Conference on Modelling Fluid Flow (CMFF'15)
The 16th International Conference on Fluid Flow Technologies
Budapest, Hungary, September 1-4, 2015

Evolutionary Algorithms and Adjoint-based CFD optimization in turbomachinery

K.C. Giannakoglou[1], V.G. Asouti[2], E.M. Papoutsis-Kiachagias[3]

[1] National Technical University of Athens. Iroon Polytechniou 9, 157 80 Athens, Greece. (+30)2107721636. kgianna@central.ntua.gr, web page: http://velos0.ltt.mech.ntua.gr/research/
[2] National Technical University of Athens. vasouti@mail.ntua.gr
[3] National Technical University of Athens. vaggelisp@gmail.com.

ABSTRACT

This paper deals with Computational Fluid Dynamics (CFD)-based shape optimization methods applied to gas and hydraulic turbomachines. In specific, two major optimization strategies, developed by the same group, are discussed: gradient–based methods (GBMs) supported by the continuous adjoint approach and metamodel–assisted evolutionary algorithms (MAEAs).

Regarding GBMs, the continuous adjoint method for the aero/hydrodynamic design of turbomachinery bladings is discussed. Full differentiation of turbulence models is considered. Recent developments allowing the computation of accurate sensitivity derivatives are presented in brief. Then, the continuous adjoint method is used for the shape optimization of two Francis turbine blades. The adjoint method for the optimization of thermal turbomachinery bladings, by taking into account conjugate heat transfer (CHT) effects, is also discussed.

Regarding MAEAs, emphasis is laid on the ways used to reduce the overall CPU cost of a CFD-based optimization. In particular, the efficient use of on–line trained surrogate evaluation models (or metamodels), the use of asynchronous search on multiprocessor platforms and the use of Principal Component Analysis (PCA) as remedies to the curse of dimensionality problem are discussed. MAEAs are demonstrated in the aero/hydrodynamic shape optimization of turbomachinery bladings.

Keywords: Computational Fluid Dynamics, Continuous Adjoint Method, Gradient–based methods, Metamodel Assisted Evolutionary Algorithms, Thermal and Hydraulic Turbomachines

NOMENCLATURE

F	[varies]	objective function
T	[K]	temperature
T_a	[F-related]	adjoint temperature
$\underline{b}$	[varies]	design variables
$\underline{u}$	[F-related]	adjoint velocity
$\underline{v}$	$[m/s]$	absolute velocity
$\underline{w}$	$[m/s]$	relative velocity
p	$[m^2/s^2]$	static pressure divided by density
q	[F-related]	adjoint pressure
λ	[-]	offsprings
μ	[-]	parents
ν	$[m^2/s]$	bulk viscosity
ν_t	$[m^2/s]$	turbulent viscosity
$\widetilde{\nu_a}$	[F-related]	adjoint turbulence model variable
$\widetilde{\nu}$	$[m^2/s]$	Spalart–Allmaras model variable

1. INTRODUCTION

During the last years, the cost benefits resulting from using CFD has given rise to an intense academic and industrial interest in the use of computational methods for the design/optimization of thermal and hydraulic turbomachines.

CFD–based optimization methods can be classified into deterministic and stochastic ones, according to the strategy used to compute the optimal set of design variables. Deterministic algorithms start with a given geometry and improve it iteratively based on the computed or approximated gradient of the objective function with respect to (w.r.t.) the design variables. Depending on the initialization, it is not unlikely for a GBM to be trapped into a local optimum. In such a case, the designer will get an optimized rather than an optimal solution. Though global optimal solutions are always the target, in practice local optima are highly welcome. The efficiency of GBMs greatly depends on the method used to compute the necessary gradient. In this respect, the adjoint method [1] has been receiving a lot of attention, since the cost of computing the gradient is, practically, independent from the number of the design variables. This makes the method an excellent choice for large scale industrial optimization problems. In this paper, recent advances in computing accurate sensitivity derivatives for turbulent flows using the continuous adjoint variant are discussed, [2]. In addition, a short discussion about adjoint methods for CHT applications is presented followed by industrial applications.

Evolutionary algorithms (EAs) are the most popular representative of stochastic population–based search methods. In EAs, entrapment to local minima is highly unlikely, unless the search is stopped early enough, since almost the entire design space can be explored. EAs are extremely flexible since the evolution operators do not interfere with the flow solver; so, in CFD–based optimization, no access to the source CFD code is required (black–box evaluation software). Furthermore, EAs can compute Pareto fronts of non-dominated solutions in multi–objective optimization (MOO) problems, with a single run. On the other hand, a great number of candidate solutions must be evaluated before reaching the optimal one(s), leading to a high optimization turnaround time, especially when the evaluation software is costly (such as in CFD applications). In addition, the number of evaluations required increases with the number of the design variables (curse of dimensionality). A number of remedies have been proposed in the literature to tackle the aforementioned two weaknesses of EAs. Among them is the use Metamodel–Assisted EAs (MAEAs), asynchronous search, performed on a cluster of many processors, and the use Principal Component Analysis (PCA) to identify correlations between the design variables, [3, 4, 5]. Industrial applications using the above techniques are presented.

