ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

COMBINING AN RBF-BASED MORPHER WITH CONTINUOUS ADJOINT FOR LOW-SPEED AERONAUTICAL OPTIMIZATION APPLICATIONS

E.M. Papoutsis-Kiachagias[1], M. Andrejašič[2], S. Porziani[3], C. Groth[4], D. Eržen[2], M.E. Biancolini[4], E. Costa[3], and K.C. Giannakoglou[1]

[1] National Technical University of Athens (NTUA), School of Mechanical Engineering,
Parallel CFD & Optimization Unit, Greece,
e-mail: vaggelisp@gmail.com, kgianna@central.ntua.gr
[2] PIPISTREL d.o.o. Ajdovščina, R&D, Department of Aerodynamics, Slovenia,
e-mail: matej.andrejasic@pipistrel.si, david@pipistrel.si
[3] D'Appolonia S.p.A. Italy,
e-mail: stefano.porziani@dappolonia.it, emiliano.costa@dappolonia.it,
[4] University of Rome Tor Vergata (UTV), Italy,
e-mail: Corrado.Groth@uniroma2.it, biancolini@ing.uniroma2.it

Keywords: Continuous Adjoint, RBF Morphing, Aeronautical Optimization

Abstract. *In this paper, the continuous adjoint method, developed by NTUA in the OpenFOAM® environment, is coupled with an RBF-based morpher developed by UTV to tackle optimization problems in low-speed aeronautics. The adjoint method provides a fast and accurate way for computing the sensitivity derivatives of the objective functions (here, drag, lift and losses) with respect to the design variables. The latter are defined as a set of variables controlling a group of RBF control points used to deform both the surface and volume mesh of the computational domain. The use of the RBF-based morpher provides a fast and robust way of handling mesh and geometry deformations, facing two challenging tasks related to shape optimization with the same tool. The coupling of the above-mentioned tools is used to tackle (a) the minimization of the cooling losses for an electric motor installed on a lightweight aircraft, by controlling the cooling air intake shape and (b) the shape optimization of a glider geometry targeting maximum lift-to-drag ratio by mainly optimizing the wing-fuselage junction. Regarding problem (a), a porous media is utilized to simulate the pressure drop caused by the radiator; the adjoint to this porosity model is developed and presented. This work was carried out in the framework of the EU-funded RBF4AERO project and the presented methods are available through the RBF4AERO platform (www.rbf4aero.eu).*

1 INTRODUCTION

During the last years, CFD-based aerodynamic shape optimization has been attracting the interest of both academia and industry. The constituents needed for executing an automated shape optimization loop include the flow solver, the geometry parameterization (the parameters of which act as the design variables), an optimization method capable of computing the optimal values of the design variables and a way to adapt (or regenerate) the computational mesh to each candidate solution.

Nowadays, a great variety of in-house and commercial flow solvers exist. In the study presented in this paper, the steady-state flow solver of the open-source CFD toolbox, OpenFOAM®, is used to numerically solve the Navier-Stokes equations for incompressible, turbulent flows.

Shape parameterization techniques can be divided into two categories, i.e. those parameterizing only the surface to be optimized and those which also deform the surrounding mesh nodes. In the context of an optimization method, the former rely on, among others, the normal displacement of surface wall nodes [14], the displacement of control points of Bézier–Bernstein or NURBS curves or surfaces and the variation in CAD parameters [15, 17]. The latter include volumetric B-splines or NURBS [9], Radial Basis Functions (RBFs) [6, 4], the harmonic coordinates method, etc. The great advantage of this category is that the interior of the computational mesh is also deformed, avoiding, thus, costly re-meshing and allowing the initialization of the flow field from the solution obtained in the previous optimization cycle, since the mesh topology is preserved. In this paper, a number of parameters controlling the positions of groups of RBF control points are used as the design variables, using technology and methods developed in the context of the RBF Morph software [3].

Gradient-based optimization methods require a high effort to develop and maintain but can have a cost per optimization cycle that does not scale with the number of design variables, when the adjoint method is used to compute the gradients of the objective function. Both discrete and continuous adjoint methods, [5, 12], have been developed. In this work, a continuous adjoint method implemented on the NTUA in-house version of the OpenFOAM® software is used.

The above-mentioned tools are combined in order to form an automated optimization loop, targeting the maximization of the lift-to-drag ratio for a glider plane and the cooling losses minimization caused by the cooling system of a small electric airplane.

The work presented in this paper was carried out in the framework of the RBF4AERO Project which aims at developing the RBF4AERO Benchmark Technology, namely a numerical platform conceived to face the requirements of top-level aeronautical design studies such as multi-physics and multi-objective optimization, fluid-structure interaction (FSI), adjoint-driven optimization and ice accretion simulation. Based on the RBF mesh morphing technique, the platform allows to significantly boost the aerodynamic design process and a relevant impact is then expected in the ever-growing technological demand posed by aeronautical manufacturers in relation to the performance and reliability of aircraft components.

2 THE CONTINUOUS ADJOINT METHOD

In this section, the formulation of the continuous adjoint PDEs, their boundary conditions and the sensitivity derivatives (gradient) expression are presented in brief. The interested reader could find more about the adjoint method used in [18, 19, 13]. Since both geometries to be studied operate at low air speeds, the development is based on the incompressible RANS equations.

2.1 Flow equations

The mean flow equations read

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

$$R_i^w = v_j \frac{\partial v_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} + f_i = 0 \tag{1b}$$

where v_i are the velocity components, p is the static pressure divided by the constant density, $\tau_{ij} = (\nu + \nu_t)\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$ are the components of the stress tensor, ν and ν_t the kinematic and turbulent viscosity, respectively. In eq. 1b, term f_i stands for any external force acting on the fluid. In one of the applications studied in section 5, a non-linear, anisotropic porosity model is used based on the Darcy-Forchheimer law, [1]. In this case, f_i reads

$$f_i = r_{ij} v_j \ \ , \ \ r_{ij} = \nu D_{ij} + \sqrt{v_k^2} E_{ij} \tag{2}$$

where D_{ij} and E_{ij} are constant Darcy and Forchheimer tensorial coefficients.

