Evolutionary and Deterministic Methods for Design,
Optimization and Control with Applications to Industrial and Societal Problems
EUROGEN 2005
R. Schilling, W.Haase, J. Periaux, H. Baier, G. Bugeda (Eds)

DISCRETE AND CONTINUOUS ADJOINT FORMULATIONS FOR TURBOMACHINERY APPLICATIONS

D.I. Papadimitriou*, A.S. Zymaris* and K.C. Giannakoglou†
Lab. of Thermal Turbomachines,
National Technical University of Athens,
P.O. Box 64069, 15710 Athens, GREECE

* PhD Student, National Technical University of Athens, Greece, dpapadim@mail.ntua.gr

* Graduate Student, NTUA, Greece

† Associate Professor, NTUA, Greece, kgianna@central.ntua.gr

Key words: Discrete and Continuous Adjoint Method, Navier Stokes Equations, Sensitivity Derivatives, Structured and Unstructured Grids.

Abstract. *In this paper, the adjoint method, in both continuous and discrete formulation, is used to solve cascade airfoil inverse design problems, the target being a desired pressure distribution along its boundaries, or optimization problems in which cascade airfoils with minimum viscous losses are designed. For the purpose of demonstration, a turbine and a compressor cascade, are analyzed. In the continuous adjoint method, the objective function gradient is computed using a field independent formulation which avoids the calculation of metrics variations over the computational domain. On the other hand, the discrete adjoint method is programmed by considering computer memory requirements and CPU time. An extra term is added to the augmented function in order to handle the geometrical constraints of mimimum airfoil blade thickness. So, both structured and unstructured grids are used; the Spalart–Allmaras one–equation model is employed for the turbulent flow cases. A comparison of the first– and second–order discrete adjoint technique is also presented.*

1 Introduction

During the last years, significant boast has been given to the development of computational tools capable of solving complex aerodynamic optimization problems. Evolutionary [1] and deterministic optimization methods are in use. In deterministic optimization methods, it is important to compute objective function gradients with accuracy and low CPU cost. The so–computed gradient can also be used as an extra information to train enhanced artificial neural networks, [2], for use in evolutionary optimization algorithms. The most well known gradient computing technique is the adjoint technique, which is based mainly on control and Lagrange multipliers theory. In the literature, two approaches exist, namely the continuous and the discrete adjoint technique. In the continuous approach, the adjoint pde's are first formed and then discretised and solved whereas, in the discrete approach, the discrete adjoint equations are derived directly from the discretised flow equations.

The continuous adjoint method was firstly introduced for potential flows by Pironneau, [3], and Jameson, [4], extended the method to face transonic aerodynamic problems. The method was extended by the same author to handle viscous flows and overall aircraft configurations [5],[6],[7]. Giles was the first to deal with the strict mathematical/theoretical aspect of the discrete adjoint formulation, [8]. Then, numerous scientists dealt with the basic adjoint technique evolution. Anderson and Venkatakrishan [9] used both approaches in inviscid and viscous flows using unstructured grids and, later, Nielsen and Anderson applied the discrete approach on 3D viscous flows [10]. Elliot and Peraire [11] developed discrete adjoint codes for Euler equations using 3D unstructured grids. Kim [12] used a discrete adjoint technique for the optimization of wing body configurations in three dimensional inviscid flows. Giles and Duta [13], [14] used the discrete approach in 2D and 3D viscous and inviscid flows, including turbomachinery flows, steady and unsteady applications [15].

This paper presents recent activities carried out in the Lab. of Thermal Turbomachines of NTUA concerning the development and use of adjoint methods for the purpose of designing or optimizing turbomachinery blades. Both discrete and continuous adjoint methods, with different objective functions will be presented and used for the design–optimization of a compressor and a turbine airfoil cascade. Some cases are analyzed using structured and some other using unstructured grids, thanks to a unique formulation which eliminates the effect of internal grid variation terms and computes derivatives using exclusively boundary terms. The presentation covers inviscid, laminar and turbulent flows without, however, considering the variation of turbulent viscosity computed by the Spalart–Allmaras model. It is interesting to note that, in optimization problems, where viscous losses should be minimized, mathematically different objective functionals are used; the continuous adjoint method aims at minimizing the entropy generation over the flow domain whereas the discrete one aims at minimizing total pressure losses between the inlet and outlet of the flow domain.

2 Objective Functions

The objective functions associated with any shape optimization problem can generally be written as

$$F = \int_{\Omega} F_{\Omega}\, d\Omega + \int_{\partial\Omega} F_{\partial\Omega}\, d\left(\partial\Omega\right) \tag{1}$$

which should be minimized. In eq. (1), Ω and $\partial\Omega$ denote the flow domain and its boundary, respectively; the shape of the latter is partially or totally controlled by the design variables $\overrightarrow{b}$. The objective function (1) is augmented by adding the flow equations multiplied by the vector of $\overrightarrow{\Psi}$ costate variables. We, thus, come up with the so called augmented objective function, whose variation is

$$\begin{aligned} \delta F_{aug} = & \int_{\Omega} \delta F_{\Omega} d\Omega + \int_{\Omega} F_{\Omega} \delta\left(d\Omega\right) + \int_{\partial\Omega} \delta F_{\partial\Omega} d\left(\partial\Omega\right) \\ & + \int_{\partial\Omega} F_{\partial\Omega}\, \delta\left(d\left(\partial\Omega\right)\right) + \int_{\Omega} \overrightarrow{\Psi}^{T} \delta \left(\frac{\partial \overrightarrow{f}_{i}}{\partial x_i} - \frac{\partial \overrightarrow{f}_{v,i}}{\partial x_i} \right) d\Omega \end{aligned} \tag{2}$$

In eq. (2), the last term stands for the steady flow equations for compressible fluids, where $\overrightarrow{f}_{i}$ and $\overrightarrow{f}_{v,i}$ are the inviscid and the viscous fluxes, respectively. The repeated index denotes summation.

Inverse design problems, dealing with the design of airfoils or ducts which reproduce a target pressure distribution $p_{tar}(s)$ over their solid walls S_w, make use of

$$F = \frac{1}{2} \int_{s_w} \left(p - p_{tar}\right)^2 dS \tag{3}$$

whereas the search of airfoils with minimum viscous losses is based on, [16],

$$F = \int_{\Omega} \frac{1}{T} \tau_{ij} \frac{\partial u_i}{\partial x_j} d\Omega \tag{4}$$

in which a term which is equivalent to the entropy generation is expressed as a volume integral. T is the temperature, τ_{ij} the viscous stress tensor components

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \delta_{ij} \frac{\partial u_k}{\partial x_k} \quad , \lambda = -\frac{2}{3}\mu \tag{5}$$

and u_i the Cartesian velocity components. As it will become clear below, expression (4) is useful only in the continuous adjoint approach. For the same problem, the discrete adjoint formulation is alternatively based on

$$F = \frac{\int_{S_i} p_t dS}{S_i} - \frac{\int_{S_o} p_t dS}{S_o} \tag{6}$$

which involves the total pressure integrals over the input (S_i) and output (S_o) boundaries of Ω.