2. ADJOINT METHODS

In this section, the formulation of the continuous adjoint PDEs, their boundary conditions and the sensitivity derivatives (gradient) expression are presented in brief. The development is based on the incompressible Navier-Stokes equations for a non-inertial Single Rotating Frame (SRF), though their extension to inertial reference systems, [2], exists too. The development for incompressible flows is based on OpenFOAM©. However, the same tools have been programmed also for compressible flows, [6], on an in–house CDF code, running on GPUs, [7].

2.1. Flow Equations

The mean flow equations together with the Spalart–Allmaras turbulence model PDE, [8], comprise the flow or primal system of equations that reads

$$R^p = -\frac{\partial w_i}{\partial x_i} = 0 \tag{1a}$$

$$R_i^w = w_j \frac{\partial w_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + \underbrace{2e_{ijk}\Omega_j w_k}_{C_R}$$
$$+ e_{ijk}\Omega_j e_{klm}\Omega_l x_m = 0 \tag{1b}$$

$$R^{\widetilde{\nu}} = \frac{\partial (w_j \widetilde{\nu})}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}}{\partial x_j}\right] - \frac{c_{b2}}{\sigma}\left(\frac{\partial \widetilde{\nu}}{\partial x_j}\right)^2$$
$$- \widetilde{\nu} P\left(\widetilde{\nu}, \Delta\right) + \widetilde{\nu} D\left(\widetilde{\nu}, \Delta\right) = 0 \tag{1c}$$

where w_i, Ω_j, x_m are the components of the relative velocity vector, rotational speed vector and position vector, respectively. The absolute (v_i) and relative (w_i) velocities are related through $v_i = w_i + e_{ijk}\Omega_j x_k$. Also, p is the static pressure divided by the constant density, $\tau_{ij} = (\nu + \nu_t)\left(\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i}\right)$ are the components of the stress tensor, ν and ν_t the bulk and turbulent viscosity, respectively, $\widetilde{\nu}$ the Spalart–Allmaras model variable and Δ the distance from the wall boundaries. Details about the turbulence model constants and source terms can be found in [8].

2.2. General Objective Function

Let F be the objective function to be minimized by computing the optimal values of the design variables $b_n, n \in [1, N]$. A general expression for an objective function defined on (parts of) the boundary S of the computational domain Ω is given by

$$F = \int_S F_{S_i} n_i dS \tag{2}$$

where $\mathbf{n}$ is the outward normal unit vector.

Differentiating Eq. 2 w.r.t. to b_n and applying the chain rule yields

$$\frac{\delta F}{\delta b_n} = \int_S \frac{\partial F_{S_i}}{\partial w_k} n_i \frac{\partial w_k}{\partial b_n} dS + \int_S \frac{\partial F_{S_i}}{\partial p} n_i \frac{\partial p}{\partial b_n} dS$$
$$+ \int_S \frac{\partial F_{S_i}}{\partial \tau_{kj}} n_i \frac{\partial \tau_{kj}}{\partial b_n} dS + \int_S \frac{\partial F_{S_i}}{\partial \widetilde{\nu}} n_i \frac{\partial \widetilde{\nu}}{\partial b_n} dS$$
$$+ \int_S n_i \frac{\partial F_{S_i}}{\partial x_k} \frac{\delta x_k}{\delta b_n} n_k dS + \int_S F_{S_i} \frac{\delta (n_i dS)}{\delta b_n} \tag{3}$$

where $\delta\Phi/\delta b_n$ is the total (or material) derivative of any quantity Φ while $\partial\Phi/\partial b_n$ is its partial derivative. Operators $\delta()/\delta b_n$ and $\partial()/\partial b_n$ are related by

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k}\frac{\delta x_k}{\delta b_n} \tag{4}$$

Computing the variation of the flow variables on the r.h.s. of Eq. 3, either through Direct Differentiation (DD) or Finite Differences (FD) would require at least N Equivalent Flow Solutions (EFS, i.e. as if the flow equations were solved instead). To avoid this computational cost that scales with N, the adjoint method is used, as presented in the next subsection.

2.3. Continuous Adjoint Formulation

Starting point of the continuous adjoint formulation is the introduction of the augmented function

$$F_{aug} = F + \int_\Omega u_i R_i^w d\Omega + \int_\Omega q R^p d\Omega + \int_\Omega \widetilde{\nu_a} R^{\widetilde{\nu}} d\Omega \tag{5}$$

where u_i are the components of the adjoint to the relative velocity vector, q is the adjoint pressure and

$\widetilde{\nu_a}$ is the adjoint turbulence model variable, respectively. Dropping the last integral on the r.h.s. of Eq. 5 would result to the so-called "frozen turbulence" assumption which neglects the differentiation of the turbulence model PDE(s). This assumption leads to reduced gradient accuracy, possibly even to wrong sensitivity signs, [9]. To avoid the "frozen turbulence" assumption implications, the Spalart–Allmaras model PDE has been differentiated, see [9]. A review on continuous adjoint methods for turbulent flows can be found in [2].