Eqs. 1 along with the turbulence model PDE(s) comprise the primal or state equations. In the applications presented in section 5, the Spalart-Allmaras, [16], and k-ω SST, [10], models are used. Though the continuous adjoint to both turbulence models has been developed by some of the authors, [18, 13, 8], the remainder of the continuous adjoint formulation will neglect their differentiation in the interest of space.

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 and the computational domain Ω is given by

$$F = \int_S F_{S_i} n_i dS + \int_\Omega F_\Omega d\Omega \tag{3}$$

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

Differentiating eq. 3 w.r.t. to b_n and applying the chain rule yields

$$\begin{aligned}\frac{\delta F}{\delta b_n} = &\int_S \left(\frac{\partial F_{S_k}}{\partial v_i} n_k + \acute{F}_{S,i}^v \right) \frac{\partial v_i}{\partial b_n} dS + \int_S \left(\frac{\partial F_{S_i}}{\partial p} n_i + \acute{F}_S^p \right) \frac{\partial p}{\partial b_n} dS + \int_S \frac{\partial F_{S_k}}{\partial \tau_{ij}} n_k \frac{\partial \tau_{ij}}{\partial b_n} dS \\ &+ \int_{S_W} n_i \frac{\partial F_{S_i}}{\partial x_k} \frac{\delta x_k}{\delta b_n} dS + \int_{S_W} F_{S_i} \frac{\delta (n_i dS)}{\delta b_n} + \int_{S_W} F_\Omega n_k \frac{\delta x_k}{\delta b_n} dS + \int_\Omega \acute{F}_{\Omega,i}^v \frac{\partial v_i}{\partial b_n} d\Omega + \int_\Omega \acute{F}_\Omega^p \frac{\partial p}{\partial b_n} d\Omega\end{aligned} \tag{4}$$

where S_W is the parameterized part of the solid wall boundaries and $\acute{F}_\Omega^\Phi$ includes the partial derivative $\partial F_\Omega / \partial \Phi$ plus any term resulting from the use of the Green-Gauss theorem for integrals of the form $\int_\Omega \frac{\partial}{\partial b_n}\left(\frac{\partial \Phi}{\partial x_j}\right) d\Omega$. Terms $\acute{F}_{S,i}^v$ and $\acute{F}_S^p$ are non-zero only if F_Ω includes differential operators of v_i or p. In eq. 4, $\delta \Phi / \delta b_n$ is the total derivative of any quantity Φ while $\partial \Phi / \partial b_n$ is its partial derivative. These are related by

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} \frac{\delta x_k}{\delta b_n} \tag{5}$$

To avoid computing variations in the flow variables, the adjoint method as presented in the next subsection, is used.

2.3 Continuous adjoint formulation

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

$$F_{aug}=F+\int_{\Omega}u_iR^v_i d\Omega+\int_{\Omega}qR^p d\Omega \tag{6}$$

where u_i are the components of the adjoint velocity vector and q is the adjoint pressure. For the development of the adjoint to the turbulence model PDEs, the reader is referred to [13]. The differentiation of eq. 6, based on the Leibniz theorem, yields

$$\frac{\delta F_{aug}}{\delta b_n}=\frac{\delta F}{\delta b_n}+\int_{\Omega}u_i\frac{\partial R^v_i}{\partial b_n}d\Omega+\int_{\Omega}q\frac{\partial R^p}{\partial b_n}d\Omega+\int_{S_W}(u_iR^v_i+qR^p)n_k\frac{\delta x_k}{\delta b_n}dS \tag{7}$$

Then, the derivatives of the flow residuals in the volume integrals on the r.h.s. of eq. 7 are developed by differentiating eqs. 1 and applying the Green-Gauss theorem, where necessary. This development can be found in [18, 8, 13].

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. 7 are set to zero, giving rise to the field adjoint equations

$$R^q=-\frac{\partial u_j}{\partial x_j}+\acute{F}^p_{\Omega}=0 \tag{8a}$$

$$R^v_i=u_j\frac{\partial v_j}{\partial x_i}-\frac{\partial (v_ju_i)}{\partial x_j}-\frac{\partial \tau^a_{ij}}{\partial x_j}+\frac{\partial q}{\partial x_i}+\acute{F}^v_{\Omega,i}+f^a_i=0 \tag{8b}$$

where $\tau^a_{ij}=(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)$ are the components of the adjoint stress tensor and f^a_i is the adjoint to the external force term. The adjoint to the Darcy-Forchheimer porosity force term reads

$$f^a_i=u_k(r_{ki}+E_{kj}v_jv_i) \tag{9}$$

The adjoint boundary conditions are derived by treating the flow variations in the boundary integrals (of the developed form of) eq. 7. This development is presented in detail in [13].

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 problems governed by the RANS turbulence models with wall functions was initially presented in [19], where the adjoint wall function technique was introduced for the $k-\varepsilon$ model and a vertex–centered finite volume method with slip velocity at the wall. 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 also been implemented in flow solvers based on cell-centered finite-volume schemes, for the Spalart–Allmaras , [18], and $k-\omega$ SST, [8], models.

