I Encontro de Jovens Investigadores do LAETA
November 15-16, 2010, Lisbon, Portugal

Blade Shape Optimization using a RANS Discrete Adjoint Solver

Andre C. Marta*, Sriram Shankaran† and Alexander Stein‡

∗ Instituto Superior Técnico, Lisboa, Portugal, andre.marta@ist.utl.pt
† Global Research Center, General Electric, Niskayuna, USA, shankaran@ge.com
‡ Global Research Center, General Electric, Niskayuna, USA, stein@research.ge.com

Abstract
Recent developments in numerical tools for turbomachinery design have made practical the use of gradient-based optimization using high-fidelity computational fluid dynamic (CFD) simulations. Such has been made possible with the use of adjoint solvers, that can efficiently provide the gradients of the functions of interest with respect to the design variables required by the optimizer, at a cost almost independent of the number of variables. The derivation and implementation of the discrete adjoint solver for a legacy Reynolds Average Navier–Stokes (RANS) CFD solver are briefly explained. The adjoint-based gradients of some functions of interest, such as mass flow, pressure ratio and efficiency, with respect to shape parameters are computed and benchmarked against finite-difference approximations and excellent agreement is demonstrated. The outline of the integration of such adjoint tool in an engineering design framework is presented and discussed. The adjoint-based design framework is tested on a shape optimization problem using a set of Hicks-Henne bump functions superimposed on the baseline shape as design variables. A simple design problem is presented: a compressor rotor blade passage is setup as an unconstrained maximization problem, where the efficiency is increased by tweaking the camberline angle distribution.
Keywords: Adjoint, Design, Optimization, Blade, Turbomachinery

1. Introduction

Turbomachinery design has changed significantly over the last ten years with the use of high-fidelity computational fluid dynamic (CFD) simulations. The use of such tools have become common place in industrial environments, and there is now an emerging trend to use numerical optimization techniques as part of the design tools. This allows the designer to explore a vaster design space or fine-tune existing machines much more efficiently.

Among the several optimization methods developed by the operations research field [13], and considering the particular nature of a turbomachinery CFD flow simulation that can take hours, if not days, to perform, the most efficient methods are gradient-based, which require a minimal number of cost function evaluations. However, these methods require an estimate of the cost function derivatives. To address this, the designer faces the problem of evaluating the derivatives [4]. Finite-difference (FD) approximations have always been popular due to their simplicity but they rapidly become computationally prohibitive when the number of variables greatly exceeds the number of functions. In this case, an adjoint method is the best-suited approach to efficiently estimate function gradients since the cost involved in calculating sensitivities using the adjoint method is therefore practically independent of the number of design variables.

The application of adjoint methods to CFD was pioneered by Pironneau [14] and it was later revisited and extended by Jameson to perform airfoil [7] and wing [8] design. More recent successful applications include multipoint aerodynamic shape optimization problems [15], aerostructural design optimization [12], and even magnetohydrodynamics flow control [10].

The major drawback of using adjoint-based gradients has always been the necessity of implementing an additional solver – the adjoint system of equations solver, that is generally of the same complexity as the primary flow solver. To address this, a methodology to develop the adjoint solver based on a hybrid approach [9, 11] is used.

2. Background

The underlying theory of adjoint-based high-fidelity CFD design optimization is presented next.

2.1. Generic Design Problem

When some component of a turbomachine is to be tuned, several characteristics are used to monitor its performance, such as efficiency, pressure ratio or mass flow. In addition, there are several machine-defining parameters that can be adjusted to change those characteristics, such as blade stagger, camber angle and thickness distributions, axial and radial stacking. In the context of optimization, the performance monitoring characteristics are called cost functions, and the adjustable parameters are designated as design variables.

A generic CFD design problem can be formally described as

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

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

When using a gradient-based optimizer to solve the design problem (1), the evaluation of the cost and constraint functions, and their gradients with respect to the design variables are also required, that is, $\frac{\mathrm{d}Y}{\mathrm{d}\alpha}$ and $\frac{\mathrm{d}C_i}{\mathrm{d}\alpha}$ have to be estimated.