The differentiation of Eq. 5, based on the Leibniz theorem, yields

$$\frac{\delta F_{aug}}{\delta b_n} = \frac{\delta F}{\delta b_n} + \int_\Omega u_i \frac{\partial R_i^w}{\partial b_n} d\Omega + \int_\Omega q \frac{\partial R^p}{\partial b_n} d\Omega + \int_\Omega R^{\widetilde{\nu}} \frac{\partial R^{\widetilde{\nu_a}}}{\partial b_n} d\Omega + \int_{S_W} (u_i R_i^w + q R^p + \widetilde{\nu_a} R^{\widetilde{\nu}}) n_k \frac{\delta x_k}{\delta b_n} dS \tag{6}$$

Then, the derivatives of the flow residuals in the volume integrals on the r.h.s. of Eq. 6 are developed by differentiating Eqs. 1 and applying the Green-Gauss theorem, where necessary. Indicatively, the development of the C_R (Coriolis) term variation yields

$$\int_\Omega u_i \frac{\partial C_{R,i}}{\partial b_n} d\Omega = -\int_\Omega 2 e_{ijk} \Omega_j u_k \frac{\partial w_i}{\partial b_n} d\Omega \tag{7}$$

contributing an extra term to the adjoint momentum equations. The development of the remaining terms can be found in [9], [10] and [2].

In order to obtain a gradient expression which does not depend on the partial derivatives of the flow variables w.r.t. b_n, their multipliers in (the developed form of) Eq. 6 are set to zero, giving rise to the field adjoint equations

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \tag{8a}$$

$$R_i^w = u_j \frac{\partial w_j}{\partial x_i} - \frac{\partial (w_j u_i)}{\partial x_j} - \frac{\partial \tau_{ij}^a}{\partial x_j} + \frac{\partial q}{\partial x_i} \underbrace{-2 e_{ijk} \Omega_j u_k}_{C_{R_a}} + \widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l} \left(\widetilde{\nu_a} \widetilde{\nu} \frac{C_Y}{Y} e_{mjk} \frac{\partial w_k}{\partial x_j} e_{mli} \right) = 0 \tag{8b}$$

$$R^{\widetilde{\nu_a}} = -\frac{\partial (w_j \widetilde{\nu_a})}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} \right] + \frac{1}{\sigma} \frac{\partial \widetilde{\nu_a}}{\partial x_j} \frac{\partial \widetilde{\nu}}{\partial x_j} + 2 \frac{c_{b2}}{\sigma} \frac{\partial}{\partial x_j} \left(\widetilde{\nu_a} \frac{\partial \widetilde{\nu}}{\partial x_j} \right) + \widetilde{\nu_a} \widetilde{\nu} C_{\widetilde{\nu}} + \frac{\partial \nu_t}{\partial \widetilde{\nu}} \frac{\partial u_i}{\partial x_j} \left(\frac{\partial w_i}{\partial x_j} + \frac{\partial w_j}{\partial x_i} \right) + (-P + D) \widetilde{\nu_a} = 0 \tag{8c}$$

where $\tau_{ij}^a = (\nu + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ are the components of adjoint stress tensor. The term marked as C_{R_a} results from the differentiation of C_R and can be seen as the adjoint Coriolis acceleration. Eq. 8c is the adjoint turbulence model equation, from which the adjoint turbulence model variable $\widetilde{\nu_a}$ is computed.

The adjoint boundary conditions are derived by treating the flow variations in the boundary integrals (of the developed form of) Eq. 6, [9, 2]. Indicatively, at the inlet (S_I) and wall (S_W) boundaries, the following conditions are imposed

$$u_j n_j = u_{\langle n \rangle} = -\frac{\partial F_{S_{I-W,i}}}{\partial p} n_i \tag{9a}$$

$$u_j t_j^I = u_{\langle t \rangle}^I = \frac{\partial F_{S_{I-W,k}}}{\partial \tau_{ij}} n_k t_i^I n_j + \frac{\partial F_{S_{I-W,k}}}{\partial \tau_{ij}} n_k t_j^I n_i \tag{9b}$$

$$u_j t_j^{II} = u_{\langle t \rangle}^{II} = \frac{\partial F_{S_{I-W,k}}}{\partial \tau_{ij}} n_k t_i^{II} n_j + \frac{\partial F_{S_{I-W,k}}}{\partial \tau_{ij}} n_k t_j^{II} n_i \tag{9c}$$

where S_{I-W} stands for either S_I or S_W, depending on the boundary under consideration. In what follows, $\mathbf{t^I}$ is the unit tangent vector parallel to the velocity at the first cell centre off the boundary and the components of the second target vector $\mathbf{t^{II}}$ are given by $t_i^{II} = e_{ijk} n_j t_k^I$, where e_{ijk} is the Levi-Civita symbol. The outlet (S_O) conditions for the adjoint problem and boundary conditions for the $\widetilde{\nu_a}$ field can be found in [9, 2].

In industrial applications, the wall function technique is used routinely in analysis and design. When the design is based on the adjoint method, considering the adjoint to the wall function model becomes necessary. The continuous adjoint method in optimization problems, governed by the RANS turbulence models with wall functions, was initially presented in [11], where the adjoint wall function technique was introduced for the $k - \epsilon$ model and a vertex–centered finite volume method. The proposed formulation led to a new concept: the "adjoint law of the wall". This bridges the gap between the solid wall and the first node off the wall during the solution of the adjoint equations. The adjoint wall function technique has been extended to flow solvers based on cell-centered finite-volume schemes, for the $k - \omega$, [12], and Spalart–Allmaras , [2], models.