After satisfying the adjoint PDEs and their boundary conditions, the remaining terms in eq. 7

yield the sensitivity derivatives

$$\begin{aligned}
\frac{\delta F_{aug}}{\delta b_n} = & -\int_{S_W}\left[(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)n_j - qn_i + \frac{\partial F_{S_{W,l}}}{\partial v_i}n_l + \acute{F}^{v}_{S_{W,i}}\right]\frac{\partial v_i}{\partial x_k}\frac{\delta x_k}{\delta b_n}dS \\
& + \int_{S_W} n_i \frac{\partial F_{S_{W,i}}}{\partial x_k}\frac{\delta x_k}{\delta b_n}dS + \int_{S_W} F_{S_{W,i}}\frac{\delta(n_i dS)}{\delta b_n}dS + \int_{S_W}(u_i R^v_i + qR^p + F_\Omega)\frac{\delta x_k}{\delta b_n}n_k dS \\
& - \int_{S_W}\left[\left(-u_{\langle n\rangle} + \frac{\partial F_{S_{W,k}}}{\partial \tau_{lm}}n_k n_l n_m\right)\left(\tau_{ij}\frac{\delta(n_i n_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m}n_m\frac{\delta x_k}{\delta b_n}n_k n_i n_j\right)\right]dS \\
& - \int_{S_W}\left[\frac{\partial F_{S_{W,k}}}{\partial \tau_{lm}}n_k t^I_l t^I_m\left(\tau_{ij}\frac{\delta(t^I_i t^I_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m}n_m\frac{\delta x_k}{\delta b_n}n_k t^I_i t^I_j\right)\right]dS \\
& - \int_{S_W}\left[\left(\frac{\partial F_{S_{W,k}}}{\partial \tau_{lm}}n_k (t^{II}_l t^I_m + t^I_l t^{II}_m)\right)\left(\tau_{ij}\frac{\delta(t^{II}_i t^I_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m}n_m\frac{\delta x_k}{\delta b_n}n_k t^{II}_i t^I_j\right)\right]dS \\
& - \int_{S_W}\left[\frac{\partial F_{S_{W,k}}}{\partial \tau_{lm}}n_k t^{II}_l t^{II}_m\left(\tau_{ij}\frac{\delta(t^{II}_i t^{II}_j)}{\delta b_n} + \frac{\partial \tau_{ij}}{\partial x_m}n_m\frac{\delta x_k}{\delta b_n}n_k t^{II}_i t^{II}_j\right)\right]dS \qquad (10)
\end{aligned}$$

New symbols appearing in eq. 10 are explained in [13]. The deformation velocities, $\delta x_k/\delta b_n$, included in eq. 10 express the dependency of the boundary wall nodes on the shape modification parameters. This can be computed by differentiating the surface parameterization scheme presented in the next section.

3 RBF-BASED MORPHING

In this section the mesh morphing algorithm based on RBFs is described. The background theory of RBFs and details of its application in the mesh morphing field are presented; the industrial implementation of the method, as provided by the stand alone version of the software RBF Morph, is then described; finally, the coupling of the mesh morphing tool with adjoint-based sensitivities is explained.

3.1 RBFs background

RBFs are mathematical functions able to interpolate data defined at discrete points only (source points) in an n-dimensional environment. The interpolation quality and its behavior depends on the chosen radial basis function.

In general, the solution of the RBF mathematical problem consists on the computation of the scalar parameters of a linear system of order equal to the number of considered source points. The RBF system solution, determined after defining a set of source points with their displacement, is employed to operate mesh morphing to the discretized domain of the computational model. Operatively, once the RBF system coefficients have been computed, the displacement of an arbitrary node of the mesh, either inside (interpolation) or outside (extrapolation) the domain, can be expressed as the sum of the radial contribution of each source point (if the point falls inside the influence domain). In such a way, a desired modification of the mesh nodes position (smoothing) can be rapidly applied preserving mesh topology.

RBFs can be classified on the basis of the type of support (global or compact) they have, meaning the domain where the chosen RBF is non zero-valued.

An interpolation function composed of a radial basis function φ and a polynomial h of order $m-1$, where m is the order of φ, introduced with the aim to guarantee the compatibility with rigid motions, is defined as follows

$$s(x) = \sum_{i=1}^{N} \gamma_i \varphi\left(\|x - x_{k_i}\|\right) + h(x) \tag{11}$$

where N is the total number of contributing source points. The degree of the polynomial has to be chosen depending on the kind of RBF adopted. A radial basis fit exists if the coefficients γ_i and the weights of the polynomial can be found such that the desired function values are obtained at source points and the polynomial terms gives no contributions at source points, i.e.

$$s(x_{k_i}) = g_i, 1 \le i \le N \ \ , \ \ \sum_{i=1}^{N} \gamma_i q(x_{k_i}) = 0 \tag{12}$$

for all polynomials q with a degree less or equal to that of polynomial h. The minimal degree of polynomial h depends on the choice of the RBF. A unique interpolant exists if the basis function is a conditionally positive definite function [11]. If the RBFs are conditionally positive definite of order $m \le 2$ [2], a linear polynomial can be used

$$h(x) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z \tag{13}$$

The subsequent development will assume that the aforementioned hypothesis is valid. The values for the coefficients γ_i of RBF and the coefficients β of the linear polynomial can be obtained by solving the system

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P^T} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \tag{14}$$

where g are the known values at the source points and $\mathbf{M}$ is the interpolation matrix defined by calculating all the radial interactions between source points

$$M_{ij} = \varphi\left(\left\|x_{k_i} - x_{k_j}\right\|\right), 1 \le i \le N, 1 \le j \le N \tag{15}$$

P is a constraint matrix that arises to balance the polynomial contribution and contains a column of ”1” and the x, y, z positions of the source points in the other three columns

$$\mathbf{P} = \begin{pmatrix} 1 & x_{k_1} & y_{k_1} & z_{k_1} \\ 1 & x_{k_2} & y_{k_2} & z_{k_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N} & y_{k_N} & z_{k_N} \end{pmatrix} \tag{16}$$

