11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

TOWARDS UNSTEADY ADJOINT ANALYSIS FOR TURBOMACHINERY APPLICATIONS

GEORGIOS D. NTANAKAS*, MARCUS MEYER†

*Rolls-Royce Deutschland
Eschenweg 11, Dahlewitz, 15827 Blankefelde-Mahlow, Germany
PhD Student, National Technical University of Athens (NTUA)
e-mail: georgios.ntanakas@rolls-royce.com

†Rolls-Royce Deutschland
Eschenweg 11, Dahlewitz, 15827 Blankefelde-Mahlow, Germany

Key words: Discrete unsteady adjoint, time-dependent flow, turbomachinery

Abstract. This paper presents a general framework to derive an unsteady discrete adjoint method as part of a gradient-based aerodynamic shape optimization process. The method calculates the gradient of an objective function w.r.t. the design variables of the aerodynamic shape. The formulation of the adjoint equations is introduced within a generic time-dependent optimal design problem. Results are shown that demonstrate the solution of the unsteady adjoint equations for a representative row of nozzle guide vanes of a high pressure turbine.

Nomenclature

R	state equations residual
a	design variables
X	coordinates of the nodes of the volume mesh
X_s	coordinates of the surface mesh on the boundaries
Q	flowfield variables
t	time
R_m	mesh deformation residual
F	objective function
T_p	duration of period
λ	flow adjoint variable
η	mesh adjoint variable

Subscripts/Superscripts

*	time-dependent
$_{aug}$	augmented
^{n}	current time-step

1 INTRODUCTION

Gradient-based optimization is well established in the field of shape optimization in Computational Fluid Dynamics (CFD). To compute gradients of a specific objective functional with respect to the design variables, the adjoint method [1, 2] is preferred instead of forward differentiation methods. This is due to the fact that its computational cost is comparable to that of a flow solution, almost independent of the number of the design variables and only dependent on the number of objective functions, which in CFD applications are significantly less. On the other hand, forward differentiation is dependent on the number of design variables, which in real life problems can be that many that computing gradients in this way is prohibitive. In the following, the pairs primal-dual and direct-adjoint will be used as equivalents to the pair of governing equations for the flow solver and the adjoint flow solver.

Adjoint methods can be subdivided into continuous and discrete variants. In the continuous approach [3, 4, 5], the formulation of the adjoint equations is done prior to the discretization of the governing equations. In the discrete approach [6, 8, 9], the opposite order is followed. The governing equations are first discretized before formulating the adjoint equations. A comparison of the two methods, their advantages and their weaknesses can be found in [10].

Until lately, the time-independent (steady) adjoint method has been the focus of attention for obtaining gradients. But unlike fixed-wing aircraft [11] or automotive applications, there are CFD problems which feature strongly unsteady flows such as helicopter rotors, turbomachinery blades and fluttering wings. Until recently, unsteady solutions were considered to be too expensive for the available computing power. However, the combination of the aforementioned needs with the recent advances in High Performance Computing brought time-dependent (unsteady) adjoint development into focus. The unsteady adjoint method provides a clearer perception of the unsteady phenomena and lays the foundation for new concepts in aerodynamic shape optimization.

In this paper, the unsteady discrete approach is examined. The discrete approach requires a complete linearization of the discrete governing equations with respect to both the flow-field variables and mesh coordinates. Slightly simplifying, to expand to unsteady flows, the evaluation of these linearizations at each physical time-step is needed. A mathematical overview of this procedure is presented and the final form of the equations is discussed. The state equations are solved forward in time and then the adjoint equations backwards in time. Consequently, the infrastructure to store and access the primal solutions is of great importance.

The adjoint solver uses a dual time-stepping scheme (physical and numerical), multigrid acceleration and the Spalart-Allmaras turbulence model with wall functions. The primal flow is imported for every time-step and the corresponding unsteady adjoint flow is calculated. Automatic Differentiation (TAPENADE) was used for selected source code differentiation. The results and the computational cost are presented, analyzed and compared with the equivalent steady results. The dominant frequencies and the variation of adjoint flow variables through time are examined.