2.2. Flow Governing Equations

The governing equations used in the present work are the Reynolds-Averaged Navier–Stokes (RANS) equations. In conservation form, the non-dimensional RANS system of equations may be written in compact vector form as

$$\frac{\partial \mathbf{q}}{\partial t} + (\nabla \cdot \mathcal{F}_i) - (\nabla \cdot \mathcal{F}_v) = 0\,, \tag{2}$$

where

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix}, \quad \mathcal{F}_i = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} + p\mathbf{I} \\ (\rho E + p)\mathbf{u} \end{pmatrix} \text{ and } \quad \mathcal{F}_v = \tfrac{1}{Re} \begin{pmatrix} 0 \\ \vec{\vec{\tau}} \\ \mathbf{u} \cdot \vec{\vec{\tau}} + \frac{\mu}{Pr(\gamma-1)M^2} \nabla T \end{pmatrix},$$

where $\mathbf{q}$ is the vector of conservative variables, and $\mathcal{F}_i$ and $\mathcal{F}_v$ are the inviscid and and viscous flux vectors, respectivelly. A two-equation turbulence model was used, in particular the $k - \omega$ model of Wilcox [17].

In semi-discrete form, the governing equations (2) can be expressed as

$$\frac{\mathrm{d}q_{ijk}}{\mathrm{d}t} + \mathcal{R}_{ijk}(\mathbf{q}) = 0\,, \tag{3}$$

where $\mathcal{R}$ is the residual described earlier with all of its components (inviscid and viscous fluxes, boundary conditions, artificial dissipation, etc.), and the triad ijk represents the three computational directions. The unsteady term of Eq.(3) is dropped out since only the steady solution of the equation is of interest in this work.

2.3. Adjoint Equations

The derivation of the adjoint equations for systems of PDEs follows the work by Giles [3]. The adjoint equations can be expressed as

$$\left[\frac{\partial \mathcal{R}}{\partial \mathbf{q}}\right]^T \psi = \left[\frac{\partial Y}{\partial \mathbf{q}}\right]^T, \tag{4}$$

where ψ is the adjoint vector.

Since the CFD solver does not handle the geometric parameters α directly, but rather a computational mesh defined by the coordinates of each node $\mathbf{x}$, the chain rule of differentiation is used to express the gradient of the cost function with respect to the design variables as

$$\frac{\mathrm{d}Y}{\mathrm{d}\alpha} = \frac{\mathrm{d}Y}{\mathrm{d}\mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\alpha}\,. \tag{5}$$

The total gradient of the cost function with respect to the grid coordinates, based on the adjoint solution ψ, is given by

$$\frac{\mathrm{d}Y}{\mathrm{d}\mathbf{x}} = \frac{\partial Y}{\partial \mathbf{x}} - \psi^T \frac{\partial \mathcal{R}}{\partial \mathbf{x}} \,. \tag{6}$$

The evaluation of the gradient of each cost or constraint function in the optimization problem (1) requires solving Eq.(4) with a new right-hand side vector. On the other hand, the computational cost of the total sensitivity (6) is almost independent of the number of grid coordinates $\mathbf{x}$, which is the feature that makes the adjoint method so attractive for gradient-based optimization involving a large number of variables and a few functions.

The simple mathematical form of Eq.(4) can be very misleading since, depending on the approach, their numerical implementation can be quite complex, if derived by manual differentiation, or quite costly, if derived using finite-differences.

A discrete adjoint approach formulation was chosen because it can be applied to any set of governing equations and it can treat arbitrary cost functions. As such, and in contrast to the continuous approach, no simplifications have to be made during the derivation: the effects of the viscosity and heat transfer and the turbulence equations could be easily by handled when deriving the discrete adjoint.

But the most interesting feature of the discrete approach is that it allows the use of automatic differentiation (AD) tools [2] in its derivation, expediting considerably the process of obtaining the differentiated form of the discretized governing equations necessary to assemble the adjoint system of equations.

2.4. Hybrid approach: AD adjoint

AD tools can automatically generated new code that computes user-specified derivatives as accurately as an analytic method. The AD tool chosen in this work was Tapenade [5] because it supports Fortran 90, which was a requirement taking into account the programming language used in the flow solver implementation.

