to the perturbation step size and a manual search had to be performed to obtain the optimal step for each control node to obtain a good trade-off between truncation error due to large step sizes and subtractive cancellation due to too small perturbations.

A comparison of the computational requirements of the direct and adjoint solvers is presented in Table 2. While the CPU time requirements of computing both solutions is approximately the same, the memory requirements of the adjoint solver increase ten-fold compared to the direct solver. This is a direct consequence of the solution method chosen for the adjoint system of equations, that includes a full matrix storage and the iterative GMRES method. This effect could be considerably mitigated if matrix-free algorithms [26] or pseudo-time marching Runge–Kutta methods [27] were employed instead. The table also presents a detailed description of the memory and CPU time required by each of the processes of computing the adjoint solution, namely the 1) preprocessing, 2) assembling of the system of equations, 3) computing the solution with the GMRES solver, 4) assembling the matrices/vectors to compute the total derivatives, 5) computing the total derivatives and 6) Output of the solutions to files. As observed, the assembly of the matrices and the solution of the adjoint system of equations take the bulk of the required CPU time and memory, being roughly the same for each part.

4.3. Sensitivity of stator pressure ratio π_1

The normalized adjoint-based sensitivity of stator pressure ratio π_1 to exit static pressure p_s^{exit} boundary condition of the rotor domain can be observed in Fig. 12, where the contour plot is shown for the rotor exit surface. It is worth noting that the sensitivity is

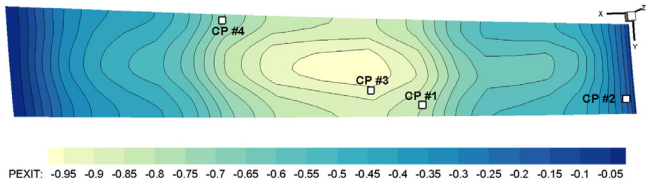


Fig. 12. Normalized adjoint-based sensitivity of stator pressure ratio π_1 to exit static pressure boundary condition.

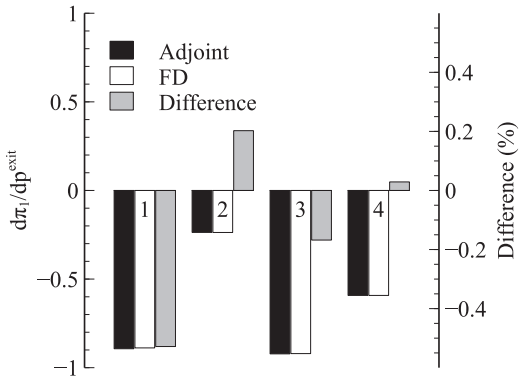


Fig. 13. Verification of adjoint-based sensitivities of stator pressure ratio to exit static pressure using FD (normalized values).

always negative, implying that stator pressure ratio would decrease with an increase of the static exit pressure, as would be expected. This effect is largest at the exit section midspan, away from the hub and casing walls and aligned with the rotor wake. This probably means that the effect is amplified when reducing the blockage effect of the blade.

The results of the verification of the adjoint-based sensitivity of the four control nodes identified in Fig. 12 are presented in Fig. 13. The sensitivities computed by both methods differ by less than 0.5%, which attests the correct numerical implementation of the adjoint mixing-plane interface.

4.4. Sensitivity of rotor pressure ratio π_2

Fig. 14 presents the normalized sensitivity of the pressure ratio of the rotor π_2 to the radial position of the hub wall r_{hub}. This radial sensitivity was obtained from the sensitivity to the hub x- and y-coordinates as

$$\frac{d\pi_2}{dr} = \frac{d\pi_2}{dx}\frac{dx}{dr} + \frac{d\pi_2}{dy}\frac{dy}{dr} = \frac{d\pi_2}{dx}\cos(\theta) + \frac{d\pi_2}{dy}\sin(\theta), \tag{25}$$