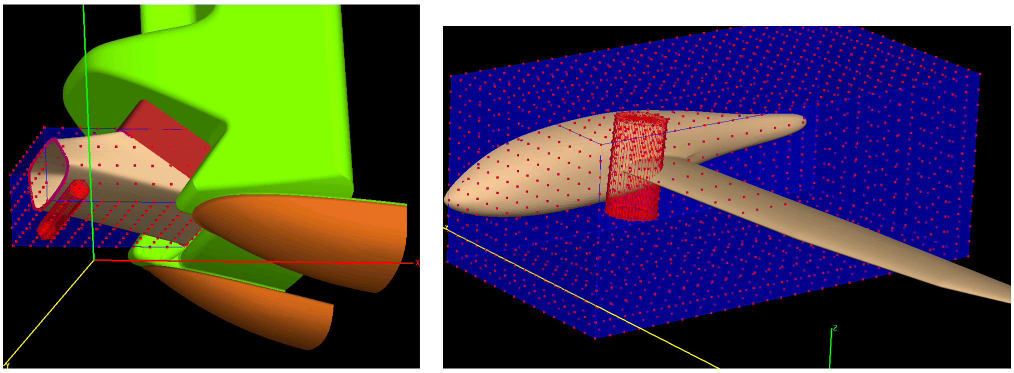
RBF interpolation works for scalar fields, hence a system of the form of eq. 14 has to be solved for each of the three spatial directions.

The RBF method has several advantages that make it very attractive for mesh smoothing. The key point is that being a meshless method only grid points are moved regardless of which elements are connected to them; this make the method suitable for parallel implementation. In fact, once the solution is known and shared in the memory of each processing node of the cluster, each partition has the ability to smooth its nodes without taking care of what happens outside, because the smoother is a global point function and the continuity at interfaces is implicitly guaranteed. Though meshless, the method is able to exactly prescribe known deformations onto the surface mesh: this effect is achieved by using all the mesh nodes as RBF centres with prescribed displacements, including the simple zero field to guarantee that a surface is left untouched by the morphing action.

3.2 RBF Morph tool

The industrial implementation of the RBF mesh morphing poses two challenges: the numerical complexity related to the solution of the RBF problem for a large number of centers and the definition of suitable paradigms to effectively control shapes using RBF. The software RBF Morph allows to deal with both as it comes with a fast RBF solver capable to fit large datasets (hundreds of thousands of RBF points can be fitted in a few minutes) and with a suite of modeling tools that allows the user to set-up each shape modification in an expressive an flexible way.

RBF Morph allows to extract and control points from surfaces and edges, to put points on primitive shapes (boxes, spheres and cylinders) or to specify them directly by individual coordinates and displacements. Primitive shapes can be combined in a Boolean fashion and allow to limit the action of the morpher itself. Two shape modifications used in this study are represented in fig. 1. It is worth noticing that the shape information coming from an individual RBF set-up are generated interactively with the help of the GUI and are used subsequently in batch commands that allows to combine many shape modifications in a non linear fashion (non linearity occurs when rotation axes are present in the RBF set-up).



(a) cooling system diffuser

(b) wing-fuselage junction

Figure 1: Example of RBF points arrangement for the definition of two shape parameters, (a) the lower wall of the cooling system diffuser is parameterized using a cluster of RBF control points forming a cylinder. A Box Encapsulation is used to limit the effect of the movement in the vicinity of the diffuser, also making sure that the edge forming the diffuser inlet will not be deformed, since it is defined by the fuselage shape which needs to remain intact, (b) a similar setup is used to define the deformation of the wing-fuselage junction close to the leading edge.

3.3 Coupling of RBF mesh morphing with adjoint sensitivities

Once the adjoint-based sensitivities are available, it is possible to easily compute the sensitivities w.r.t. shape parameters exploiting the parametric mesh available using the mesh morphing tool. In order to take into account the non-linear fashion of the morphing field, the mesh deformation velocities are generated by numerical differentiation of the morphing field around the current design point in the parametric space. For a given set of shape parameters, the morpher is capable to update the baseline mesh into the current one. A perturbed mesh, w.r.t. the current one, can then be obtained for each shape parameter, computing the mesh resulting from its perturbation while keeping all the other constant. The sensitivity w.r.t. each given parameter is then obtained by multiplying the surface perturbation field by the surface sensitivities, eq. 10. It is worth noting that the aforementioned coupling works not just at the origin of the parametric space (baseline model) but at any given design point; adjoint data need to be recomputed for each design point for which local sensitivities are required.

4 OPTIMIZATION ALGORITHM

The gradient-based algorithm used to minimize the objective function is described in brief below:

1. Define the shape modification parameters, section 3.
2. Solve the flow equations, eqs. 1.
3. Compute F.
4. Solve the adjoint equations, eqs. 8.
5. Compute the deformation velocities and through them, the sensitivity derivatives, eq. 10.
6. Update the design variables by using a descent method.
7. Morph the parameterized surface and displace the interior mesh nodes.
8. Unless the stopping criterion is met, go to step 2.

5 APPLICATIONS

5.1 Cooling losses minimization for an electric aircraft

The first application of the automated optimization loop presented in section 4 deals with the minimization of the fluid power dissipation within the cooling system of the electric engine mounted on a lightweight aircraft. The RANS equations are solved around a simplified fuselage geometry which includes the cooling system configuration, fig. 2; a hex-dominated mesh consisting of about 2.1 million cells is utilized. The flow Reynolds number is $Re = 1.22 \times 10^5$ based on the hydraulic diameter of the cooling system inlet and the k-ω SST turbulence model is used. To avoid simulating the flow within the heat exchanger, its impact on the flow is modeled through the addition of a non-isotropic porosity term, eqs. 1b and 2. The objective function to be minimized reads