2 MATHEMATICAL BACKGROUND

Turbomachinery flows are characterized by temporal (and spatial) periodicity, which means that the unsteadiness can be simulated by looking only at a fragment of time, the period. The governing equations of the flow are considered to be the unsteady Reynolds-averaged Navier-Stokes equations coupled with the Spalart-Allmaras (one-equation) turbulence model. Compared to the steady equations, an additional term expressing the derivative of the flow variables with respect to time is also considered. These equations can be expressed using the following residual form.

$$R^{*}(a,X,Q,t) = \frac{\partial Q\left(t\right)}{\partial t} + R\left(a,X,Q\left(t\right)\right) = 0\ . \tag{1}$$

The mesh deformation residual is also introduced which describes the deformation of the volume mesh as a function of the surface mesh.

$$R_m(X_s,X) = 0\ . \tag{2}$$

Usually, in CFD calculations only a few specific objective functions are of interest. For turbomachinery applications, these include for example efficiency, pressure loss, reduced massflow and reaction. We define the objective function in an unsteady calculation using the following formula:

$$F^{*}(a,X,Q,t) = \frac{1}{T_p}\int_0^{T_p} F\left(a,X,Q\left(t\right)\right)dt\ . \tag{3}$$

Equation (3) can be augmented to form the so-called Lagrangian with the addition of two zero terms which include the adjoint variables.

$$\begin{aligned} F^{*}_{aug}(a,X_s,X,Q,\lambda,\eta,t) = {} & \tfrac{1}{T_p}\int\limits_0^{T_p} F(a,X,Q,t) + \tfrac{1}{T_p}\int\limits_0^{T_p} \lambda^T R(a,X,Q,t) \\ + {} & \eta^T R_m(X_s,X)\ . \end{aligned} \tag{4}$$

Applying the chain rule to differentiate equation (4) w.r.t. the design variables leads to:

$$\begin{aligned}\frac{dF^*_{aug}}{da} = & \tfrac{1}{T_p}\int_0^{T_p}\left(\tfrac{\partial F^*}{\partial a} + \left[\tfrac{\partial X}{\partial a}\right]^T\tfrac{\partial F^*}{\partial X} + \left[\tfrac{\partial Q}{\partial a}\right]^T\tfrac{\partial F^*}{\partial Q}\right)dt \\ + & \tfrac{1}{T_p}\int_0^{T_p}\left[\lambda^T\left(\tfrac{\partial R^*}{\partial a} + \left[\tfrac{\partial X}{\partial a}\right]^T\tfrac{\partial R^*}{\partial X} + \left[\tfrac{\partial Q}{\partial a}\right]^T\tfrac{\partial R^*}{\partial Q}\right)\right]dt \\ + & \eta^T\left(\left[\tfrac{\partial X_s}{\partial a}\right]^T\tfrac{\partial R_m}{\partial X_s} + \left[\tfrac{\partial X}{\partial a}\right]^T\tfrac{\partial R_m}{\partial X}\right) \ . \end{aligned} \tag{5}$$

Rearranging a few terms, we obtain:

$$\begin{aligned}\frac{dF^*_{aug}}{da} = & \tfrac{1}{T_p}\int_0^{T_p}\tfrac{\partial F^*}{\partial a}dt + \eta^T\left[\tfrac{\partial X_s}{\partial a}\right]^T\tfrac{\partial R_m}{\partial X_s} + \tfrac{1}{T_p}\int_0^{T_p}\lambda^T\tfrac{\partial R^*}{\partial a}dt \\ + & \tfrac{1}{T_p}\int_0^{T_p}\left[\tfrac{\partial Q}{\partial a}\right]^T\left(\tfrac{\partial F^*}{\partial Q} + \lambda^T\tfrac{\partial R^*}{\partial Q}\right)dt \\ + & \left[\tfrac{\partial X}{\partial a}\right]^T\left[\tfrac{1}{T_p}\int_0^{T_p}\left(\tfrac{\partial F^*}{\partial X} + \lambda^T\tfrac{\partial R^*}{\partial X}\right)dt + \eta^T\tfrac{\partial R_m}{\partial X}\right] \ . \end{aligned} \tag{6}$$