where θ is the tangential angle in cylindrical coordinates measured from the x to the y-axis. This sensitivity information can be extremely important in hub and/or casing shape optimization processes, often referred as *endwall contouring*, which can lead to significant performance improvement of the turbomachine by significantly impacting the secondary flows [28]. From Fig. 14, it can be inferred that the rotor pressure ratio can be increased by contouring the hub wall in different ways: making humps on the positive derivative regions and/or making recessions at the negative derivative regions. The multi-row coupling manifests itself in this example since there is a clear effect of stator hub endwall contouring on the rotor pressure ratio. Some oscillations, visible near the inlet and outlet of the stage, may result from the pointwise perturbation of the mesh and from non-reflectivity only being enforced at the mixing plane and not at the inlet and exit of the stage. A designer (or an optimizer) would not be interested in perturbing the mesh directly, but in changing a set of design parameters $\boldsymbol{\beta}$ that would represent the geometry/deformation by some method of parameterization, which would introduce some smoothing to the perturbation and thus smoothing the unwanted oscillations. The sensitivity information of the performance metrics to the design parameters $\boldsymbol{\beta}$ would be obtained using the adjoint-based sensitivity information to the mesh as

$$\frac{d\mathcal{F}}{d\boldsymbol{\beta}} = \frac{d\mathcal{F}}{d\boldsymbol{x}}\frac{d\boldsymbol{x}}{d\boldsymbol{\beta}} + \frac{d\mathcal{F}}{d\boldsymbol{y}}\frac{d\boldsymbol{y}}{d\boldsymbol{\beta}} + \frac{d\mathcal{F}}{d\boldsymbol{z}}\frac{d\boldsymbol{z}}{d\boldsymbol{\beta}}. \tag{26}$$

Depending on the tool used to generate the flow grid mesh ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$,) from the geometry parameterization $\boldsymbol{\beta}$, the sensitivities $d\boldsymbol{x}/d\boldsymbol{\beta}$, $d\boldsymbol{y}/d\boldsymbol{\beta}$ and $d\boldsymbol{z}/d\boldsymbol{\beta}$ can be obtained by either analytic methods (if source code is available) or by FD approximations (if a blackbox is used).

The results of the verification the sensitivity of π_2 to the hub wall x-coordinates are shown in Fig. 15, exhibiting again very good agreement with the FD approximation, with relative differences below 0.9%. Similar results were obtained for the y-coordinates, present in Eq. (25).

4.5. Sensitivity of rotor efficiency η_2

Fig. 16 presents the adjoint based sensitivity of the rotor efficiency η_2 to the shape of the blade. In this case, the contour shown is the magnitude of the sensitivity vector projected onto the blade surface outer normal at each point, evaluated as

$$\frac{d\pi_2}{dn} = \frac{d\pi_2}{dx}\frac{dx}{dn} + \frac{d\pi_2}{dy}\frac{dy}{dn} + \frac{d\pi_2}{dz}\frac{dz}{dn} = \frac{d\pi_2}{dx}n_x + \frac{d\pi_2}{dy}n_y + \frac{d\pi_2}{dz}n_z, \tag{27}$$

where the surface outer normal unit vector is given by $\boldsymbol{n} = (n_x, n_y, n_z)$. Similarly to the hub contouring test case, this test case also demonstrates the coupling between rows by quantitatively showing the impact of the stator blade shape on the rotor efficiency. Analyzing Fig. 16, it can be seen that the rotor efficiency can be increased by moving the stator blade in the positive (negative) outer normal direction at the regions of positive (negative) derivatives.

The control nodes used for the verification procedure are also identified Fig. 16, two at the suction side and two at the pressure side of the blade. The derivatives of rotor efficiency with respect to the y-coordinate of the selected control nodes of the stator blade surface were compared to finite-differences and the results are shown in Fig. 17 (note that the values are normalized to the maximum value of the derivative at both row domains). Good agreement is again obtained, with a maximum relative difference of 1.1%. As with the hub control nodes, a good agreement with finite-differences was also obtained for the other two coordinates.

Another example of coupling is shown in Fig. 18 which shows the effect of inlet tangential velocity on the stator efficiency, represented on the inlet plane. From the contour plot Fig. 18, the stator efficiency sensitivity to the inlet tangential velocity varies considerably depending on both the radial and tangential location. This variation is particularly strong (positive) closer to the hub and at midspan (negative). Such rich information can be extremely useful with analyzing turbomachines at off-design conditions, such as when inflow distortion occurs [29].