3 The Discrete Adjoint Formulation

The gradient of the objective function F and the discretized flow equations denoted by $\overrightarrow{R} = 0$ with respect to the design variables $\overrightarrow{b}$ read

$$\begin{aligned} \frac{d\,F}{d\,\overrightarrow{b}} &= \frac{\partial\,F}{\partial\,\overrightarrow{U}}\frac{\partial\,\overrightarrow{U}}{\partial\,\overrightarrow{b}} + \frac{\partial\,F}{\partial\,\overrightarrow{b}} \\ \frac{d\,\overrightarrow{R}}{d\,\overrightarrow{b}} &= \frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\frac{\partial\,\overrightarrow{U}}{\partial\,\overrightarrow{b}} + \frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{b}} = 0 \end{aligned} \tag{7}$$

The derivative of the augmented function with respect to the design variables yields

$$\frac{d\,F_{aug}}{d\,\overrightarrow{b}} = \frac{d\,F}{d\,\overrightarrow{b}} + \overrightarrow{\Psi}^T\frac{d\,\overrightarrow{R}}{d\,\overrightarrow{b}} = \left(\frac{\partial\,F}{\partial\,\overrightarrow{U}} + \overrightarrow{\Psi}^T\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\right)\frac{\partial\,\overrightarrow{U}}{\partial\,\overrightarrow{b}} + \frac{\partial\,F}{\partial\,\overrightarrow{b}} + \overrightarrow{\Psi}^T\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{b}} \tag{8}$$

The first term on the r.h.s. is eliminated by formulating and solving the adjoint equations

$$\left[\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\right]^T\overrightarrow{\Psi} = -\left[\frac{\partial\,F}{\partial\,\overrightarrow{U}}\right]^T \tag{9}$$

so, the final expression for the gradient of F becomes

$$\frac{d\,F_{aug}}{d\,\overrightarrow{b}} = \frac{\partial\,F}{\partial\,\overrightarrow{b}} + \overrightarrow{\Psi}^T\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{b}} \tag{10}$$

which can be used to support any descent algorithm. For the solution of the flow equations, on either structured or unstructured (with triangular elements) grids, a time–marching Navier–Stokes equations solver based on the finite–volume technique is used. The discretization of the flow equations is carried out through the Roe's approximate Riemann solver, where second–order accuracy is obtained by means of variables' extrapolation. From the computational point of view, the Jacobian $\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}$ of the flow equations appears in both the flow and adjoint equations, on their l.h.s. and r.h.s. For the numerical solution, the point–implicit Jacobi method is used, where only the first–order $\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}$ terms are used in the l.h.s. of the delta formulation of the direct equations

$$\left[\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\right]_{0(\Delta x)}\Delta\overrightarrow{U}^n = -\overrightarrow{R}^n \quad , \quad \overrightarrow{U}^{n+1} = \overrightarrow{U}^n + \Delta\overrightarrow{U}^n \tag{11}$$

The corresponding scheme applied to the adjoint equations yields

$$\left[\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\right]^T_{0(\Delta x)}\Delta\overrightarrow{\Psi}^n = -\left[\frac{\partial\,F}{\partial\,\overrightarrow{U}}\right] - \left[\frac{\partial\,\overrightarrow{R}}{\partial\,\overrightarrow{U}}\right]^T_{0(\Delta x^2)}\overrightarrow{\Psi}^n \quad , \quad \overrightarrow{\Psi}^{n+1} = \overrightarrow{\Psi}^n + \Delta\overrightarrow{\Psi}^n \tag{12}$$

Working with unstructured grid, the storage of $\left[\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}\right]^T_{0(\Delta x^2)}$ is cumbersome, since it corresponds to an extended computational stencil compared to that formed by any node and its adjacent ones. To avoid excess storage requirements, only the coefficients of $\left[\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}\right]^T_{0(\Delta x)}$ are stored and the supplementary second–order terms are recomputed at each iteration of the adjoint equations solver.

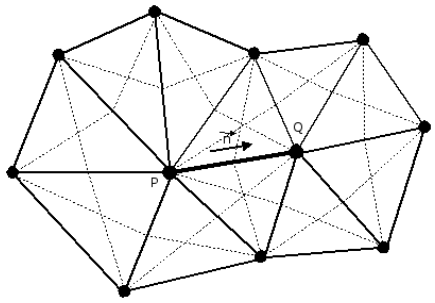


Figure 1: The flux crossing the finite–volume boundary part associated with edge PQ depends on the flow variables at node P, its neighboring nodes (including Q) as well as the neighbors of Q.

4 The Continuous Adjoint Formulation–Inviscid Terms

This section gives the key points of the continuous adjoint method used in inverse design problems in which the target is a given pressure distribution and the objective function is that of eq. (3). From the viewpoint of body–fitted structured grids and the associated coordinate transformation $(x_1, x_2) \longleftrightarrow (\xi^1, \xi^2)$, the variation of transformation metrics is

$$\delta\left(\frac{\partial \xi^j}{\partial x_i}\right) = -\frac{\partial \xi^j}{\partial x_k}\frac{\partial\left(\delta x_k\right)}{\partial x_i} \tag{13}$$

which can be used to express the variation of the divergence of the inviscid fluxes, [17]

$$\delta\left(\frac{\partial \overrightarrow{f}_i}{\partial x_i}\right) = \frac{\partial\left(\delta \overrightarrow{f}_i\right)}{\partial x_i} + \frac{\partial \overrightarrow{f}_i}{\partial \xi_j}\delta\left(\frac{\partial \xi_j}{\partial x_i}\right) = \frac{\partial\left(\delta \overrightarrow{f}_i\right)}{\partial x_i} - \frac{\partial \overrightarrow{f}_i}{\partial x_k}\frac{\partial\left(\delta x_k\right)}{\partial x_i} \tag{14}$$

Since, in the last expression, metrics are not present, this can be generalized to cover problems handled using either structured or unstructured grids. Thus, eq. (14) can be used in the development of the continuous adjoint equations regardless of the grid type.