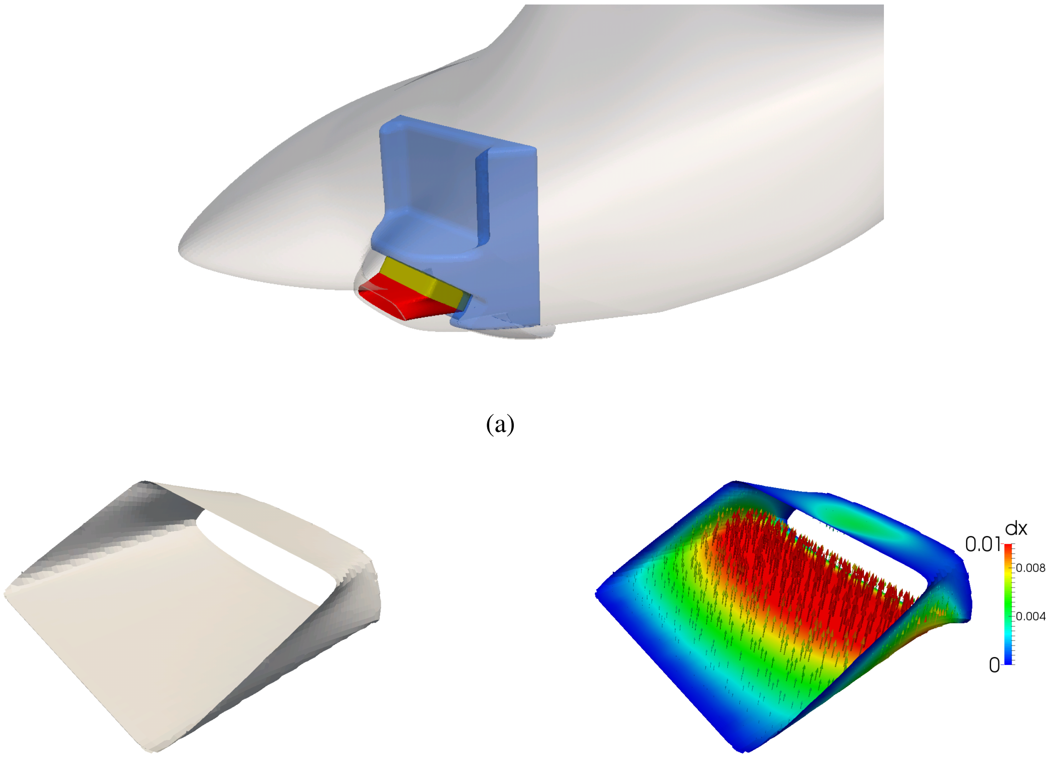
$$F = \frac{F_1}{F_2}, \quad F_1 = \int_{\Omega_c} \left[\frac{(\nu + \nu_t)}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right)^2 + r_{ij} v_i v_j \right] d\Omega, \quad F_2 = \int_{S_{I_C}} v_i n_i dS \tag{17}$$

In eq. 17, F_1 expresses the fluid power dissipation within the cooling system domain Ω_c, fig. 2, in the presence of an anisotropic porous medium. Since the flow rate through the cooling system is not fixed, the denominator F_2, quantifying the volume flow rate through the cooling system inlet S_{I_C}, is included in F in order to prevent the optimization algorithm from reducing the power dissipation by dramatically reducing the flow through the radiator. The latter is important since the cooling capacity directly depends on the flow mass passing through the cooling configuration.

Since S_{I_C} is not a mesh boundary, computing the appropriate contributions from the differentiation of F_2 to the adjoint flow is not a trivial task. In order to facilitate the imposition of the adjoint boundary conditions, a couple of coinciding boundaries are defined at S_{I_C}, abbreviated as $S^L_{I_C}$ and $S^R_{I_C}$; flow variables are equated in the corresponding faces of $S^L_{I_C}$ and $S^R_{I_C}$. Due to the applied boundary conditions, $S_{I_C}^L$ and $S_{I_C}^R$ practically act as internal mesh faces during the solution of the flow equations. Then, for the solution of the adjoint equations, F_2 is defined along only one of these boundaries, say $S_{I_C}^L$. Developing the adjoint to the aforementioned objective function and taking into consideration the primal boundary conditions, the following adjoint boundary conditions are derived along $S_{I_C}^L$ and $S_{I_C}^R$

$$u_i^L = u_i^R \quad , \quad q^L = q^R + \frac{F_1}{F_2^2} \tag{18}$$

Only the shape of the diffuser of the cooling system, located directly downstream of its inlet, fig. 2(a) is allowed to vary in this optimization study. Eight RBF-based design variables are defined, controlling the upper, lower and side walls of the diffuser, fig. 3. The CG method is used to update the design variables and a 5% reduction in F is obtained within 4 optimization cycles, caused by an 8% reduction in fluid power dissipation and a slight reduction of the volume flow rate by 3%. The optimized diffuser geometry along with the cumulative surface displacement is depicted in fig. 2(c). In fig. 4, the flow streamlines inside the initial and optimized cooling system geometries are presented. The reduction in the objective value can be attributed to the fact that the flow recirculation present close to the upper diffuser wall has practically vanished in the optimized geometry.