The approach used in this work is hybrid as follows the work of Marta [11]. The discrete adjoint solver is derived with the aid of an automatic differentiation tool that is selectively applied to the CFD source code that handles the residual and function evaluations. This tool produces the routines that evaluate the partial derivative matrices $\partial\mathcal{R}/\partial\mathbf{q}$, $\partial Y/\partial\mathbf{q}$, $\partial Y/\partial\mathbf{x}$ and $\partial\mathcal{R}/\partial\mathbf{x}$ that are necessary to compute gradients (6) using the adjoint method (4).

This hybrid approach retains the accuracy of the adjoint methods, while it adds the ease of implementation of the automatic differentiation methods.

3. Implementation

The development and verification of the discrete adjoint solver and its integration into a design system are described next.

3.1. Flow Solver

The implementation of the proposed hybrid adjoint solver had a legacy flow solver as its starting point. This flow solver is multi-block, three-dimensional, finite-volume, structured grid, non-linear and linear, Reynolds-Averaged Navier–Stokes solver for turbomachinery blade rows. It is capable of efficiently performing three-dimensional analysis for aeromechanics, aerodynamic design, parametric studies, and robust design applications.

As typical for most iterative CFD flow solvers, the residual calculation is done in a subroutine that loops through the three-dimensional domain and accumulates the several fluxes and boundary conditions contributions in the residual $\mathcal{R}$. However, the residual at each cell only depends on the flow variables at that cell and at the cells adjacent to it, which define the stencil of dependence. This stencil is shown in Fig. 1 for the case of a viscous flow analysis.

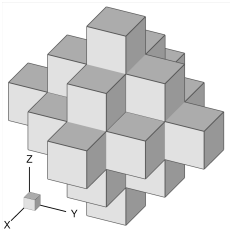


Figure 1: Computational flow stencil: 25 cells.

3.2. Adjoint Solver

The gradients of the functions of interest with respect to the grid coordinates are computed by first assembling the discrete adjoint equations (4), solving them, and then using the sensitivity equation (6). The sizes of the matrices involved in this process are

$$\begin{array}{llll} \dfrac{\partial \mathcal{R}}{\partial \mathbf{q}} & (N_q \times N_q) & , & \dfrac{\partial Y}{\partial \mathbf{q}} \quad (N_Y \times N_q) \ , \\ \dfrac{\partial \mathcal{R}}{\partial \mathbf{x}} & (N_q \times N_x) & , & \dfrac{\partial Y}{\partial \mathbf{x}} \quad (N_Y \times N_x) \ , \end{array} \tag{7}$$

where N_Y is the number of cost functions, N_x the number of grid coordinates and N_q the size of the state vector. The size of the vector $\mathbf{q}$ depends on the number of governing equations, N_e, and the number of cells of the computational mesh, N_c, that discretize the physical domain, according to the relation $N_q = N_e \times N_c$, which for the solution of a large, three-dimensional problem involving a system of conservation laws, can be very large. The size of the grid coordinates vector $\mathbf{x}$, is given by dimensionality of the problem times the number of vertices corresponding to the computational mesh used, that is, $N_x = 3 \times N_v$ for three-dimensional problems. According to the proposed hybrid adjoint approach by Marta [11], automatic differentiation tools are used to generate code that computes the non-zero entries of these matrices of partial sensitivities.

The adjoint linear system of equations (4) has to be solved N_Y times because ψ is valid for all grid coordinates $\mathbf{x}$, but must be recomputed for each function Y. In order to solve this large sparse discrete adjoint problem, the Portable, Extensible Toolkit for Scientific Computation (PETSc) [1] was used. All the adjoint and partial sensitivity matrices and vectors Eq.(7) were created as PETSc's data structures and, due to their structure, stored as sparse entities. Once the sparse data structures were assembled, the adjoint system of equations was solved using a PETSc built-in Krylov subspace (KSP) method, more specifically, a Generalized Minimum Residual (GMRES) method [16] was used.

3.3. Function Gradient Evaluation

Once the adjoint solution, ψ, is found, the gradient of the cost function with respect to the grid coordinates is obtained from Eq.(6). The adjoint-based gradient was evaluated using the matrix-vector multiplication and the vector addition built-in operation routines provided in PETSc.