For the Euler equations ($\overrightarrow{f}_{v,i} = 0$), the last term in eq. (2) yields

$$\int_\Omega \overrightarrow{\Psi}^T \delta\left(\frac{\partial \overrightarrow{f}_i}{\partial x_i}\right) d\Omega = -\int_\Omega \delta\overrightarrow{U}^T\left(A_i\frac{\partial \overrightarrow{\Psi}}{\partial x_i}\right) d\Omega + \int_\Omega \overrightarrow{\Psi}^T\frac{\partial f_i}{\partial x_i}\frac{\partial\left(\delta x_k\right)}{\partial x_i} d\Omega$$

$$+\int_{Sw}\Psi_i n_i \delta p dS+\int_{Sw}\left(\Psi_i p-\overrightarrow{\Psi}^T\overrightarrow{f}_i\right)\delta\left(n_i dS\right)$$
$$+\int_{S_i,o}\delta\overrightarrow{U}^T\left(A_n^T\overrightarrow{\Psi}\right)dS \quad (15)$$

Through the Green–Gauss theorem and the necessary rearangement of terms, the variation in F_{aug} takes the form

$$\delta F_{aug}=\frac{1}{2}\int_{S_w}(p-p_{tar})^2\delta(dS)-\int_{S_w}\frac{\partial\overrightarrow{U}}{\partial x_k}^T {A_n}^T\overrightarrow{\Psi}\delta x_k dS+\int_{S_w}(\Psi_{i+1}p-\overrightarrow{\Psi}^T\overrightarrow{f}_i)\delta(n_i dS) \quad (16)$$

Eq. 16 can be derived once the adjoint equations have been satisfied

$$\frac{\partial\overrightarrow{\Psi}}{\partial t}-{A_i}^T\frac{\partial\overrightarrow{\Psi}}{\partial x_i}=\overrightarrow{0} \quad (17)$$

with inlet / outlet boundary conditions given by

$$\delta\overrightarrow{U}^T(A_n^T\overrightarrow{\Psi})=\overrightarrow{0} \quad (18)$$

and the solid wall condition cast in the form

$$(p-p_{tar})+\Psi_{i+1}n_i=0 \quad (19)$$

5 The Continuous Adjoint Formulation–Viscous Terms

Starting from eq. (14), written for the viscous rather than the inviscid terms, and after some lengthy algebraic manipulations, the last term integral in eq. (2), that corresponds to the viscous terms, takes the form

$$\int_\Omega\overrightarrow{\Psi}^T\delta\left(\frac{\partial\overrightarrow{f}_{v,i}}{\partial x_i}\right)d\Omega=-\int_\Omega\left(\delta\overrightarrow{W}-\frac{\partial\overrightarrow{W}}{\partial x_k}\delta x_k\right)^T\overrightarrow{K}d\Omega-\int_\Omega\overrightarrow{\Psi}^T\frac{\partial}{\partial x_k}\left(\frac{\partial\overrightarrow{f}_{v,i}}{\partial x_i}\right)\delta x_k d\Omega+$$
$$\int_{S_w}\delta u_i\left[\mu\left(\frac{\partial\Psi_{j+1}}{\partial x_i}+u_j\frac{\partial\Psi_m}{\partial x_i}+\frac{\partial\Psi_{i+1}}{\partial x_j}+u_i\frac{\partial\Psi_m}{\partial x_j}\right)+\lambda\delta_{ij}\left(\frac{\partial\Psi_{k+1}}{\partial x_k}+u_k\frac{\partial\Psi_m}{\partial x_k}\right)-\Psi_4\tau_{ij}\right]n_j dS+$$
$$\int_{S_w}\delta T\left(k\frac{\partial\Psi_4}{\partial x_i}n_i\right)dS-\int_{S_w}\Psi_4\delta(q_j n_j dS)+\int_{S_w}\Psi_4 q_j\delta(n_j dS)-$$
$$\int_{S_w}\frac{\Psi_{i+1}}{n_i}[\delta\tau_{ij}n_i n_j+\tau_{ij}\delta(n_i n_j)]dS+\int_{S_w}\frac{\Psi_{i+1}}{n_i}\tau_{ij}\delta(n_i n_j)dS-\int_{S_w}u_i\Psi_4\delta\tau_{ij}n_j dS-$$
$$\int_{S_w}\frac{\partial u_i}{\partial x_l}\left[\mu\left(\frac{\partial\Psi_{j+1}}{\partial x_i}+u_j\frac{\partial\Psi_m}{\partial x_i}+\frac{\partial\Psi_{i+1}}{\partial x_j}+u_i\frac{\partial\Psi_m}{\partial x_j}\right)+\lambda\delta_{ij}\left(\frac{\partial\Psi_{k+1}}{\partial x_k}+u_k\frac{\partial\Psi_m}{\partial x_k}\right)\right]\delta x_l n_j dS+$$
$$\int_{S_w}\frac{\partial T}{\partial x_k}\left(k\frac{\partial\Psi_4}{\partial x_i}\right)\delta x_k n_i dS-\int_{S_w}\overrightarrow{\Psi}^T\frac{\partial\overrightarrow{f}_{v,i}}{\partial x_k}\delta x_k n_i dS \quad (20)$$

where

$$K_1 = -\frac{T}{\rho}\frac{\partial}{\partial x_i}\left(k\frac{\partial \Psi_4}{\partial x_i}\right)$$

$$K_{i+1} = \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \Psi_{j+1}}{\partial x_i}+u_j\frac{\partial \Psi_m}{\partial x_i}+\frac{\partial \Psi_{i+1}}{\partial x_j}+u_i\frac{\partial \Psi_m}{\partial x_j}\right)+\lambda\delta_{ij}\left(\frac{\partial \Psi_{k+1}}{\partial x_k}+u_k\frac{\partial \Psi_m}{\partial x_k}\right)\right]-\tau_{ij}\frac{\partial \Psi_4}{\partial x_j}$$

$$K_4 = \frac{T}{p}\frac{\partial}{\partial x_i}\left(k\frac{\partial \Psi_4}{\partial x_i}\right)$$

and $\overrightarrow{W} = [\rho, \overrightarrow{V}, p]^T$ is the vector of non–conservative variables