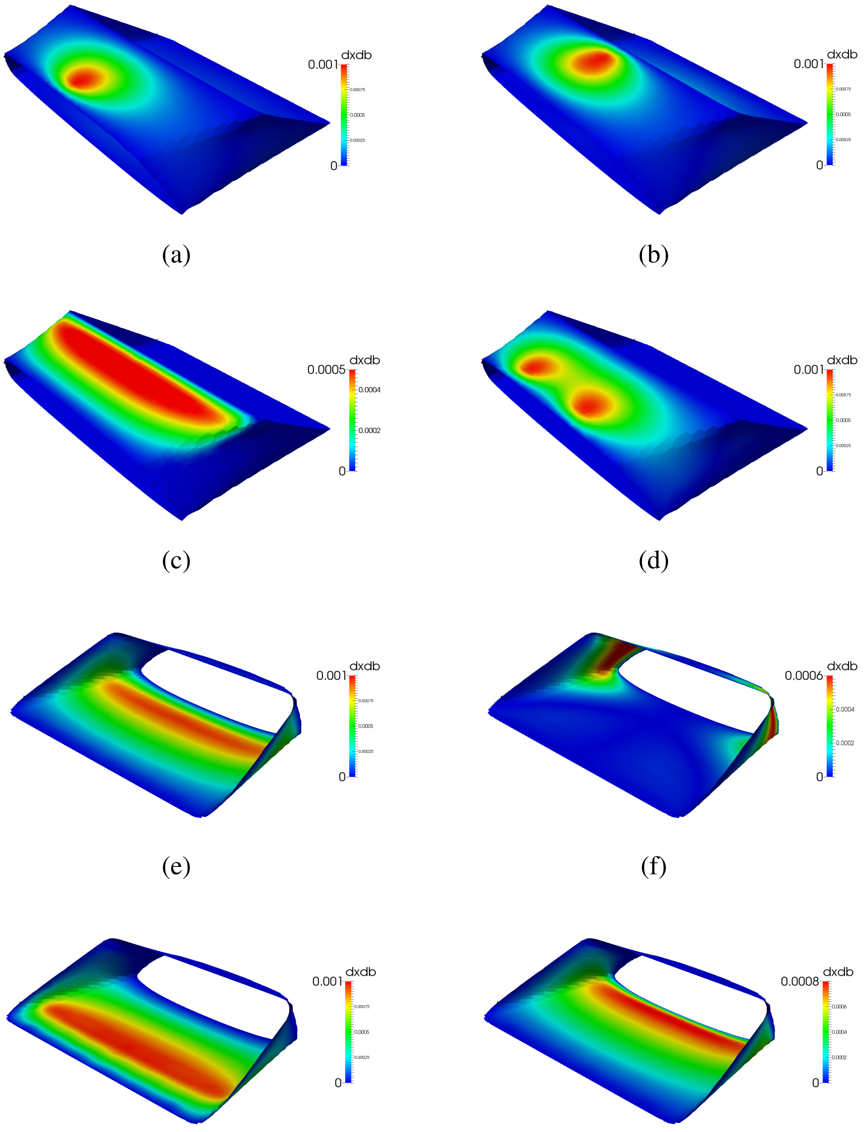


(a)

(b)

(c)

Figure 2: Cooling system optimization: (a) the cooling system and its position in the airplane geometry. The diffuser is coloured in red and is the only part of the geometry that is allowed to vary during the optimization. The yellow area is where the actual cooling takes place and is modeled by adding an anisotropic porosity term in the momentum equations. The combination of the red, yellow and blue parts comprises Ω_c, (b) the initial diffuser geometry (flow from right to left) and (c) the optimized diffuser geometry, colored based on the cumulative (after four optimization cycles) surface displacement; the maximum displacement is located at the lower diffuser wall.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

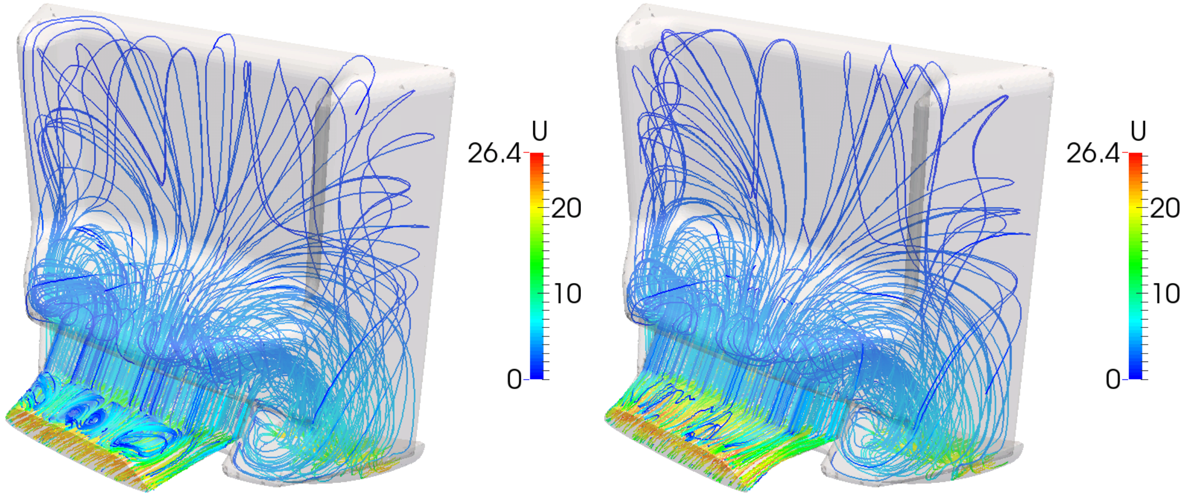
Figure 3: Cooling system optimization: deformation velocity ($\delta x_k/\delta b_m$) for the eight design variables parameterizing the diffuser shape. Design variables (a)-(d) parameterize the upper diffuser wall while those shown in (e)-(h) the side and lower walls.

5.2 Lift-to-drag ratio maximization for a glider plane

This section is concerned with the shape optimization of a glider plane targeting the maximization of the lift-to-drag ratio or, equivalently, the minimization of

$$F = -\frac{\int_{S_W} \left(-\tau_{ij} n_j + p n_i\right) r_i^L dS}{\int_{S_W} \left(-\tau_{ij} n_j + p n_i\right) r_i^D dS} \qquad (19)$$

where $\mathbf{r^L}$ and $\mathbf{r^D}$ are the lift and drag force projection (unit) directions, respectively. The flow Reynolds number is $Re = 1.55 \times 10^6$ based on the wing chord, the Spalart–Allmaras turbulence model is used, the mesh consists of about 4.7 million cells and the far-field flow angle is 10^o. The geometry is parameterized using four RBF-based design variables depicted in fig. 5, controlling the wing-fuselage junction close to the leading and trailing edges as well as parts of the



(a) initial, $||v||$ (b) optimized, $||v||$

Figure 4: Cooling system optimization: streamlines plotted through the initial (left) and optimized (right) geometries, coloured based on the flow velocity. The intense flow recirculation present close to the upper diffuser wall has practically disappeared.