Thus, we obtain the flow adjoint and the mesh adjoint equations (7) and (8), by trying to eliminate the last two terms of equation (6):

$$\frac{\partial F^*}{\partial Q} + \lambda^T\frac{\partial R^*}{\partial Q} = 0 \Leftrightarrow \left[\frac{\partial R^*}{\partial Q}\right]^T\lambda = -\left[\frac{\partial F^*}{\partial Q}\right]^T \ , \tag{7}$$

$$\frac{1}{T_p}\int_0^{T_p}\left(\frac{\partial F^*}{\partial X} + \lambda^T\frac{\partial R^*}{\partial X}\right)dt + \eta^T\frac{\partial R_m}{\partial X} = 0$$

$$\Leftrightarrow \left[\frac{\partial R_m}{\partial X}\right]^T\eta = -\frac{1}{T_p}\int_0^{T_p}\left(\frac{\partial F^*}{\partial X} + \lambda^T\frac{\partial R^*}{\partial X}\right)dt \ . \tag{8}$$

The final equation for the computation of the derivative of the objective function w.r.t. the design variables in an unsteady periodic problem can now be written as:

$$\frac{dF^*}{da} = \frac{dF^*_{aug}}{da} = \frac{1}{T_p}\int_0^{T_p}\frac{\partial F^*}{\partial a}dt + \eta^T\left[\frac{\partial X_s}{\partial a}\right]^T\frac{\partial R_m}{\partial X_s} + \frac{1}{T_p}\int_0^{T_p}\lambda^T\frac{\partial R^*}{\partial a}dt \ . \tag{9}$$

For the purpose of this paper, we will focus only on obtaining the solution of the flow adjoint equation (7).

2.1 Time Discretization

In order to numerically solve the equations, the duration of a period T_p is discretized in N time-steps. Using a second order accurate scheme, the time-derivative of the flow variables can be discretized according to:

$$\frac{\partial Q}{\partial t} = \tfrac{3Q^n - 4Q^{n-1} + Q^{n-2}}{2\Delta t}, \qquad \forall n \in [3, N] \ , \tag{10}$$

$$\frac{\partial Q}{\partial t} = \tfrac{Q^n - Q^{n-1}}{\Delta t}, \qquad n = 2 \ . \tag{11}$$

The flow residual can now be written in a discrete form as:

$$R^n(X, Q^{n-2}, Q^{n-1}, Q^n) = \frac{3Q^n - 4Q^{n-1} + Q^{n-2}}{2\Delta t} + R\left(X, Q^n\right) = 0 \ . \tag{12}$$

The discretized form of the objective function is given by:

$$F^* = \frac{1}{T_p} \sum_{n=1}^{N} F^n(X, Q^n) \ . \tag{13}$$

Rewriting equation (7) using the time discretization leads to:

$$\begin{bmatrix} \frac{\partial R^1}{\partial Q^1} & \frac{\partial R^2}{\partial Q^1} & \frac{\partial R^3}{\partial Q^1} & \cdots & \frac{\partial R^N}{\partial Q^1} \\ \frac{\partial R^1}{\partial Q^2} & \frac{\partial R^2}{\partial Q^2} & \frac{\partial R^3}{\partial Q^2} & \cdots & \frac{\partial R^N}{\partial Q^2} \\ \vdots & & \ddots & \ddots & \vdots \\ \frac{\partial R^1}{\partial Q^N} & \frac{\partial R^2}{\partial Q^N} & & & \frac{\partial R^N}{\partial Q^N} \end{bmatrix} \begin{bmatrix} \lambda^1 \\ \lambda^2 \\ \vdots \\ \lambda^N \end{bmatrix} = \begin{bmatrix} \frac{\partial F^1}{\partial Q^1} & \frac{\partial F^2}{\partial Q^1} & \frac{\partial F^3}{\partial Q^1} & \cdots & \frac{\partial F^N}{\partial Q^1} \\ \frac{\partial F^1}{\partial Q^2} & \frac{\partial F^2}{\partial Q^2} & \frac{\partial F^3}{\partial Q^2} & \cdots & \frac{\partial F^N}{\partial Q^2} \\ \vdots & & \ddots & \ddots & \vdots \\ \frac{\partial F^1}{\partial Q^N} & \frac{\partial F^2}{\partial Q^N} & & & \frac{\partial F^N}{\partial Q^N} \end{bmatrix} \mathbf{I} \tag{14}$$