Fig. 19 presents the comparison of the adjoint-based sensitivities to FD approximations. In this case, the optimum step for the FD approximation was harder to obtain, particularly in control node 2, where the minimum relative error we were able to obtain was approximately 3.5%. This was probably due to the differences of order of magnitude of the derivatives $\mathcal{O}(10^{-6})$, function

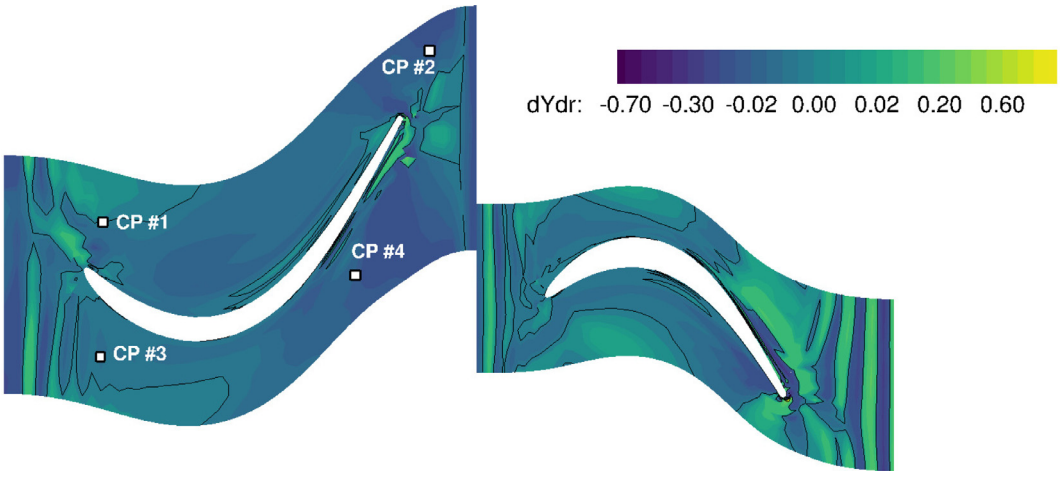


Fig. 14. Normalized adjoint-based sensitivity of rotor pressure ratio to hub wall radial position.

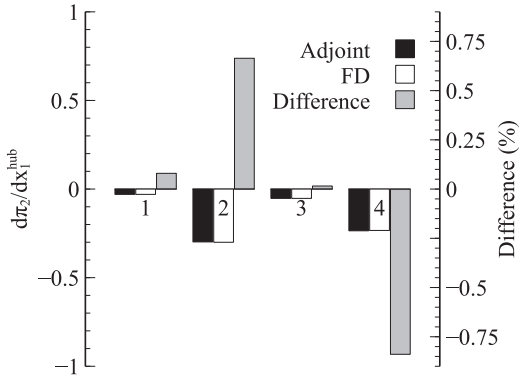


Fig. 15. Verification of adjoint-based sensitivities of rotor pressure ratio to hub wall grid *x*-coordinates using FD (normalized values).

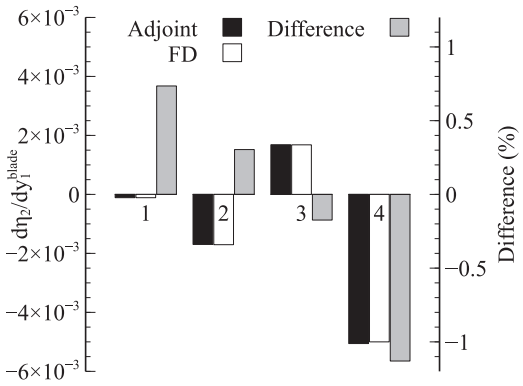
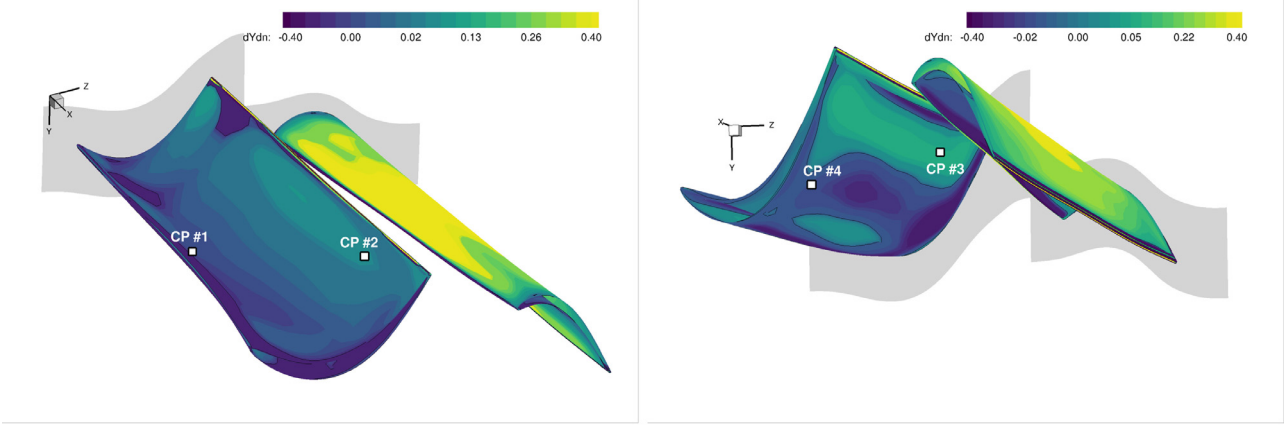