After satisfying the adjoint PDEs and their boundary conditions, the remaining terms in Eq. 6 yield the sensitivity derivatives

$$\begin{aligned}
\frac{\delta F}{\delta b_n} = & -\int_{S_W} \left[\tau_{ij}^a n_j - q n_i + \frac{\partial F_{S_{W,k}}}{\partial w_i} n_k \right] \frac{\partial w_i}{\partial x_k} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_W} n_i \frac{\partial F_{S_{W,i}}}{\partial x_k} \frac{\delta x_k}{\delta b_n} dS + \int_{S_W} F_{S_{W,i}} \frac{\delta (n_i dS)}{\delta b_n} \\
& - \int_{S_W} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu_a}}{\partial x_j} n_j + \frac{\partial F_{S_k}}{\partial \widetilde{\nu}} n_k \right] \frac{\partial \widetilde{\nu}}{\partial x_k} \frac{\delta x_k}{\delta b_n} dS \\
& + \int_{S_W} (u_i R_i^w + q R^p + \widetilde{\nu_a} R^{\widetilde{\nu}}) \frac{\delta x_k}{\delta b_n} n_k dS \\
& - \int_{S_W} \left(-u_{\langle n \rangle} + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k n_l n_m \right) \mathcal{T} \mathcal{S}_1 dS \\
& - \int_{S_W} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^I t_m^I \mathcal{T} \mathcal{S}_2 dS \\
& - \int_{S_W} \left(\frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k (t_l^{II} t_m^I + t_l^I t_m^{II}) \right) \mathcal{T} \mathcal{S}_3 dS
\end{aligned}$$

$$
-\int_{S_W} \frac{\partial F_{S_{W_{p,k}}}}{\partial \tau_{lm}} n_k t_l^{II} t_m^{II} \mathcal{TS}_4 dS \tag{10}
$$

where $\mathcal{TS}_1$ to $\mathcal{TS}_4$ can be found in [2].

2.4. Differentiation of Turbulence Models: A Convincing Example

In the application of this section, the gain from overcoming the "frozen turbulence" assumption is discussed. In Figure 1, the sensitivity derivatives of the total pressure losses objective function,

$$
F_{pt} = -\int_{S_{I,O}} \left(p + \frac{1}{2}v_k^2\right) v_i n_i dS
$$

w.r.t. the coordinates of Bézier–Bernstein control points parameterizing a compressor cascade airfoil are illustrated. Here, the low-Re variant of the Spalart–Allmaras model is used. It can be seen that the "frozen turbulence" assumption leads to quite wrong sensitivities while the adjoint approach that takes into consideration the differentiation of the turbulence model reproduces the outcome of the reference method (FD). More on the gain in accuracy from using the adjoint law of the wall when the flow simulation employs wall functions can be found in [2].

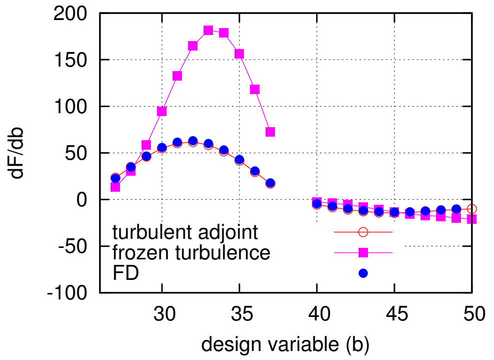

Figure 1. Shape optimization of a compressor cascade with $Re = 3.3 \times 10^5$. Sensitivity derivatives of the total pressure losses function F w.r.t. the coordinates of the Bézier–Bernstein control points parameterizing the suction (first half) and pressure (last half of the horizontal axis) airfoil sides.

2.5. Continuous Adjoint for Conjugate Heat Transfer Analysis

This section discusses some points of the mathematical development and implementation of the continuous adjoint method to CHT applications. CHT comprises the concurrent solution of the mean flow and energy equations over the fluid domain and the energy equation over an adjacent solid domain. The fluid and solid domains communicate through the Fluid/Solid Interface (FSI). The conditions imposed along the FSI boundary, Figure 2, are (index F stands for fluid-related quantities and S for solid-related ones)

$$
Q^S = -Q^F \Rightarrow k^S \left.\frac{\partial T^S}{\partial n}\right|_{FSI_S} = -k^F \left.\frac{\partial T^F}{\partial n}\right|_{FSI_F} \tag{11a}
$$

$$
T^S = T^F = T^{FSI} \tag{11b}
$$

where Q is the heat-flux and $k^F = a_{eff} c_p$ is the thermal conductivity. Eq. 11a expresses the equality of heat fluxes along the FSI while Eq. 11b states that the temperature at the coinciding nodes of the solid and fluid meshes is the same.

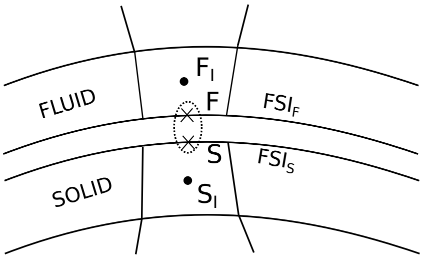

Figure 2. The fluid(F)/solid(S) interface. Faces F and S coincide. F_I and S_I are the centres of the first cells off the fluid and solid boundaries, respectively.