From the discretization scheme and equation (12) the following can be extracted:

$$\frac{\partial R^k}{\partial Q^l} = 0, \qquad \forall (k < l) \cup (k > l + 2) \ . \tag{15}$$

In addition, from the definition of the objective function it follows that:

$$\frac{\partial F^k}{\partial Q^l} = 0, \qquad \forall k \neq l \ . \tag{16}$$

Now, equation (14) can be rewritten as:

$$\begin{bmatrix} \frac{\partial R^1}{\partial Q^1} & \frac{\partial R^2}{\partial Q^1} & \frac{\partial R^3}{\partial Q^1} & 0 & \cdots & & 0 \\ 0 & \frac{\partial R^2}{\partial Q^2} & \frac{\partial R^3}{\partial Q^2} & \frac{\partial R^4}{\partial Q^2} & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ & & & & & & \frac{\partial R^N}{\partial Q^{N-2}} \\ 0 & & & & & & \frac{\partial R^N}{\partial Q^{N-1}} \\ 0 & 0 & \cdots & & & 0 & \frac{\partial R^N}{\partial Q^N} \end{bmatrix} \begin{bmatrix} \lambda^1 \\ \lambda^2 \\ \vdots \\ \\ \vdots \\ \lambda^N \end{bmatrix} = \begin{bmatrix} \frac{\partial F^1}{\partial Q^1} & 0 & \cdots & & 0 \\ 0 & \frac{\partial F^2}{\partial Q^2} & & & \\ \vdots & & \ddots & & \vdots \\ & & & & \\ & & & & 0 \\ 0 & & \cdots & 0 & \frac{\partial F^N}{\partial Q^N} \end{bmatrix} \mathbf{I} = \begin{bmatrix} \frac{\partial F^1}{\partial Q^1} \\ \frac{\partial F^2}{\partial Q^2} \\ \vdots \\ \\ \\ \frac{\partial F^N}{\partial Q^N} \end{bmatrix} \tag{17}$$

Based on equation (17) we end up with the following formula mimicing a backward substitution technique:

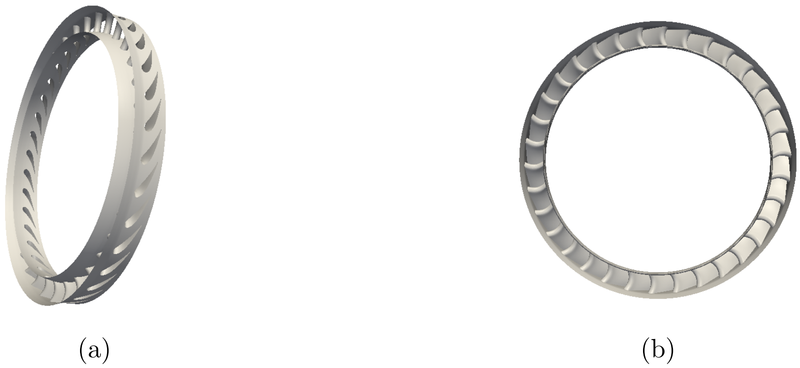
$$\begin{aligned} (\lambda^N)^T \frac{\partial R^N}{\partial Q^N} &= \frac{\partial F^N}{\partial Q^N} \ , \\ (\lambda^n)^T \frac{\partial R^n}{\partial Q^n} + (\lambda^{n+1})^T \frac{\partial R^{n+1}}{\partial Q^n} &= \frac{\partial F^n}{\partial Q^n}, \qquad n = N - 1 \ , \\ (\lambda^n)^T \frac{\partial R^n}{\partial Q^n} + (\lambda^{n+1})^T \frac{\partial R^{n+1}}{\partial Q^n} + (\lambda^{n+2})^T \frac{\partial R^{n+2}}{\partial Q^n} &= \frac{\partial F^n}{\partial Q^n}, \qquad \forall n \in [1, N - 2] \ . \end{aligned} \tag{18}$$