Fig. 17. Verification of adjoint-based sensitivities of rotor efficiency to stator blade surface grid *y*-coordinates using FD (normalized values).

of interest $\mathcal{O}(10^1)$ and quantity to perturb $\mathcal{O}(10^2)$. The difference in magnitudes is not visible in the bar plots due to normalization. This difficulty highlights the advantages of the adjoint method over the FD method, as the adjoint-based sensitivities avoid the concept of perturbation step altogether.

4.6. Sensitivity of exit mass flow $\dot{m}^{out}$

The sensitivity of the mass flow at the exit of the rotor to the total pressure boundary condition imposed at the inlet of the rotor is presented in Fig. 20. The values are normalized by the maximum absolute value of the derivative. The positive derivative, exhibited in almost all inlet section locations, reveals the expected increase of mass flow with the increase of inlet total pressure.



(a) Suction side (b) Pressure side

Fig. 16. Adjoint based sensitivity of rotor efficiency η_2 to blade shape in normal direction.

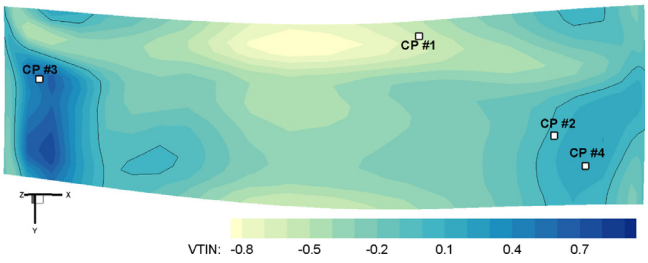


Fig. 18. Normalized adjoint based sensitivity of rotor efficiency, η_2 to inlet tangential velocity, v_t^{inlet} .

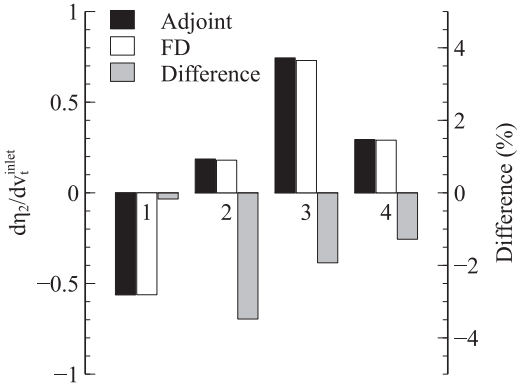


Fig. 19. Verification of adjoint-based sensitivities of rotor efficiency to tangential velocity inlet boundary conditions using FD (normalized values).

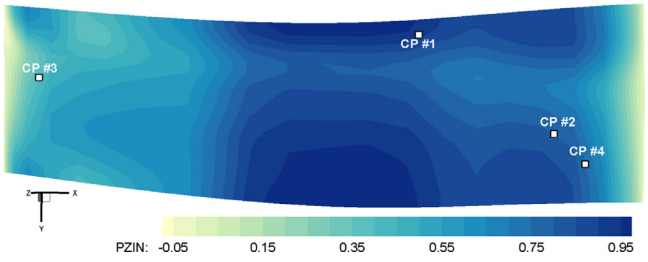


Fig. 20. Normalized adjoint based sensitivity of outlet mass flow, $\dot{m}^{\text{out}}$ to inlet total pressure, p_{Ta}^{inlet} .

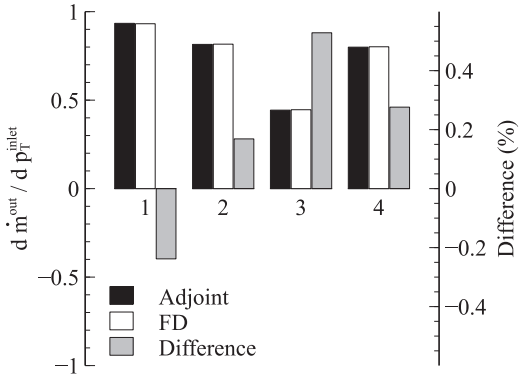


Fig. 21. Verification of adjoint-based sensitivities of mass flow to total pressure inlet boundary conditions using FD (normalized values).