In inverse design problems, aiming at the minimization of function (3), the variation in F_{aug} gives

$$\delta F_{aug} = \frac{1}{2}\int_{S_w}(p-p_{tar})^2\delta(dS)+\int_{S_w}(\Psi_i p-\overrightarrow{\Psi}^T\overrightarrow{f}_i)\delta(n_i dS)+\int_{S_w}\frac{\Psi_{i+1}}{n_i}\tau_{ij}\delta(n_i n_j)dS+G \quad (21)$$

where

$$G = -\int_{S_w}\frac{\partial \overrightarrow{U}^T}{\partial x_k}(A_i{}^T n_i)\overrightarrow{\Psi}\delta x_k dS+\int_{S_w}\left(\overrightarrow{\Psi}^T\frac{\partial \overrightarrow{f}_{vi}}{\partial x_k}\right)\delta x_k n_i dS+\int_{S_w}\Psi_4 q_j\delta(n_j dS) \quad (22)$$

$$-\int_{S_w}\frac{\partial u_i}{\partial x_l}\left[\mu\left(\frac{\partial \Psi_{j+1}}{\partial x_i}+u_j\frac{\partial \Psi_4}{\partial x_i}+\frac{\partial \Psi_{i+1}}{\partial x_j}+u_i\frac{\partial \Psi_4}{\partial x_j}\right)+\lambda\delta_{ij}\left(\frac{\partial \Psi_{k+1}}{\partial x_k}+u_k\frac{\partial \Psi_4}{\partial x_k}\right)\right]\delta x_l n_j dS$$

The adjoint variable field is computed by satisfying the adjoint equation

$$\frac{\partial \overrightarrow{\Psi}}{\partial t}-A_i^T\frac{\partial \overrightarrow{\Psi}}{\partial x_i}-M^{-T}\overrightarrow{K} = \overrightarrow{0} \quad (23)$$

where $M = \frac{\partial U}{\partial W}$ is the transformation matrix from the non-conservative to the conservative variables. At the inlet and outlet, the condition

$$\delta\overrightarrow{U}^T\left(A_n^T\overrightarrow{\Psi}\right)dS = \overrightarrow{0} \quad (24)$$

should be satisfied whereas over the wall boundaries

$$\Psi_{i+1} = -(p-p_{tar})n_i \quad , i=1,2 \quad (25)$$

The wall boundary condition for Ψ_4 depends on the type of temperature conditions and is written as

$$\Psi_4 = 0 \quad or \quad \frac{\partial \Psi_4}{\partial n} = 0 \quad (26)$$

for constant temperature or adiabatic conditions respectively.

For design with minimum viscous losses, according to objective function (4), the variation in F_{aug} gives

$$\delta F_{aug} = -\int_{S_w} \left(\overrightarrow{\Psi}^T \overrightarrow{f}_i \right) \delta(n_i dS) - \int_{S_w} R_{ij} \frac{\partial u_j}{\partial x_k} n_i \delta x_k dS + \int_{S_w} \frac{1}{T} R \delta x_i n_i dS + G \quad (27)$$

where G is still given by eq. (22). The adjoint equation is

$$\frac{\partial \overrightarrow{\Psi}}{\partial t} - A_i^T \frac{\partial \overrightarrow{\Psi}}{\partial x_i} - M^{-T} \overrightarrow{K} - M^{-T} \overrightarrow{L} = \overrightarrow{0} \quad (28)$$

where vector $\overrightarrow{L} = (L_1, L_2, L_3, L_4)^T$ is defined by

$$L_1 = \frac{1}{T^2} R \frac{\partial T}{\partial \rho} \quad , \quad L_{i+1} = \frac{1}{T^2} R \frac{\partial T}{\partial u_i} \frac{\partial}{\partial x_j} \left(\frac{\mu}{T} R_{ij} \right) \quad , \quad L_4 = \frac{1}{T^2} R \frac{\partial T}{\partial E}$$

and

$$R = \tau_{ij} \frac{\partial u_i}{\partial x_j} \quad , \quad R_{ij} = 2(1+\delta_{ij}) \frac{\partial u_i}{\partial x_j} + 2(1-\delta_{ij}) \frac{\partial u_j}{\partial x_i} - \frac{4}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k}$$

The imposed boundary conditions are given by eqs. (24),(26) and (27); zero Dirichlet conditions should be imposed for Ψ_2, Ψ_3, instead of eq. (25).

During the design of realistic aerodynamic shapes, such as isolated or cascade airfoils, care subject to a number of geometrical constraints ($C_i \leq 0 \ , i = 1, \cdots, n$), often related to the thickness of the shape. These constraints are handled by setting up a new augmented function

$$F_{c,aug} = F_{aug} \ + \ F_c \quad (29)$$

where F_c is a functional expressing the degree of geometrical constraints violation. For inequality constraints, F_c is expressed by [18]

$$F_c \ = \ \sum_{i=1,n} \lambda_i \left(C_i + z_i^2 \right) \ + \ \sum_{i=1,n} \frac{\omega}{2} \left(C_i + z_i^2 \right)^2 \quad (30)$$

where

$$z_i^2 \ = \ max \left[0, -\left(\frac{\lambda_i}{\omega} + C_i \right) \right] \quad (31)$$

In each step (k) of the steepest descent algorithm, the Lagrange multiplier λ_i is updated as follows

$$\lambda_i^{k+1} = \lambda_i^k + \omega^k \left(C_i^k + z_i^2 \right) \quad (32)$$

6 Results and Discussion–Conclusions

The previously described adjoint methods are used in a series of shape optimization and inverse design problems, related to compressor and turbine blade airfoils. In both cases, reference and/or starting shapes are selected and the cascade geometries are defined by determining stagger angles (30^o for both compressor and turbine) and pitch–to–chord ratios (0.55 for the compressor, 0.8 for the turbine). In what follows, these cases are handled through either structured or unstructured grids; the latter are generated using advancing front routines with structured layers of triangular elements around the airfoil contour.

The parameterization of the airfoil is based on Bezier–Bernstein polynomials. The leading and trailing edges are fixed and the pressure and suction contours are separately modelled. For the parameterization of each compressor/turbine airfoil side 14/13 control points are used. All but the leading and trailing control points are allowed to vary. In the compressor case, these points are located at fixed chordwise positions and allowed to vary only along the normal–to–the–chord direction. The turbine control points are free to vary in either direction (both chordwise and normal–to–the–chord). Thus, compared to the compressor case, the turbine involves about twice as many design variables.

The flow conditions are: $\alpha_{in} = 50^o$ and $M_{out,is} = 0.32$ for the compressor and $\alpha_{in} = 0^o$ and $M_{out,is} = 0.50$ for the turbine cascade. As previously stated, inviscid, laminar and turbulent flow problems will be presented. In the laminar flow, the chord–based Reynolds number is $Re_c = 2000$. In the turbulent case, supported by the Spalart–Allmaras one–equation turbulence model, $Re_c = 10^6$.