From equation (18) it can be seen that the unsteady adjoint equation needs to be solved backwards in time. It is also clear that for the adjoint variable calculation in every time-step the primal solution is needed. On the other hand, the primal solution is obtained by forward time integration. At present, the following process is employed: The primal is solved for the full period forward in time and the solutions for every time-step are stored on the hard disk. Then the adjoint is solved backwards in time reading in the corresponding primal solution in every time-step.

This approach can lead to significant file I/O and requires sufficient disk space. Thus, techniques to reduce the needed storage space and file I/O will be investigated in the future.

3 APPLICATION

For illustration purposes, a representative row of nozzle guide vanes (NGV) of a high pressure turbine (HPT) was selected (see Figure 1). Because of spatial periodicity, only one blade will be used for the calculations. The mixed-element, unstructured mesh consists of approximately 1 million grid nodes, as shown in Figure (2).



(a)

(b)

Figure 1: Row of nozzle guide vanes

(a) Radial cut at mid-height of volume mesh

(b) Surface mesh

Figure 2: Mesh and CFD domain

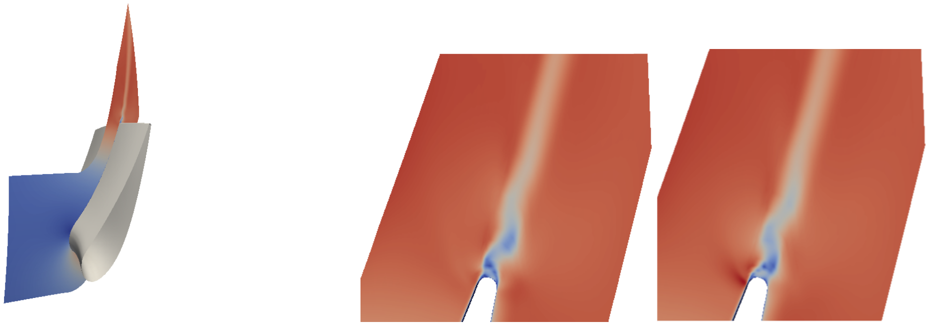
3.1 Unsteady Non-Linear Flow Solution

The following boundary conditions are employed:

- viscous wall treatment for the airfoil, the hub and the casing,
- prescribed total pressure and total temperature at the inflow,
- prescribed static pressure at the outflow and
- periodicity treatment for the upper and lower boundaries

The simulation is run for a time period of 128 time-steps, with $T_p = 2 \cdot 10^{-4}$ *sec* (i.e. $\Delta t \simeq 1,56 \cdot 10^{-5}$ *sec*). The selection of the time-step was based on theoretical and empirical equations concerning the appearance of turbulent vortex shedding. The primal flow is saved for every time-step for the subsequent use in the adjoint solver.

After the solver's convergence, a vortex shedding appears downstream of the airfoil's trailing edge. The contours of the relative velocity magnitude and a close up of the trailing edge is shown in Figure (3).