In the application examined, the optimization aims at minimizing the maximum temperature inside an internally cooled turbine cascade, Figure 3. Since such a min./max. objective can not be differentiated, a differentiable surrogate should be used. It is proposed to use a sigmoid function

$$
F_T = \frac{\int_{\Omega_S} f_{sig} d\Omega}{\int_{\Omega_S} d\Omega}, \; f_{sig} = 1 - \frac{1}{1 + e^{k_2(T-T_c)+k1}} \tag{12}
$$

where Ω_S is the volume of the solid domain, T_c is a critical (high) temperature threshold and $T_s < T_c$ is a safety threshold to be defined by the designer. Constants k_1 and k_2 take on values that lead to $f_{sig}(T_s) = \epsilon$ and $f_{sig}(T_c) = 1 - \epsilon$, where ϵ is a user–defined infinitesimal positive number.

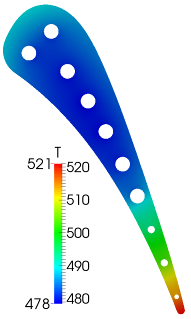

Figure 3. Temperature distribution inside an internally cooled turbine blade. The fluid (not shown in the figure) and solid domains are coupled based on the boundary conditions presented in Eqs. 11, while heat exchange between the solid domain and the cooling passages is simulated using a $1D$ heat exchange equation.

The augmented objective function for CHT optimization problems is written as

$$F_{aug} = F + \int_{\Omega_F} u_i R_i^w d\Omega + \int_{\Omega_F} q R^p d\Omega + \int_{\Omega_F} \widetilde{\nu}_a R^{\widetilde{\nu}} d\Omega$$
$$+ \int_{\Omega_F} T_a^F R^{T^F} d\Omega + \int_{\Omega_S} T_a^S R^{T^S} d\Omega \qquad (13)$$

where R^{T^F}, R^{T^S} are the energy equation PDEs over the fluid and solid domains, respectively, and T_a^F, T_a^S are the corresponding adjoint temperatures. Following a process similar to that described in sections 2.2 to 2.4, the field adjoint equations, adjoint boundary conditions and sensitivity derivatives expression can be derived. In the interest of space, these are omitted herein. However, it is interesting to note that the adjoint boundary conditions at the FSI are of the same type as the primal conditions. Eq. 11. They include the conservation of the adjoint heat flux at the FSI and the same adjoint temperature values for both sides of the FSI. This remark holds only for objective functions which do not include the temperature values along the FSI.

The continuous adjoint approach to CHT problems was utilized to support a gradient-based algorithm to minimize F_T with $T_s = 515\ K$ and $T_c = 525\ K$ for the geometry presented in Figure 3; the turbine blade airfoil is parameterized using NURBS control points and the cooling holes are at fixed positions. The highest deformation is located close to the trailing edge, Figure 4, decreasing the maximum blade temperature by more than 2 K.

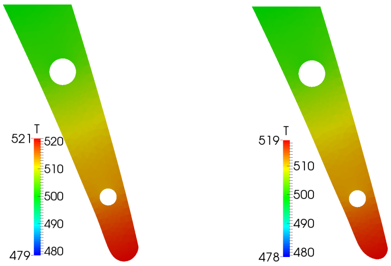

Figure 4. Shape optimization of an internally cooled turbine blade, targeting the minimization of the maximum temperature over the solid. Temperature distributions in a blow-up view close to the trailing edge of the initial (left) and optimized (right) geometries.

2.6. *Turbomachinery Applications of Adjoint-based Optimization*

Two industrial applications are presented in this section. The first one is concerned with the shape optimization of a Francis turbine runner in order to suppress cavitation, i.e. maximizing the minimum pressure on the blade surface. Following the same line of reasoning for differentiating a max./min problem as the one presented in section 2.5, a sigmoid function similar to Eq. 12 is used, by defining a cavitation threshold p_c and a safety threshold p_s. No shape parameterization was used. Instead, the normal displacements of the blade wall nodes acted as the design variables, after appropriately smoothing the computed sensitivity derivatives. The pressure distributions over the initial and optimized bladings are presented in Figure 5.

The second application deals with the multi-point design of a different Francis runner targeting the maximization of the weighted sum of the efficiencies at three operating points, ranging from 40% to 100% of the nominal mass flow rate Q_{nom}

$$F = 0.6F_{Q_{100}} + 0.25F_{Q_{71.5}} + 0.15F_{Q_{40}} \qquad (14)$$

Blade shapes resulting after 8 optimization cycles of the multi-point as well as the three (separate) single-point optimizations (for the three mass flow rates) are depicted in Figure 6. The multi-point optimization deforms the blade in the same direction as the single-point optimization for the nominal flow rate, since this point has the highest weight in Eq. 14. The other two single-point optimizations for $Q = 0.715Q_{nom}$ and $Q = 0.40Q_{nom}$ deform the blade in the opposite direction. This clearly reveals the contradictory targets in multi-point optimization.