The adjoint-based derivative values also show good agreement with the FD approximation, as seen in Fig. 21, where the results of the verification presented for the four control nodes exhibit differences smaller than 0.5%.

5. Conclusions

The formulation of the discrete adjoint mixing-plane was developed and implemented in a proprietary multistage turbomachinery CFD solver, using automatic differentiation tools to compute coupling terms of the discrete adjoint equations. The multistage adjoint solution was computed for a stator-rotor case and the final derivative of selected function of interest with respect to the boundary conditions at the inlet and exit of the stator-rotor stage, as well as with respect to the geometry of the blades and hub, were presented and compared with finite-difference approximations. Good agreement was obtained between the two approaches, with relative differences typically below 1% for most of the verification cases.

While the adjoint method required one additional solver run for each function of interest, with a computational cost similar to flow direct run in terms of CPU time, the finite-difference approach required many direct solver runs to obtain a converged value, thus emphasizing the benefits of using the adjoint method for sensitivity analysis.

The importance of a coupled multistage turbomachinery analysis and design is highlighted by the selected results presented, which clearly demonstrate the physical flow coupling between adjacent blade rows. By using the adjoint solver with handling of the multistage interface between adjacent rows, it is possible to efficiently and accurately quantify the impact of: i) boundary inlet conditions to downstream blade rows performance; ii) boundary exit conditions to upstream rows performance; iii) upstream blade or hub/casing shapes to downstream row performance; and iv) downstream blade or hub/casing shapes to upstream row performance.

The proposed inclusion of adjoint multistage handling in a gradient-based multistage turbomachinery design framework is thus paramount to achieve the best overall results, both in terms of computational cost but mainly in terms of optimal design outcome.

Acknowledgments

This work was supported by the Fundação para a Ciência e Tecnologia (FCT), Portugal, through the scholarship SFRH/BD/97521/2013; and through IDMEC (Institute of Mechanical Engineering), under LAETA, project UID/EMS/50022/2013. The authors would like to thank Dr. Sriram Shankaran, from GE Aviation, for his support and feedback. Moreover, the authors are thankful to General Electric for giving permission to publish this paper.

References

[1] Pironneau O. On optimum design in fluid mechanics. J Fluid Mech 1974;64(01):97–110. doi:10.1017/S0022112074002023.
[2] Jameson A. Aerodynamic design via control theory. J Sci Comput 1988;3(3):233–60. doi:10.1007/BF01061285.
[3] Jameson A, Martinelli L, Pierce NA. Optimum aerodynamic design using the Navier–Stokes equations. Theor Comput Fluid Dyn 1998;10(1–4):213–37. doi:10.1007/s001620050060.
[4] Reuther JJ, Jameson A, Alonso JJ, Rimllnger MJ, Saunders D. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 22. J Aircr 1999;36(1):61–74. doi:10.2514/2.2414.
[5] Martins JR, Alonso JJ, Reuther JJ. High-fidelity aerostructural design optimization of a supersonic business jet. J Aircr 2004;41(3):523–30. doi:10.2514/1.11478.
[6] Marta AC, Alonso JJ. Toward optimally seeded airflow on hypersonic vehicles using control theory. Comput Fluids 2010;39(9):1562–74. doi:10.1016/j.compfluid.2010.05.009.
[7] Marta AC, Shankaran S, Wang Q, Venugopal P. Interpretation of adjoint solutions for turbomachinery flows. AIAA J 2013;51(7):1733–44. doi:10.2514/1.J052177.
[8] Denton JD. The calculation of three-dimensional viscous flow through multistage turbomachines. J Turbomach 1992;114(1):18–26. doi:10.1115/1.2927983.