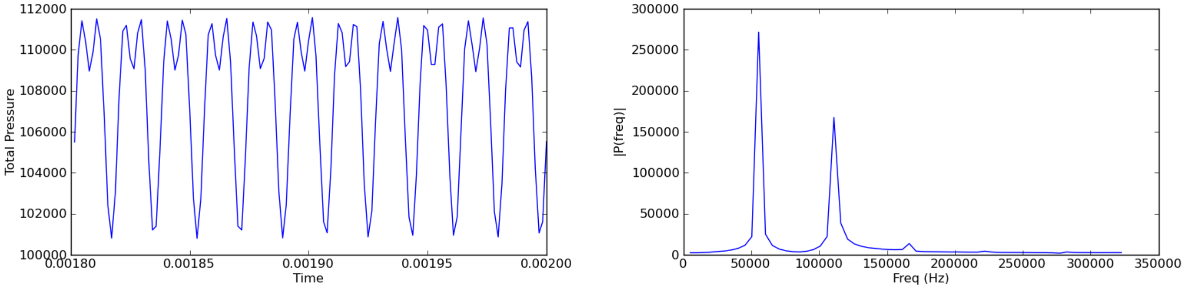


(a) Entire calculation domain

(b) Trailing edge close up at two different time-steps

Figure 3: Radial cut at mid - height of relative velocity magnitude

In order to further confirm the presence of a time-dependent flow, a grid point near the trailing edge is picked and the total pressure is plotted over time, as shown in Figure (4). Moreover, a Fast Fourier Transform analysis is performed to obtain the dominant frequencies of the unsteadiness.



(a) Total pressure over 1 calculation period (T_p)

(b) Dominant frequencies

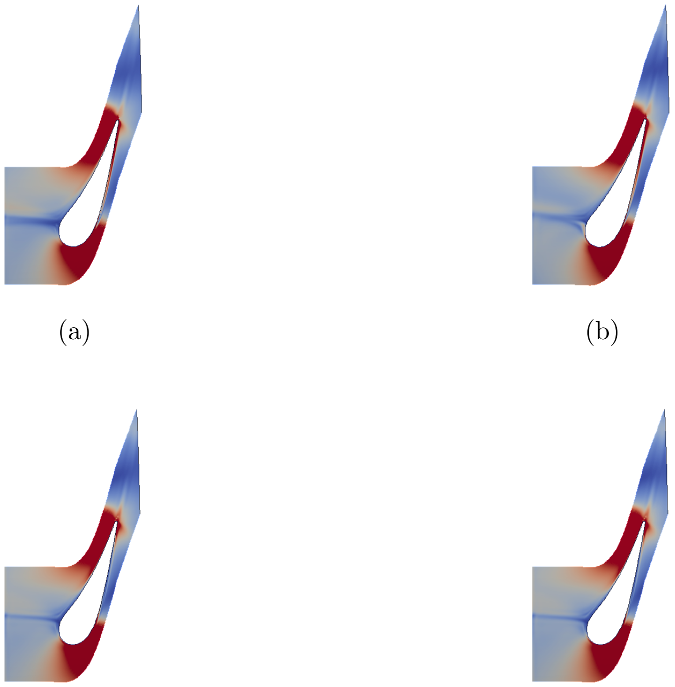
Figure 4: Total pressure of grid point near the trailing edge

3.2 Unsteady Adjoint Flow Solution

The backbone of the unsteady adjoint flow solver is eq. 18. The solver reads in the primal flow for every time-step in order to calculate the adjoint flow going backwards in time. The number and duration of timesteps are identical to these of the primal problem. The calculations are initialized from a converged steady adjoint flow solution. This permits to run the solver for less iterations in every timestep and reduce computing time. The selected objective function is mass flow at the inlet.

In Figure (5), the adjoint flow of four different time-steps is shown. The differences are visible especially near the leading edge.

Below, the unsteady adjoint mean value results are compared with the steady adjoint results. The unsteady adjoint mean value relative velocity magnitude is plotted in comparison with the corresponding steady result in Figure (6). The two contour plots are very similar, but the influence of the unsteady vortex shedding can be seen on the unsteady result, when compared to the steady result, where this physical phenomenon is not captured.



(a)

(b)

(c)

(d)

Figure 5: Unsteady adjoint relative velocity magnitude on a radial cut at mid-height for different time-steps

(a) Steady

(b) Unsteady-mean

Figure 6: Comparison of steady and unsteady mean adjoint relative velocity magnitude on radial cut at mid-height

4 CONCLUSIONS

Within this work an unsteady discrete adjoint solver was developed and applied to the flow of a nozzle guide vane of a high pressure turbine. The results were compared to those of the steady adjoint solver. The unsteady adjoint appoach sheds light on more details of the unsteady flow and can yield additional information to improve current aerodynamic designs.