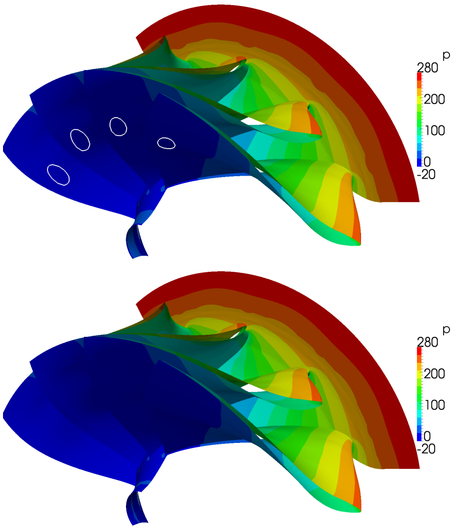

Figure 5. Optimization of a Francis runner blade targeting cavitation suppression. Top: pressure distribution over the initial blading; white isolines encircle the cavitated areas. Bottom: pressure distribution over the optimized blading.

3. OPTIMIZATION METHODS BASED ON EAS

Regarding EAs, emphasis is laid on ways to reduce the optimization turnaround time in large–scale applications. The most frequently used technique is the use of surrogate evaluation models (metamodels) giving rise to MAEAs. Either EAs or MAEAS can further be enhanced by the Principal Component Analysis (PCA) technique aiming at efficiently handling problems involving a great number of design

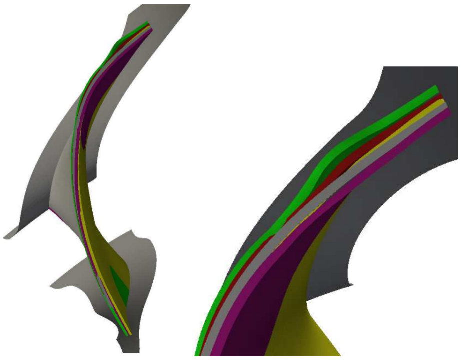

Figure 6. Multi-point optimization of a Francis runner Blades as seen from the trailing edge (left) and a blow–up view close to the shroud (right). The initial blade is depicted in grey, the result of the multi-point optimization in red, while the results of single-point optimizations at the three operating points are shown in green ($Q=Q_{nom}$), yellow ($Q=0.715Q_{nom}$) and magenta ($Q=0.4Q_{nom}$).

variables. Over and above to MAEAs (with or without PCA), the concurrent evaluation of the offspring of each generation on the available processors of a multi-processor system may further reduce the optimization turnaround time. Asynchronous EAs (AEAs), remove the synchronization barrier at the end of each generation, and fully exploit all the available computational resources. All these techniques are incorporated in the general purpose optimization platform EASY (Evolutionary Algorithm SYstem, http://velos0.ltt.mech.ntua.gr/EASY) developed by the authors' group.

On–line trained metamodels (radial basis function/RBF networks) for each candidate solution are used according to the Inexact Pre-Evaluation (IPE) technique, [3]. The first few generations are carried out as a conventional EA (with μ parents and λ offspring) and the MAEA starts once a user–defined number of entries have been stored in the database (DB) of already evaluated individuals. During the IPE phase all population members are pre–evaluated on the surrogate models trained on–the–fly. This training is carried out on the neighboring (in the design space) individuals in the DB. Then, based on the outcome of the pre–evaluations on the metamodels, a small number of the most promising members ($\lambda_{IPE} \ll \lambda$) are re–evaluated on the CFD model.

Metamodels can be also employed in AEAs (AMAEAs), after appropriately adapting the IPE scheme, [4], since the notion of generation does not exist anymore. Once an evaluation is completed and the corresponding processor is idle, a new individual is generated (using the evolution operators) and assigned to this processor. When the IPE is activated on an instantaneously idle processor, instead of generating a single individual, a small number (N_{IPE}) trial ones are generated. For each one of them, a local metamodel is trained and its objective function value is approximated. The best (according to the metamodel) among the N_{IPE} individuals is, then, re–evaluated on the exact model. An example of the gain in the optimization turnaround time by using AMAEA instead of MAEA, is shown in 7. This is concerned with the design of a peripheral compressor cascade for minimum viscous losses, where AMAEA and MAEA were allowed to perform up to 12 concurrent evaluations on a many–GPUs platform. The IPE was activated after 80 entries were gathered in the DB and λ_{IPE} = 8 members were re–evaluated on the exact tool for the MAEA. For the AMAEA, $N_{IPE}=8$ trial members were generated before selecting the one to be re–evaluated on the idle processor.

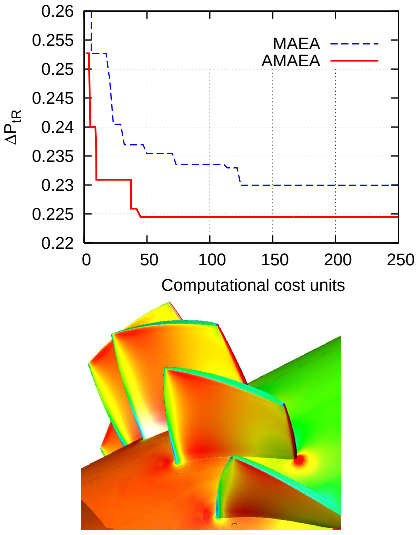

Figure 7. Optimization of a peripheral compressor cascade for minimum viscous losses. Top: Comparison of the convergence histories of MAEA and AMAEA. AMAEA outperforms MAEA which is known to perform much better than a conventional EA. Bottom: Pressure distribution on the optimal geometry.