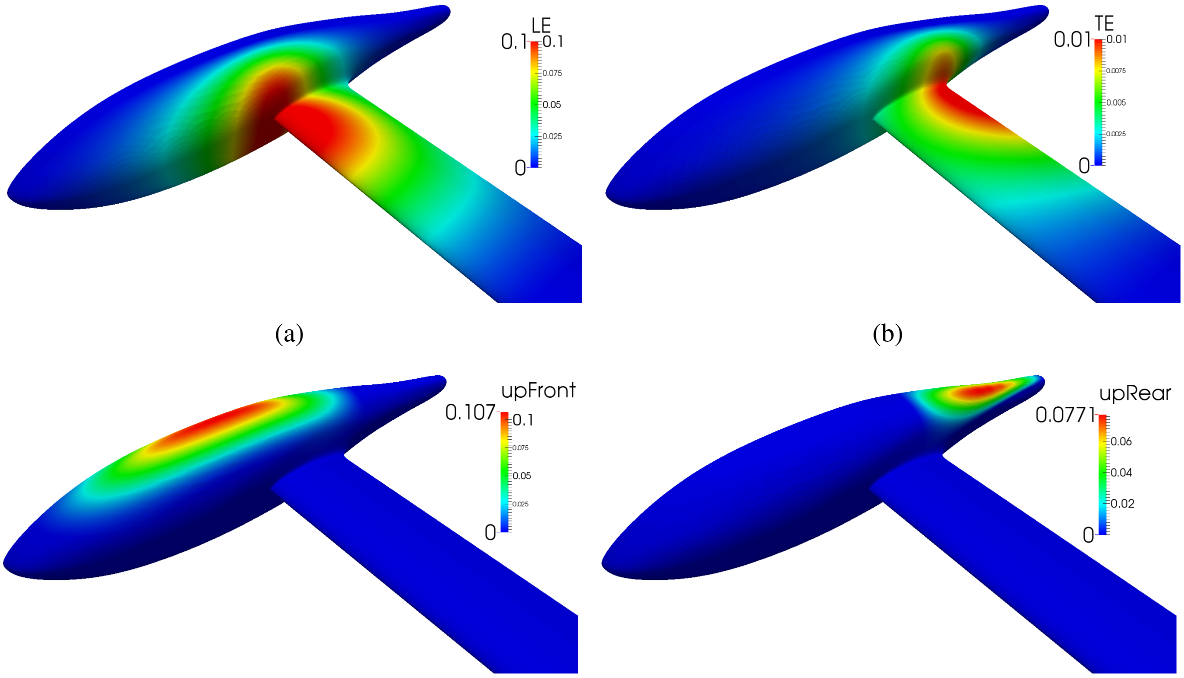
upper fuselage surface. The convergence of the steepest descent-driven algorithm presented in section 4 is showcased in fig. 6(a). It can be observed that the lift-to-drag ratio has increased by 15%, caused by 10% drag reduction and a 4% lift increase. The optimized geometry is illustrated in fig. 6(b). In fig. 7, the near-wall velocity isolines are plotted on the glider surface for the initial and optimized geometries. It can be observed that the flow recirculation formed close to the trailing edge-fuselage junction has been significantly reduced.

6 CONCLUSIONS

The continuous adjoint method and an RBF-based morpher, combined into an automated optimization software were used to build a gradient-based optimization algorithm, applied to two low-speed aeronautical applications; the first targeted the losses minimization within the cooling system of a small electric airplane while the second one the increase of the lift-to-drag ratio for a glider plane. The first application required the differentiation of a non-linear, anisotropic porosity model, used to simulate the flow resistance within the radiator of the cooling system. In both cases, the optimized solution was achieved within very few optimization cycles, leading to a small optimization turnaround time. The coupling of the adjoint-based optimization software and the RBF-based morpher is part of the RBF4AERO platform. In a companion paper, [7], the part of RBFAERO platform which combines the RBF-based morpher with an Evolutionary Algorithm-based optimization strategy is also presented.

ACKNOWLEDGMENT

This work was funded by the RBF4AERO ”Innovative benchmark technology for aircraft engineering design and efficient design phase optimisation” project funded in the Aeronautics and Air Transport (AAT) research thematic area of the EUs 7th Framework Programme (FP7-AAT, 2007-2013) under Grant Agreement no. 605396.



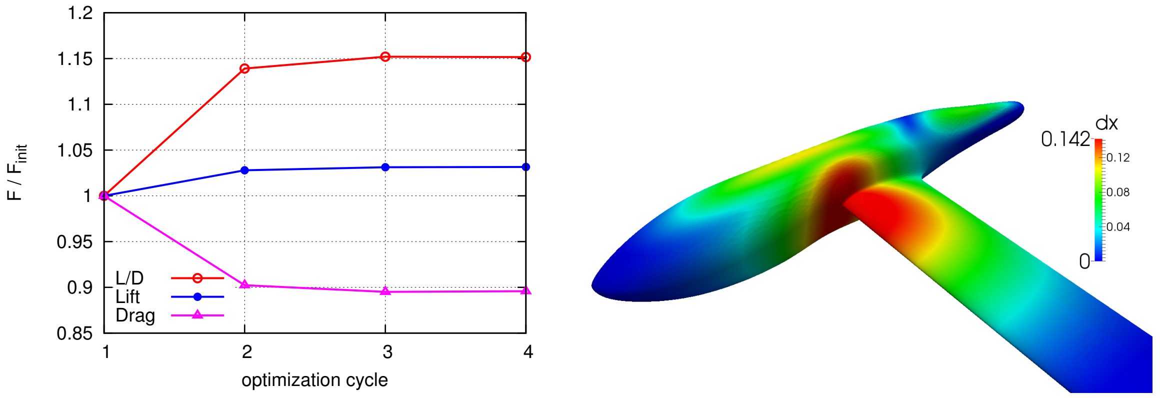
(a)