Next steps will focus on the implementation of the adjoint version of a sliding plane boundary condition, which is similar to overset grids [9] and couples the stationary and rotating components in turbomachinery applications. In addition, methods to reduce file I/O overhead and disk space reduction will be investigated to improve the computational efficiency of the unsteady adjoint method.

REFERENCES

[1] Giles, M. Pierce, N. An introduction to the adjoint approach to design. *Flow Turbulence and Combustion.* 65 (3) (2000) 393415.

[2] Jameson, A. Aerodynamic shape optimization using the adjoint method. VKI Lecture Series on Aerodynamic Drag Prediction and Reduction. von Karman Institute of Fluid Dynamics, Rhode St Genese, Belgium.

[3] Jameson, A. Sriram, S. and Martinelli, L. A Continuous Adjoint Method for Unstructured Grids. AIAA Paper 2003- 3955, 16th AIAA CFD Conference, Orlando, Florida, 2003.

[4] Giannakoglou, K.C. Papadimitriou, D.I. Papoutsis-Kiachagias, E.M. The Continuous Adjoint Method: Theory and Industrial Applications. von Karman Institute Lectures Series on Introduction to Optimization and Multidisciplinary Design in Aeronautics and Turbomachinery. May 7-11, 2012.

[5] Asouti, V.G. Zymaris, A.S. Papadimitriou, D.I. Giannakoglou, K.C. Continuous and Discrete Adjoint Approaches for Aerodynamic Shape Optimization with Low Number preconditioning. *International Journal for Numerical Methods in Fluids.* Vol. 57, pp. 1485-1504, 2008.

[6] Giles, M.B. and Pierce, N.A. Adjoint equations in CFD: duality, boundary conditions and solution behavior. AIAA 97-1850, 1997.

[7] Giles, M.B. Adjoint methods for aeronautical design. ECCOMAS CFD conference, 2001.

[8] Mavriplis, D. Discrete adjoint-based approach for optimization problems on three-dimensional unstructured meshes. AIAA journal 45.4 (2007): 741-750.

[9] Nielsen, E.J. Diskin B. Discrete Adjoint-Based Design for Unsteady Turbulent Flows on Dynamic Overset Unstructured Grids. *AIAA Journal*. Vol. 51, No. 6 (2013), pp. 1355-1373.

[10] Nadarajah, S. and Jameson, A. A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization. *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*. Reno NV, 2000, AIAA Paper 20000667.

[11] Nadarajah, S. Kim, S. Jameson, A. Alonso, J. Sonic boom reduction using an adjoint method for supersonic transport aircraft configuration. Symposium transsonicum IV, International Union of Theoretical and Applied Mechanics, September 26, 2002, DLR Gottingen, Germany.

[12] Nadarajah, S. Jameson, A. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic optimization, AIAA p. 2001-2530. AIAA 15th computational fluid dynamics conference, Anaheim, CA, June 2001.

[13] Campobasso, M.S. Duta, M. Giles M.B. Adjoint methods for turbomachinery design. ISABE conference, Bangalore, 2001.

[14] Campobasso, M.S. Duta, M. Giles M.B. Adjoint calculation of sensitivities of turbomachinery objective functions AIAA J Propulsion Power, 19 (4), 2003.

[15] Denton, J.D. and Singh, U.K. Time-marching methods for turbomachinery flow calculations. VKI-LEC-SER-1979-9, von Karman Inst. for Fluid Dynamics, Belgium, 1979.

[16] Mohammadi, B. Pironneau, O. Shape optimization in fluid mechanics. Annu Rev Fluid Mech, 36, pp. 225279, 2004.

[17] Spalart, P.R. Allmaras, S.R. A one-equation turbulence model for aerodynamic flows. La Recherche Aerospatiale, pp. 521, 1994.

[18] Hascoët, L. Tapenade: a tool for automatic differentiation of programs. In Proceedings of 4th European Congress on Computational Methods, ECCOMAS 2004, Jyvaskyla, Finland, 2004. 13, 32

Acknowledgement

This work was supported by the project *AboutFlow* (`http://aboutflow.sems.qmul.ac.uk`), funded by the European Commission under FP7-PEOPLE-2012- ITN-317006.