EAs or MAEAs (along with their asynchronous variants), when applied to engineering optimization problems with a great number of design variables, suffer from the co–called "curse of dimensionality". A remedy to this problem is to process the elite set in each generation, using PCA and, based on the so acquired information to: (a) better guide the application of the evolution operators (to be referred as EA(PCA)) and (b) reduce the number of sensory units during the metamodels training (M(PCA)AEA), [5]. In (a), the design space is temporarily aligned with the principal component directions and the crossover and mutation operators are applied on the rotated individuals. This rotation according to the principal directions leads to a problem with as much as possible separable objective function, which is highly beneficial. In MAEAs, during the metamodel (RBF network) training, PCA can be used to reduce the dimension of the metamodels built. The variances of the design variables are used to identify the directions along which the elite members are less or more scattered. In the developed method, the RBF network sensory units corresponding to the directions of the design space with high variances are filtered out. Reducing the number of input parameters in the metamodel increases the prediction accuracy and accelerates the training process. The simultaneous use of PCA for both purposes is the so–called M(PCA)AEA(PCA).

An example of the use of PCA in both the metamodels and the evolution operators is shown in figure 8. This is concerned with the two–objective constrained design of a Francis runner parameterized using 372 design variables. The first objective (f_1) is related to the "quality" of the velocity profile at the runner outlet while the second one (f_2) to the blade loading. This case is studied with both MAEA and M(PCA)AEA(PCA) using a (μ,λ)=(20, 90) EA. During the IPE phase, $\lambda_{IPE} = 8$ members of each generation were re–evaluated on the CFD model. For the MAEA, the IPE phase was activated after 600 entries were stored in the DB, while for the M(PCA)AEA(PCA) only after 300 DB entries.

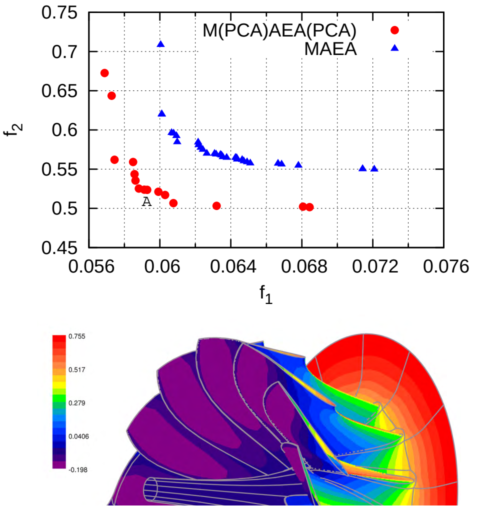

Figure 8. Two–objective design of a Francis runner at three operating points for optimizing the outlet velocity profile (min f_1) and the blade loading (min f_2). Top: Comparison of the fronts of non–dominated solutions computed by the MAEA and the M(PCA)AEA(PCA), at the same CPU cost. Bottom: 3D view and pressure field over the Francis runner, at the best efficiency operating point corresponding to non–dominated solution A.

4. USE OF EAS AND ADJOINT WITHIN THE RBF4AERO PROJECT

The aforementioned optimization methods, either stochastic (EAs) or gradient–based (adjoint) ones, are used for external aerodynamic optimization problems too. Some of these methods were appropriately adapted to fit the needs of the RBF4AERO, EU funded, project http://www.rbf4aero.eu/. The aim of the project is to develop the so–called RBF4AERO Benchmark Technology, an assembly of numerical (CFD, CSD solvers etc), morphing and optimization tools, capable of handling aerodynamic design/optimization problems. The morphing tool used is based on RBF networks and allows for fast morphing of the shapes to be optimized and the surrounding computational mesh, [13]. The optimization tool comprises both EAs and gradient–based methods assisted by the continuous adjoint method.

One of the cases to be studied within RBF4AERO is concerned with the minimization of the drag coefficient of a small aircraft underwing nacelle at two angles of attack, namely 0^o and 8^o. The nacelle is designed for the altitude of 2000m, with $M_\infty = 0.08$ and $Re_c = 3\times10^6$ based on the wing chord. The nacelle rotations around the y and z axes and the nacelle nose scaling were the three design variables used. Starting from a baseline geometry (figure 9 top), for each candidate solution the nacelle shape and the computational mesh was morphed using the customized RBF–based morphing tool of the RBF4AERO platform. The basic incompressible flow solver of OpenFOAM© (*simpleFoam*) was used as the evaluation tool, using the Spalart–Allmaras turbulence model.

In this paper, a (μ,λ)=(10, 30) MAEA was used for the optimization. The MAEA is capable of locating the Pareto front of non–dominated solutions after 100 evaluations on the CFD tool (Figure 9 bottom).

The limited range of the C_D values in the non–dominated front is due to the fact that, in all elite members, the two first design variables do not vary significantly and only the third design variable (i.e. the one related to the nose scaling) varies. This observation can be also backed-up by the magnitude of the drag force sensitivities w.r.t. the three design variables, presented in table 1 and computed using the continuous adjoint method.

b	y rot.	z rot.	scaling
dF/db	-1.9×10^{-4}	-4.4×10^{-7}	5.9×10^{-4}

Table 1. Sensitivity derivatives w.r.t. the three design variables parameterizing the nacelle shape, at 8^o farfield flow angle. It can be observed that the nose scaling has the greatest impact on the drag force value.