[9] Walther B, Nadarajah S. An adjoint-based optimization method for constrained aerodynamic shape design of three-dimensional blades in multi-row turbomachinery configurations. In: Proceedings of the ASME turbo expo 2014: power for land, sea, and air, Duesseldorf, Germany; V02BT39A031; 2014. https://doi.org/10.1115/GT2014-26604.
[10] Marta AC, Mader CA, Martins JRRA, Van der Weide E, Alonso JJ. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. Int J Comut Fluid Dyn 2007;21(9–10):307–27. doi:10.1080/10618560701678647.
[11] Frey C, Kersken H-P, Nurnberger D. The discrete adjoint of a turbomachinery RANS solver. In: Proceedings of the ASME turbo expo 2009: power for land, sea, and air, Orlando, USA; 2009. p. 345–54. doi:10.1115/GT2009-59062.
[12] Wang DX, Li YS. 3D direct and inverse design using NS equations and the adjoint method for turbine blades. In: Proceedings of the ASME turbo expo 2010: power for land, sea, and air, Glasgow, UK; 2010. p. 537–45. doi:10.1115/GT2010-22049.
[13] Wang DX, He L, Li YS, Wells RG. Adjoint aerodynamic design optimization for blades in multistage turbomachines - part II: validation and application. J Turbomach 2010;132(2):021012. doi:10.1115/1.3103928.
[14] Backhaus J, Engels-Putzka A, Frey C. A code-coupling approach to the implementation of discrete adjoint solvers based on automatic differentiation. In: Proceedings of the VII European congress on computational methods in applied sciences and engineering (ECCOMAS), Crete Island, Greece; 2016. p. 3828–42. doi:10.7712/100016.2076.6428.
[15] Marta AC, Shankaran S. Assessing turbomachinery performance sensitivity to boundary conditions using control theory. J Propul Power 2014;30(5):1–14. doi:10.2514/1.B35087.
[16] Wilcox DC. Multiscale model for turbulent flows. AIAA J 1988;26(11):1311–20. doi:10.2514/3.10042.
[17] Denton JD, Singh U. Time marching methods for turbomachinery flow calculation. In: Appl of numerical methods to flow calculations in turbomachines. Von Karman Inst. for Fluid Dyn.; 1979. p. 47.
[18] Holmes DG. Mixing planes revisited: A steady mixing plane approach designed to combine high levels of conservation and robustness. In: Proceedings of the ASME turbo expo 2008: power for land, sea and air, Berlin, Germany; 2008. p. 2649–58. doi:10.1115/GT2008-51296.
[19] Giles MB. UNSFLO: a numerical method for unsteady inviscid flow in turbomachinery. Tech. Rep. Gas Turbine Laboratory, Massachusetts Institute of Technology; 1988.
[20] Giles MB, Pierce NA. An introduction to the adjoint approach to design. Flow Turbul Combust 2000;65(3–4):393–415. doi:10.1023/A:1011430410075.
[21] Griewank A. Evaluating derivatives: principles and techniques of algorithmic differentation. Philadelphia, PA: SIAM; 2008. ISBN 9780898716597.
[22] Holmes DG, Moore BJ, Connell SD. Unsteady vs. steady turbomachinery flow analysis: Exploiting large-scale computations to deepen our understanding of turbomachinery flows. In: Proceedings of the SciDAC Conference; 2011. Denver, CO.
[23] Hascoët L, Pascual V. The Tapenade automatic differentiation tool: principles, model, and specification. ACM Trans Math Software 2013;39(3). doi:10.1145/2450153.2450158.
[24] Balay S., Abhyankar S., Adams M., Brown J., Brune P., Buschelman K., Eijkhout V., Gropp W., Kaushik D., Knepley M., et al. PETSc users manual revision 3.5. 2014.
[25] Marta AC, Shankaran S. On the handling of turbulence equations in RANS adjoint solvers. Comput Fluids 2013;74:102–13. doi:10.1016/j.compfluid.2013.01.012.
[26] Pini M, Persico G, Pasquale D, Rebay S. Adjoint method for shape optimization in real-gas flow applications. J Eng Gas Turbines Power 2014;137(3):032604. doi:10.1115/1.4028495.
[27] Mueller L, Verstraete T. CAD integrated multipoint adjoint-based optimization of a turbocharger radial turbine. Int J Turbomach Propuls Power 2017;2(3):14. doi:10.3390/ijtpp2030014.
[28] Lynch SP, Thole KA. Comparison of the three-dimensional boundary layer on flat versus contoured turbine endwalls. J Turbomach 2016;138(4):041008. doi:10.1115/1.4032165.
[29] Hairun Xie AW, Wu Y, Ouyang H. Numerical investigation of inlet distortion for different rear mounted engine installations at taking-off conditions. In: Proceedings of the asme turbo expo 2015: power for land, sea and air, Montreal, Canada; V001T01A009; 2015. https://doi.org/10.1115/GT2015-42350.