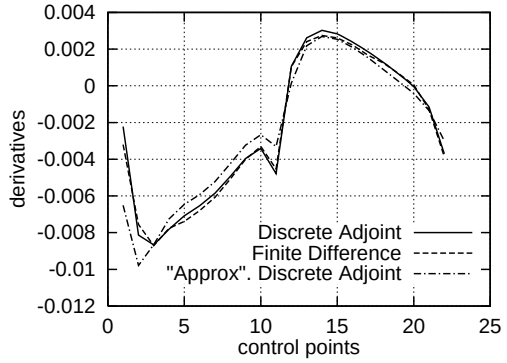


Figure 2: Inverse design of the compressor cascade, inviscid flow, unstructured grid: Objective function gradient computed using the exact (second–order accurate) discrete adjoint method, the "approximate" discrete adjoint method (second–order accuracy in the direct solver, first–order in the adjoint equations) and finite–differences.

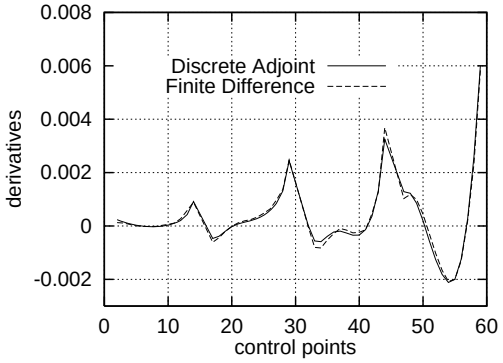
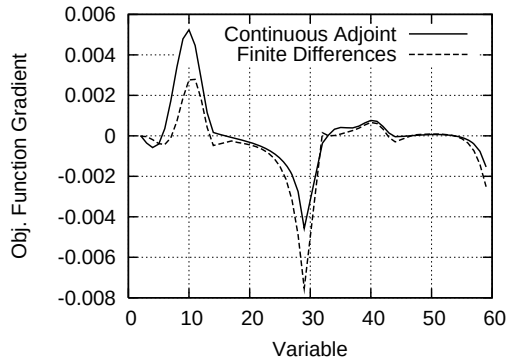


Figure 3: Inverse design of the turbine cascade, inviscid flow, unstructured grid: Objective function gradient computed using the discrete adjoint method and finite–differences.

Figures 2 to 6 show the objective function gradient values computed using the discrete

and continuous adjoint techniques in comparison to those estimated using central finite–differences. In general, the agreement is satisfactory. The only exception is the turbulent case, where the observed deviation between the two curves (fig. 6) can be attributed to the fact that the variation of the turbulent viscosity coefficient $\delta\mu_t$ was not taken into account. In the turbulent case, it is also worth noting that the finite–difference step used was selected "arbitrarily", i.e. without performing a parametric study as in the other cases; this was due to the high computational cost needed for such a study. However, despite the discrepancy between adjoint and finite–difference gradient components, the so computed derivatives support effectively the steepest descent method and lead to the optimal solution.



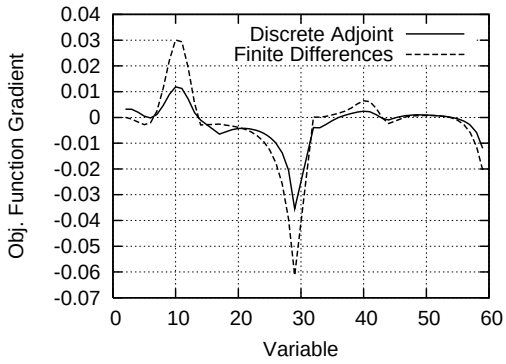


Figure 4: Losses minimization in the turbine cascade, laminar flow: Objective function gradient computed using the continuous adjoint method (left), the discrete adjoint method (right) and finite–differences.

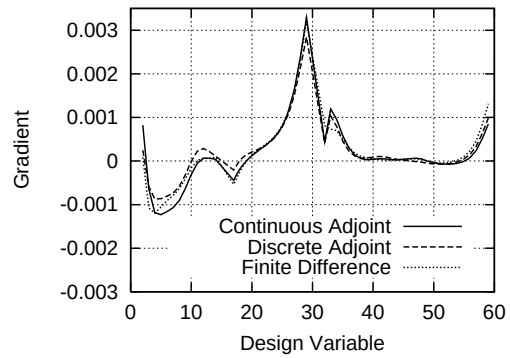


Figure 5: Inverse design of the turbine cascade, laminar flow: Objective function gradient components computed using the discrete adjoint method and finite–differences.

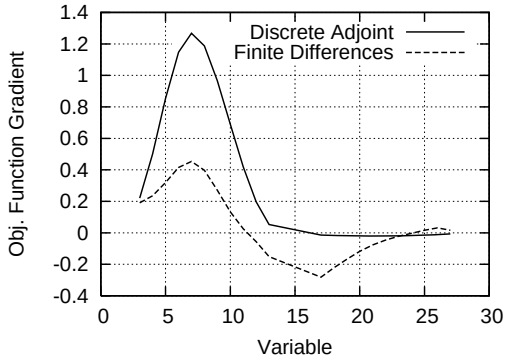


Figure 6: Losses minimization in the compressor cascade, turbulent flow: Comparison of the objective function gradient components computed using the discrete adjoint method and finite–differences.

The discrete adjoint method, especially if unstructured grids along with the finite–volume method are used, requires the storage of the transpose of $\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}$. As explained in a previous section, maintaining second–order accuracy in $\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}$ increases storage require-

ments which could be avoided if a first–order discretization was used. Note that the $\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}$ matrix is involved only in the l.h.s. term of the direct equations, so its level of accuracy does not affect the flow solution quality. However, this is not the case from the point of view of the adjoint method. This is the reason of comparing (fig. 2) the accuracy of the computed gradient using first–order $\left(\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}\right)_{0(\Delta x)}$ and second–order $\left(\frac{\partial \overrightarrow{R}}{\partial \overrightarrow{U}}\right)_{0(\Delta x^2)}$ matrices in the adjoint equation (the former is marked with "Approx." Discrete Adjoint, in the figure's caption). It can be seen that the first–order adjoint gives quite accurate derivatives; however, the second–order one performs much better and reproduces satisfactorily the values computed by means of finite–differences. The second–order discrete adjoint method is also used in the turbine case, fig. 3, where the finite–difference computed gradients are exactly reproduced by the adjoint method, using unstructured grids.