(b)

(c)

(d)

Figure 5: Glider shape optimization: the magnitude of the parametric velocity ($\delta x_k/\delta b_m$) for the four design variables parameterizing the glider shape. The first two parameterize the wing-fuselage junction close to the leading and trailing edges, while the second two affect parts of the upper glider surface. All design variables are allowed to vary within certain limits in order to prevent the generation of non-manufacturable solutions.



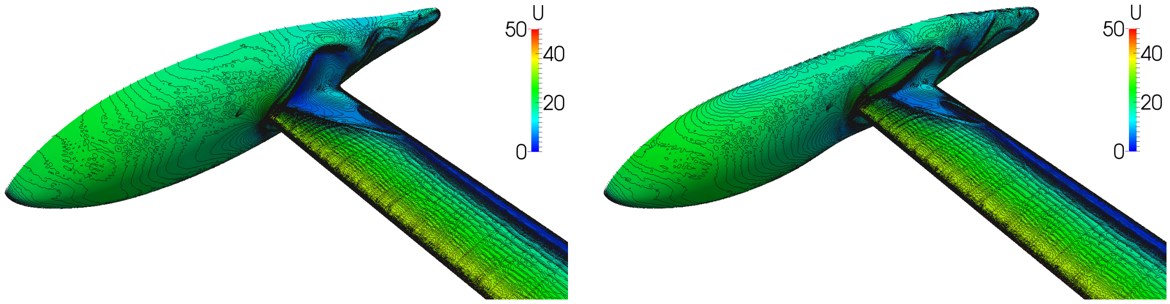
(a)

(b)

Figure 6: Glider shape optimization: (a) convergence of the lift-to-drag ratio (L/D), along with the lift and drag values. All values plotted w.r.t. the ones obtained using the initial geometry. A 15% lift-to-drag increase is observed in 4 optimization cycles by mainly reducing the drag value and slightly increasing lift, (b) the optimized glider geometry, coloured based on the cumulative surface displacement. A maximum displacement of 14.2 *cm* is observed close to the leading-edge and fuselage junction.

REFERENCES

[1] JS. Andrade, UMS. Costa, MP. Almeida, HA. Makse, and HE. Stanley. Inertial effects on fluid flow through disordered porous media. *Physical Review Letters*, 82(26):5249–5252, 1999.



(a)

(b)

Figure 7: Glider shape optimization: near wall velocity isolines, plotted on the glider surface for the (a) initial and (b) optimized geometries. It can be observed that the low velocity area close to the trailing edge has been considerably reduced.

[2] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Constructive Approximation*, 5(2):125–134, 2011.

[3] ME. Biancolini. Mesh morphing and smoothing by means of radial basis functions (RBF): A practical example using Fluent and RBF Morph. In *Handbook of Research on Computational Science and Engineering: Theory and Practice (2 vol)*, pages 347–380, 2011.

[4] ME. Biancolini, IM. Viola, and M. Riotte. Sails trim optimisation using CFD and RBF mesh morphing. *Computers & Fluids*, 93:46–60, 2014.

[5] M. Giles and N. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

[6] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36:1119–1136, 2007.

[7] DH. Kapsoulis, VG. Asouti, KC. Giannakoglou, E. Costa S. Porziani, C. Groth, U. Cella, and ME. Biancolini. Evolutionary aerodynamic shape optimization through the RBF4AERO platform. In *ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering*, Greece, 5-10 June 2016.

[8] IS. Kavvadias, EM. Papoutsis-Kiachagias, G. Dimitrakopoulos, and KC. Giannakoglou. The continuous adjoint approach to the k–ω SST turbulence model with applications in shape optimization. *Engineering Optimization*, 47(11):1523–1542, 2015.

[9] MJ. Martin, E. Andres, C. Lozano, and E. Valero. Volumetric B-splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, 37:26–36, 2014.

[10] FR. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. *Turbulence, Heat and Mass Transfer*, 4:625–632, 2003.

[11] C. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constructive Approximation*, 2(1):11–22, 1986.

[12] S. Nadarajah and A. Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. In *AIAA Paper 2001-2530, 15th Computational Fluid Dynamics Conference*, Anaheim, CA, 2001.

[13] EM. Papoutsis-Kiachagias and KC. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering, DOI 10.1007/s11831-014-9141-9*, 2014.

[14] EM. Papoutsis-Kiachagias, SA. Kyriacou, and KC. Giannakoglou. The continuous adjoint method for the design of hydraulic turbomachines. *Computer Methods in Applied Mechanics and Engineering*, 278:621–639, 2014.

[15] TT. Robinson, CG. Armstrong, HS. Chua, C. Othmer, and T. Grahs. Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Computer-Aided Design & Applications*, 9(3):253–268, 2012.

[16] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *AIAA Paper 1992-0439, 30th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 6-9 January 1992.

[17] PM. Thompson, TT. Robinson, and CG. Armstrong. Efficient CAD-based aerodynamic design optimization with adjoint CFD data. In *21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*, June 2013.

[18] AS. Zymaris, DI. Papadimitriou, KC. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.

[19] AS. Zymaris, DI. Papadimitriou, KC. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245, 2010.