3.4. Verification of Gradients

The adjoint-based gradients obtained using the hybrid adjoint approach are verified against finite-difference (FD) derivative approximations. To minimize the necessary number of evaluations of the flow solver, the 1^{st}-order forward-difference formula for the first derivative, obtained from the Taylor series expansion of function Y for a perturbation about a point α, is used. That is,

$$\frac{\mathrm{d}Y}{\mathrm{d}\alpha} = \frac{Y(\alpha + \Delta) - Y(\alpha)}{\Delta} + \mathcal{O}(\Delta) \, , \tag{8}$$

where α corresponds to some high-level geometric parameter defining the blade ,and Δ is a perturbation of that parameter. This expression suffers from a large sensitivity to the choice of step size. If h is chosen too large, the derivative estimate might be inaccurate because of the large truncation error; if it is made too small, then subtractive cancellation might occur and the estimate is again inaccurate. Finding the sweet spot of Δ is often problem dependent so several values have to be tried.

3.5. Gradient-Based Optimization Framework

From a design perspective, a turbomachine is geometrically represented not by the surface nodes coordinates but rather by some higher-level descriptors, such as stagger, camber angle distribution and thickness distribution. Let α denote the high-level geometric parameters that form the set of design variables.

While the function values can easily be computed by post processing the flow solution obtained from running the flow solver, the corresponding gradients require an additional solver – the adjoint solver. After the gradient of the objective function with respect to the grid coordinates is computed by Eq.(6), it is still necessary to evaluate the gradient of the computational mesh with respect to those high-level parameters, $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\alpha}$, according to Eq.(5).

Unless the source code of every tool involved in the grid generation process is available, it is necessary to use an approximation to estimate $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\alpha}$. In this work, a simple finite-difference approximation was used to accomplish that. This meant that for every design variable, it was necessary to re-grid the computational

domain. However, since the grid topology was kept constant, it was possible to accelerate this process by means of grid morphing.

The computed final sensitivity $\frac{\mathrm{d}Y}{\mathrm{d}\alpha}$ is then used by the gradient-based optimizer to find the search direction.

The schematic of such adjoint-based optimization algorithm is illustrated in Fig. 2.

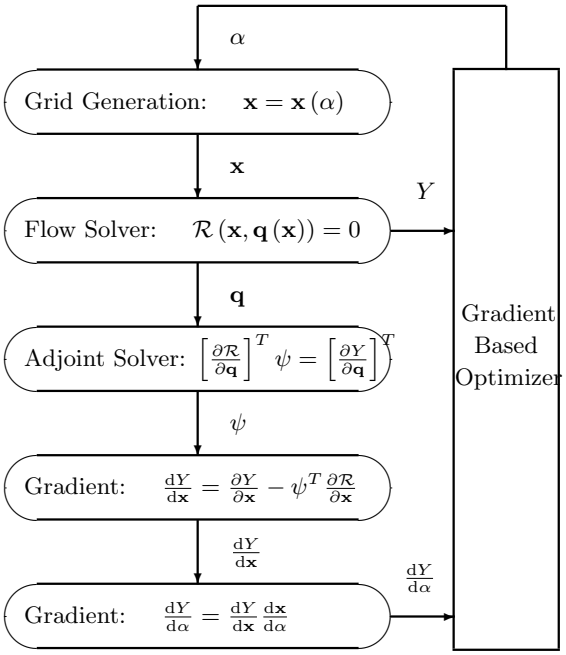


Figure 2: Schematic of the adjoint-based optimization algorithm.

In terms of computational cost, the flow solver and the adjoint solver are the two main blocks of the process, being that the cost of solution of the adjoint equations is similar to that of the solution of the governing equations since they are of similar size and complexity.

4. Results

A transonic blade passage of a high-pressure compressor stage was used to demonstrate the capabilities of the discrete adjoint solver developed using the proposed hybrid approach. The three-dimensional geometry is shown in Fig. 3, where the casing wall has been removed, and the passage has been duplicated for visual clarity.

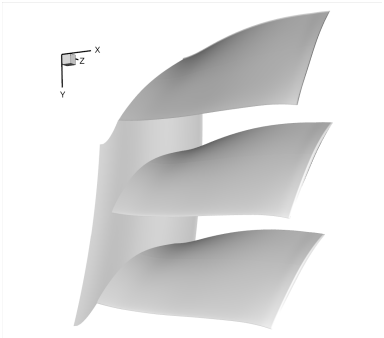


Figure 3: Rotor blade passage.