5. SUMMARY

This paper presented the use of either stochastic or gradient–based optimization methods in shape optimization of thermal and hydraulic turbomachines. Regarding continuous adjoint methods, some recent advances in the computation of accurate sensitivities were discussed and applied to industrial cases, lead-

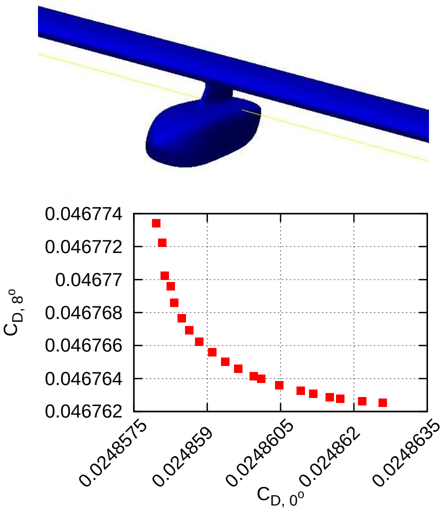

Figure 9. Two–objective optimization of an underwing nacelle (an RBF4AERO test case). Top: Baseline geometry. Bottom: Front of non–dominated solutions resulted after 100 **evaluations on the exact/CFD model.**

ing to the optimization of two Francis runners at a very small CPU cost (20 and 8 optimization cycles for each of the two cases). Regarding evolutionary algorithms, techniques involving surrogate evaluation models, asychronous search on multi-processor platforms and PCA have made the optimization of industrial cases with a great number of design variables and objectives possible.

ACKNOWLEDGEMENTS

Part of this work was funded by the RBF4AERO "Innovative benchmark technology for aircraft engineering design and efficient design phase optimisation" project funded by the EUs 7th Framework Programme (FP7-AAT, 2007-2013) under Grant Agreement no. 605396. The authors would also like to acknowledge contributions from Dr. Stylianos Kyriacou and Christos Kapellos.

REFERENCES

[1] Jameson, A., 1988, “Aerodynamic design via control theory.”, *Journal of Scientific Computing*, Vol. 3, pp. 233–260.

[2] Papoutsis-Kiachagias, E., and Giannakoglou, K., 2014, “Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications”, *Archives of Computational Methods in Engineering*, 10.1007/s11831-014-9141-9.

[3] Karakasis, M., Giotis, A., and Giannakoglou, K., 2003, “Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization”, *International Journal for Numerical Methods in Fluids*, Vol. 43 (10-11), pp. 1149–1166.

[4] Asouti, V., Kampolis, I., and Giannakoglou, K., 2009, “A Grid-Enabled Asynchronous Metamodel-Assisted Evolutionary Algorithm for Aerodynamic Optimization”, *Genetic Programming and Evolvable Machines*, Vol. 10 (3), pp. 373–389.

[5] Kyriacou, S., Asouti, V., and Giannakoglou, K., 2014, “Efficient PCA–driven EAs and metamodel-assisted EAs, with applications in turbomachinery”, *Engineering Optimization*, Vol. 46 (7), pp. 895–911.

[6] Papadimitriou, D., and Giannakoglou, K., 2007, “A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows”, *Computers & Fluids*, Vol. 36 (2), pp. 325–341.

[7] Kampolis, I., Trompoukis, X., Asouti, V., and Giannakoglou, K., 2010, “CFD-based analysis and two-level aerodynamic optimization on Graphics Processing Units”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 199 (9-12), pp. 712–722.

[8] Spalart, P., Jou, W., Stretlets, M., and Allmaras, S., 1997, “Comments on the Feasibility of LES for Wings and on the Hybrid RANS/LES Approach.”, *Proceedings of the first AFOSR International Conference on DNS/LES*.

[9] Zymaris, A., Papadimitriou, D., Giannakoglou, K., and Othmer, C., 2009, “Continuous Adjoint Approach to the Spalart-Allmaras Turbulence Model for Incompressible Flows”, *Computers & Fluids*, Vol. 38 (8), pp. 1528–1538.

[10] Papoutsis-Kiachagias, E., Kyriacou, S., and Giannakoglou, K., 2014, “The Continuous Adjoint Method for the Design of Hydraulic Turbomachines”, *Computer Methods in Applied Mechanics and Engineering*, Vol. 278, pp. 612–639.

[11] Zymaris, A., Papadimitriou, D., Giannakoglou, K., and Othmer, C., 2010, “Adjoint wall functions: A new concept for use in aerodynamic shape optimization”, *Journal of Computational Physics*, Vol. 229 (13), pp. 5228–5245.

[12] Kavvadias, I., Papoutsis-Kiachagias, E., Dimitrakopoulos, G., and Giannakoglou, K., 2014, “The continuous adjoint approach to the k–ω SST turbulence model with applications in shape optimization”, *Engineering Optimization, to appear.*

[13] Biancolini, M. E., “Mesh Morphing and Smoothing by Means of Radial Basis Functions (RBF): A Practical Example Using Fluent and RBF Morph”, *Handbook of Research on Computational Science and Engineering: Theory and Practice (2 vol)*, pp. 347–380.