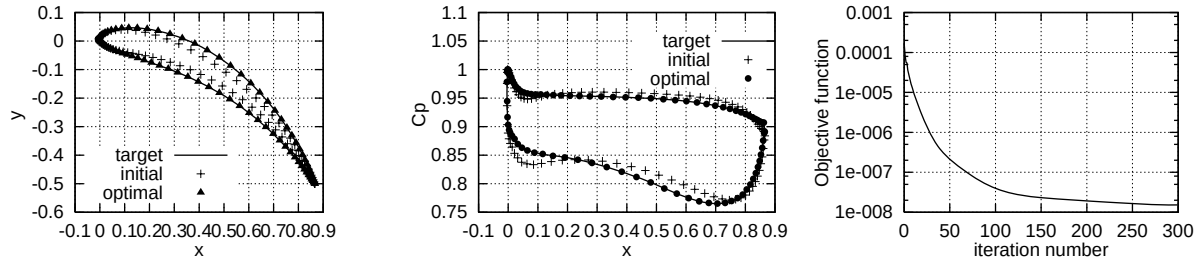


Figure 7: Inverse design of the turbine cascade, inviscid flow, unstructured grid. Left: reference, initial and optimal cascade shape , Middle: target, initial and optimal pressure distribution, Right: Convergence history.

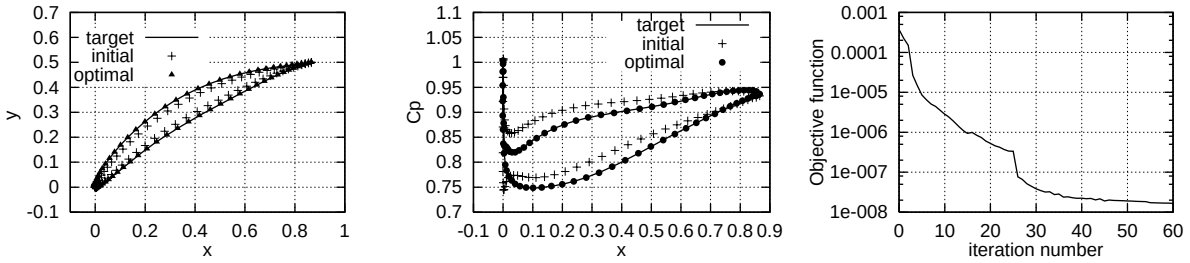


Figure 8: Inverse design of the compressor cascade, inviscid flow, unstructured grid. Same as fig. 7.

In the case of minimization of losses in the compressor cascade at laminar flow conditions, the gradient values computed using the adjoint technique are also compared to those computed by finite–differences, fig. 4. On the left, the continuous approach was used for the minimization of entropy generation while on the right the discrete adjoint method was employed for the minimization of total pressure losses. The comparison with finite–differences is quite reasonable; the observed small deviations do not affect the minimization algorithm convergence. In addition, there is a qualitative similarity between the derivatives for both objective functionals which, both theoretically and practically, are considered to be equivalent.

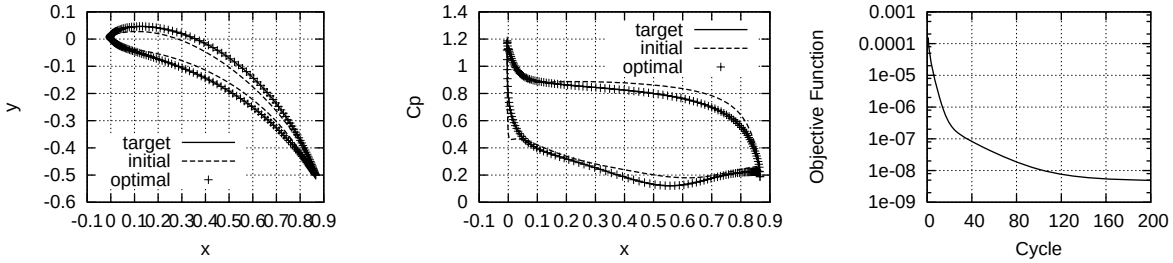


Figure 9: Inverse design of the turbine cascade, laminar flow. Same as fig. 7.

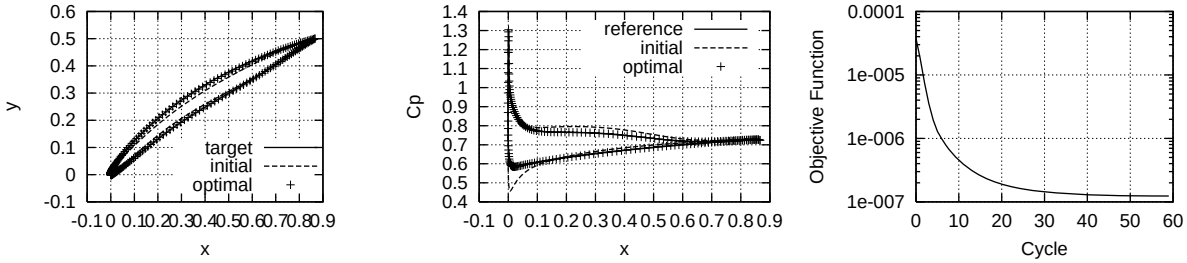


Figure 10: Inverse design of the compressor cascade, laminar flow. Same as fig. 7.

In the case of inverse design of the turbine cascade at laminar flow conditions, an even better coincidence of the gradient values computed with either the discrete or the continuous approach and those using finite–differences is observed in fig. 5. Finally, in the case of turbulent flow, fig. 6, although discrepancies can be observed in the computed derivatives, they can still be used successfully for the minimization of losses.

In figs. 7 to 10, one can observe the reference, initial and optimal blade contour (left),