The inlet boundary conditions were absolute tangential velocity fixed and pressure extrapolated from the interior. The exit static pressure was held fixed. All solid walls were considered impermeable with a no-slip condition. The remaining faces were either periodic, in which the state vector undergone a coordinate transformation according to rotationally periodicity of the geometry.

A H-grid topology was used and a total of four blocks were created, as shown in Figure 4. Several block-to-block interfaces were accounted due to the multi-block mesh topology used.

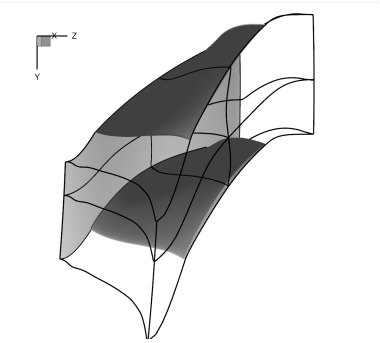


Figure 4: Multi-block computational mesh.

Figure 5 shows the contour of pressure on the hub and blade surface planes corresponding to the baseline blade geometry.

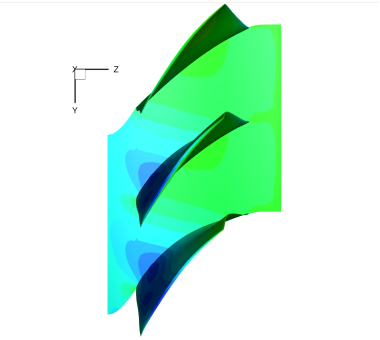


Figure 5: Pressure distribution.

Having obtained the baseline flow solution, the corresponding adjoint solution is computed using Eq.(4). Although an adjoint solution *per se* is of little use to a designer, for completeness, the adjoint solution is shown in Fig. 6 for adiabatic efficiency ($Y = \eta$). The contour plot corresponds to the adjoint of the continuity equation. As typical for the adjoint solution, the plot shows an adjoint flow some how reverse of the real flow.

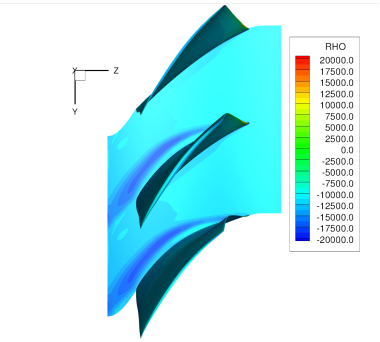
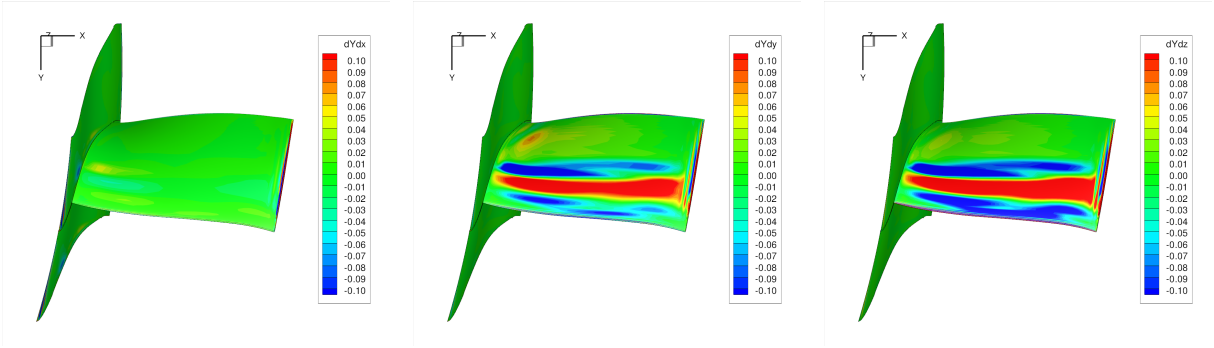


Figure 6: Adjoint solution of the continuity equation ($Y = \eta$).

The adjoint-based gradient of efficiency with respect to the grid coordinates evaluated using Eq.(6) is illustrated in Fig. 7. The plot components correspond to the gradients of pressure ratio with respect to each coordinate component,

$$\frac{\mathrm{d}\eta}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{d}\eta}{\mathrm{d}x}\mathbf{e}_x + \frac{\mathrm{d}\eta}{\mathrm{d}y}\mathbf{e}_y + \frac{\mathrm{d}\eta}{\mathrm{d}z}\mathbf{e}_z \,. \tag{9}$$

Each point indicates the direction of increased efficiency and its magnitude is the improvement per unit change in the grid node coordinate. Consequently, a designer can easily infer from Fig. 7 how to tune the blade and/or endwalls for increased η, since those vectors tell him how the surface geometry should change to accomplish it. The large values at the blade leading and trailing edges reveal how sensitive the machine performance is relative to these regions.



(a) $\frac{\mathrm{d}\eta}{\mathrm{d}x}$ (b) $\frac{\mathrm{d}\eta}{\mathrm{d}y}$ (c) $\frac{\mathrm{d}\eta}{\mathrm{d}z}$

Figure 7: Gradient of efficiency w.r.t. node coordinates.

To verify the adjoint-based gradients, finite-difference derivative approximations were used as benchmark values, using Eq.(8). Hicks-Henne bump functions [6] were used to test the integration with the grid generation module and compute higher-level gradients of the form $\frac{\mathrm{d}Y}{\mathrm{d}\alpha}$. As an example, Figure 8 shows the perturbation produced by a bump located on the blade camberline angle, located at the 50% chord and 50% span of the blade.

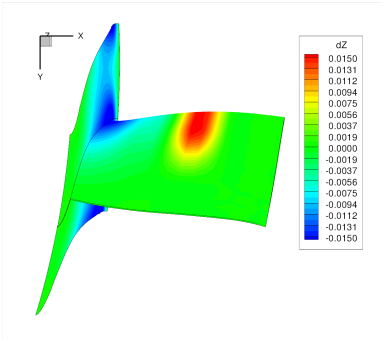


Figure 8: Hicks-Henne bump: applied to the blade camberline angle.

The gradient of different cost functions, namely mass flow, $\dot{m}$, efficiency, η and pressure ratio, PR, with respect to the bump amplitude, a, were computed using the adjoint solver. The term $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\alpha}$ in Eq.(5) was approximated by finite-differences using the baseline and perturbed computational grids. Table 1 summarizes these results, together with the comparison using full finite-difference derivative approximation using a perturbation step of $h = 5 \times 10^{-2}$ on the bump amplitude. This table also includes the different run modes tested in the adjoint solver, namely running single-block/single-processor, multi-block/multi-processor and multi-block/multi-processor modes.

Table 1: Comparison of function gradients.

	Finite-difference (benchmark)	Adjoint Single-block Single-proc	Adjoint Multi-block Single-proc	Adjoint Multi-block Multi-proc
$\frac{\mathrm{d}\dot{m}}{\mathrm{d}a}$	9.852E-2	1.004E-1 (-1.8%)	1.003E-1 (-1.8%)	1.003E-1 (-1.8%)
$\frac{\mathrm{d}PR}{\mathrm{d}a}$	1.298E-3	1.289E-3 (0.7%)	1.289E-3 (0.7%)	1.289E-3 (0.7%)
$\frac{\mathrm{d}\eta}{\mathrm{d}a}$	1.048E-2	1.016E-2 (-3.2%)	1.015E-2 (-3.3%)	1.015E-2 (-3.3%)

As shown in Tab. 1, there is again an excellent agreement between the adjoint-based gradient and the finite-difference derivative approximation. Similar findings were obtained for other bump locations. This also proved that the multi-processor implementation of the adjoint solver was correct. At this point, the adjoint solver implementation was considered successful and ready to be integrated in a gradient-based optimization framework (2).

A sample optimization application, an unconstrained maximization optimization problem using the efficiency as the cost function, was performed. A total of nine design variables were used, corresponding to Hicks-Henne bumps on the camberline angle, whose locations are indicated in Table 2.

Table 2: Design variables: location of bumps on the camberline angle.