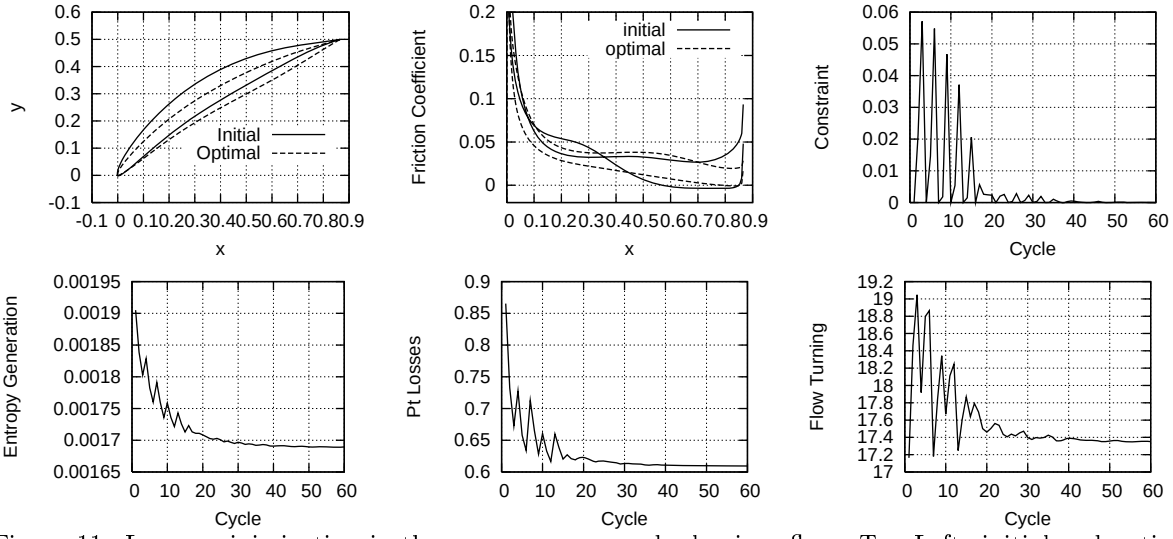


Figure 11: Losses minimization in the compressor cascade, laminar flow. Top Left: initial and optimal cascade shapes, Top Middle: initial and optimal friction coefficient distribution, Top Right: thickness deviations. Bottom from left to right : Entropy generation, total pressure losses and flow turning.

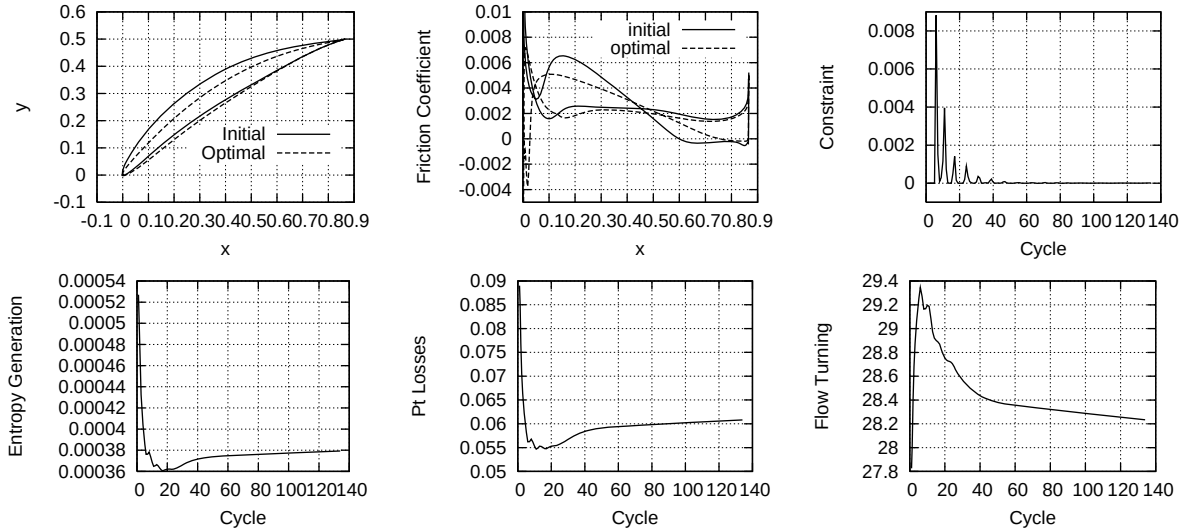


Figure 12: Losses minimization in the compressor cascade, turbulent flow. Same as in fig. 11.

the target, initial and optimal pressure distribution (middle) and the convergence history (right) for the inverse design of the compressor and the turbine blade in inviscid and laminar flow conditions.

In all cases, the reference blade geometry is totally reconstructed with the target and optimal pressure distributions being almost identical. The objective function is reduced by about three or four orders of magnitude. The excessive number of iterations required for convergence is due to the selection of the steepest descent step value; the main part of the reconstruction is completed within the first few iterations while the subsequent

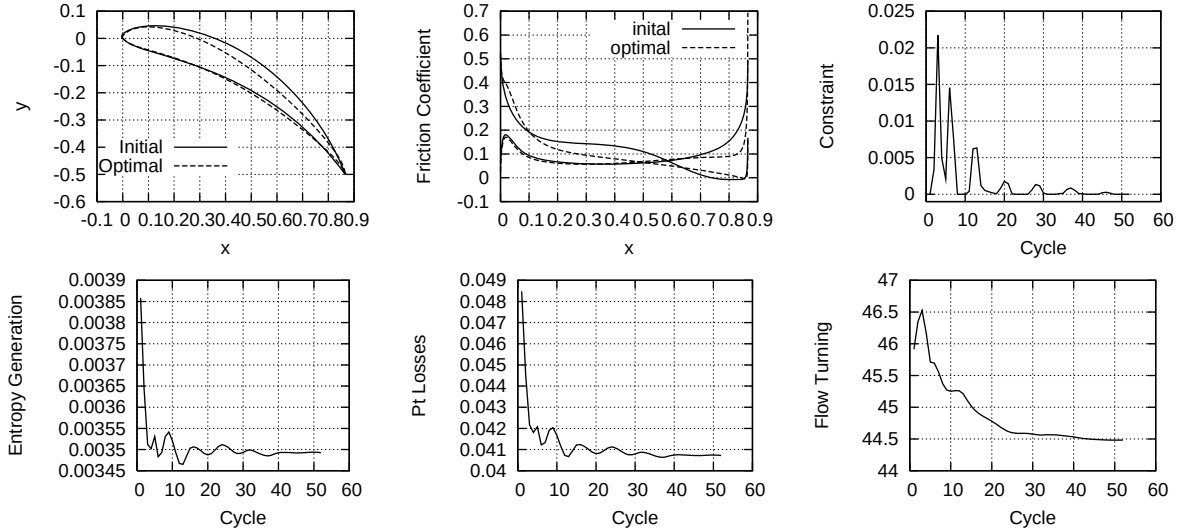


Figure 13: Losses minimization in the turbine cascade, laminar flow. Same as in fig. 11.

iterations yield only minor improvements.