Bump #	Chordwise	Spanwise
1	30%	30%
2	50%	30%
3	70%	30%
4	30%	50%
5	50%	50%
6	70%	50%
7	30%	70%
8	50%	70%
9	70%	70%

The relative evolution of the cost function using a gradient-based optimizer based on the steepest descent method is illustrated in Figure 9, where the initial efficiency value was used as reference. As it can be seen, even though the baseline blade corresponded to a tuned geometry, the optimizer was able to further improve its performance. This simplistic test shows that the design framework, equipped with more sophisticated gradient-based optimizer algorithms, is capable of performing highly efficient automated blade shape optimization tasks in industrial turbomachinery design environments.

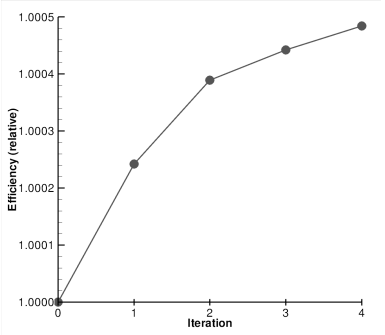


Figure 9: Optimization test: maximization of efficiency.

5. Conclusions

A methodology for developing an adjoint solver for a legacy CFD solver that models the traditional flow governing equations has been revisited and implemented. The discrete adjoint solver was derived with the aid of an automatic differentiation tool that was selectively applied to the CFD source code to produce code that computes the transpose of the flux Jacobian matrix and the other partial derivatives that are necessary to compute gradients using an adjoint method.

The discrete adjoint solver developed was tested on a high-pressure rotor blade passage and the adjoint-based gradients of some cost functions with respect to high-level shape parameters were verified against finite-difference derivative approximations.

Following the successful implementation of the adjoint solver, it was integrated it into a gradient-based optimization framework. The potential of such framework was tested in a blade shape optimization problem were some marginal improvement was still achieved on an already highly tuned blade.

The authors believe that the discussed methodology to develop turbomachinery design frameworks based on adjoint-gradients is an adequate solution to provide the designer with a state-of-the-art tuning tool that, given a set of baseline geometric parameters, re-shapes the blade and endwall surfaces to meet or exceed the design goals, in a very timely manner.

Acknowledgements

The authors would like to thank Daniel Wilkin, 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] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.3.0, Argonne National Laboratory, 2004.

[2] P. Cusdin and J.-D. Müller. On the performance of discrete adjoint CFD codes using automatic differentiation. *International Journal of Numerical Methods in Fluids*, 47(6–7):939–945, 2005.

[3] M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. In *Flow, Turbulence and Combustion*, volume 65, pages 393–415. Kluwer Academic Publishers, 2000.

[4] A. Griewank. *Evaluating Derivatives*. SIAM, Philadelphia, 2000.

[5] L. Hascoët and V. Pascual. Extension of TAPENADE towards Fortran 95. In H. M. Bücker, G. Corliss, P. Hovland, U. Naumann, and B. Norris, editors, *Automatic Differentiation: Applications, Theory, and Tools*, Lecture Notes in Computational Science and Engineering. Springer, 2005.

[6] R. M. Hicks and P. A. Henne. Wing design by numerical optimization. *AIAA Journal*, 15(7):407–412, July 1978.

[7] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, Sept. 1988.

[8] A. Jameson, N. A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. In *Theoretical and Computational Fluid Dynamics*, volume 10, pages 213–237. Springer-Verlag GmbH, Jan. 1998.

[9] A. C. Marta. *Rapid Development of Discrete Adjoint Solvers With Applications to Magnetohydrodynamic Flow Control*. Ph.D. Dissertation, Stanford University, Stanford, CA, USA, June 2007.

[10] A. C. Marta and J. J. Alonso. Toward optimally seeded airflow on hypersonic vehicles using control theory. *Computers & Fluids*, 2010. Accepted for publication.

[11] A. C. Marta, C. A. Mader, J. R. R. A. Martins, E. van der Weide, and J. J. Alonso. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. *International Journal of Computational Fluid Dynamics*, 21(9–10):307–327, Oct. 2007.

[12] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, May 2004.

[13] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 1999.

[14] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64:97–110, 1974.

[15] J. J. Reuther, J. J. Alonso, A. Jameson, M. J. Rimlinger, and D. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *Journal of Aircraft*, 36(1):51–60, 1999.

[16] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal for Scientific and Statistical Computing*, 7(3):856–869, July 1986.

[17] D. C. Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal*, 26(11):1299–1310, Nov 1998.