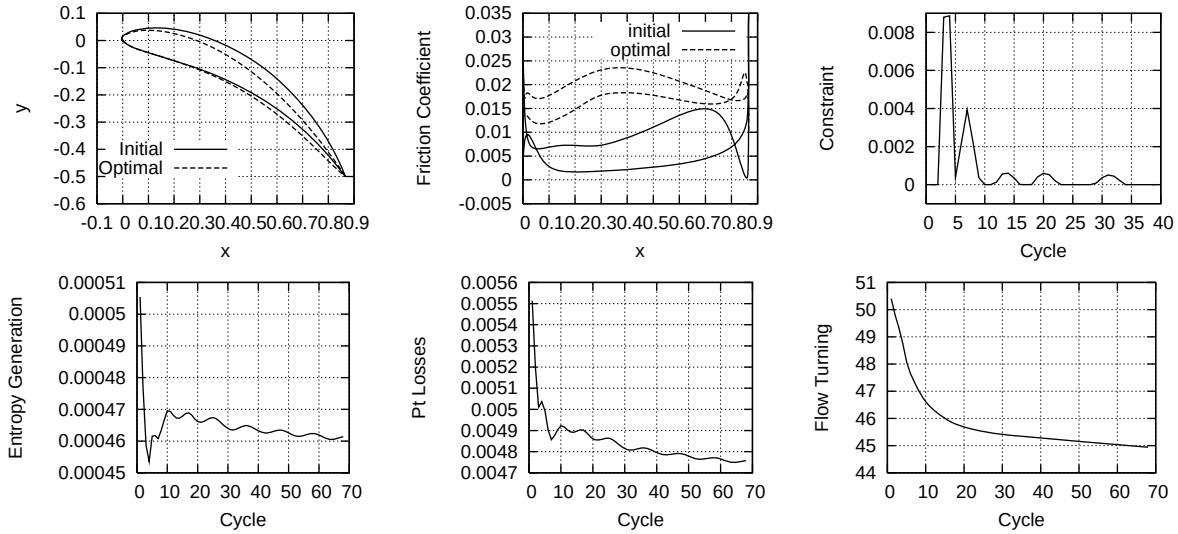


Figure 14: Losses minimization in the turbine cascade, turbulent flow. Same as in fig. 11.

Figures 11 to 14, refer to the viscous (laminar and turbulent) losses minimization cases, for the flow through the compressor and turbine cascades. In the top row, one may see: the initial and optimal cascade shapes, the initial and optimal friction coefficient distributions and the evolution of the term $\sum_{i=1,n} max\left(\tilde{t}_i - t_i\,, 0\right)$, that expresses the difference between the thickness of the blade airfoil and the minimum acceptable thickness (geometrical constraints). The bottom row shows the convergence histories of the entropy generation, total pressure losses and flow turning. Depending on the particular case, the entropy generation or the total pressure losses is used as the objective function and the gradient was computed using both the continuous and discrete adjoint method. One approach is shown for each case, for the sake of economy in space.

In all cases, a noticeable improvement in the friction coefficient distribution can be observed. Moreover, in the compressor cases, the flow separation on the initial airfoil is reduced or even disappears. The convergence history of both functionals presents the same tendency observed by the gradient values; regardless of the adjoint approach chosen, both functionals reduce their values in the same manner and show identical oscillations. These oscillations are due to the constraint violation and disappear after the final functional convergence. Note that the flow turning is just monitored without taking it into account during the optimization.

ACKNOWLEDGMENT

The project is co–funded by the European Social Fund (75%) and National Resources (25%) - Operational Program for Educational and Vocational Training (EPEAEK) and particularly the Program IRAKLITOS.

REFERENCES

[1] K.C. Giannakoglou, *Design of Optimal Aerodynamic Shapes using Stochastic Optimization Methods and Computational Intelligence*, Progress in Aerospace Sciences, 38, 43-76, 2002.

[2] I.C. Kampolis, D.I. Papadimitriou, K.C. Giannakoglou, *Evolutionary Optimization using a new Radial Basis Function Network and the Adjoint Formulation*, submitted for publication in the Journal of Inverse Problems in Science and Engineering, 2005.

[3] O. Pironneau, *Optimal Shape Design for Elliptic Systems*, Springer-Verlag, New-York, 1984.

[4] A. Jameson, *Aerodynamic Design via Control Theory*, Journal of Scientific Computation, 3, 33-260, 1988.

[5] A. Jameson, J. Reuther, *Control Theory Based Airfoil Design using the Euler Equations*, AIAA Paper 94-4272-CP, 1994.

[6] A. Jameson, J.J. Alonso, J.J. Reuther, L. Martinelli, J.C. Vassberg, *Aerodynamic Shape Optimization Techniques Based on Control Theory*, AIAA Paper 98-2538, 1998.

[7] A. Jameson, N. Pierce, L. Martinelli, *Optimum Aerodynamic Design Using the Navier–Stokes Equations*, J. Theor. Comp. Fluid. Mech, 10, 213-237, 1998.

[8] M.B. Giles, N.A. Pierce, *Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behaviour*, AIAA Paper 97-1850, 1997,

[9] W.K. Anderson, V. Venkatakrishnan, *Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation*, AIAA Paper 97-0643, 1997.

[10] E.J. Nielsen, W.K. Anderson, *Recent Improvements in Aerodynamic Design Optimization On Unstructured Meshes*, AIAA-2001-0596, 2001.

[11] J. Elliot, J. Peraire, *Aerodynamic Optimization on Unstructured Meshes wit Viscous Effects*, AIAA-97-1849, June 1997.

[12] H. Kim, D. Sasaki, S. Obayashi, K. Nakahashi, *Aerodynamic Optimization of Supersonic Transport Wing Using Unstructured Adjoint Method*, AIAA Journal Vol. 39, No. 6, June 2001.

[13] M.B. Giles, M.C. Duta, J. Müller, *Adjoint Code Developments Using the Exact Discrete Approach*, AIAA Paper 2001-2596, 15th Computional Fluid Dynamics Conference, Anaheim , California, June 2001.

[14] M.B. Giles, M.C. Duta, *Algorithm Developments for Discret Adjoint Methods*, AIAA Journal, Vol. 41, No2, February 2003.

[15] M.S. Campobasso, M.C. Duta, M.B. Giles, *Adjoint Methods for Turbomachinery Design*, ISABE-2001-1055.

[16] K.C. Giannakoglou, D.I. Papadimitriou, *A Continuous Adjoint Method for the Minimization of Losses in Cascade Viscous Flows*, submitted to AIAA CFD Conference, Reno-Nevada, 1/2006.

[17] D.I. Papadimitriou, K.C.Giannakoglou, *A Continuous Adjoint Method with Objective Function Derivatives Based on Boundary Integrals for Inviscid and Viscous Flows*, submitted for publication in the Journal of Computers and Fluids, February 2005.

[18] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, 1996.