Aerodynamic Optimization of Car Shapes using the Continuous Adjoint Method and an RBF Morpher

E.M. Papoutsis-Kiachagias, S. Porziani, C. Groth, M.E. Biancolini, E. Costa and K.C. Giannakoglou

Abstract This paper presents the application of the continuous adjoint method, programmed in OpenFOAM©, combined with an RBF-based morpher to the aerodynamic optimization of a generic car model. The continuous adjoint method produces accurate sensitivities by utilizing the full differentiation of the Spalart–Allmaras turbulence model, based on wall functions, while the RBF-based morpher provides a fast and versatile way to deform both the surface of the car and the interior mesh nodes. The integrated software is used to minimize the drag force exerted on the surface of the DrivAer car model.

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 and are in widespread use. In the study presented in this paper, the steady-state flow solver

E.M. Papoutsis-Kiachagias, K.C. Giannakoglou
National Technical University of Athens, Parallel CFD & Optimization Unit, Greece, e-mail: vaggelisp@gmail.com, kgianna@central.ntua.gr

S. Porziani, E. Costa
D'Appolonia S.p.A., Viale Cesare Pavese, 305 - 00144 Rome, Italy, e-mail: stefano.porziani@dappolonia.it, emiliano.costa@dappolonia.it

C. Groth, M.E. Biancolini
University of Rome Tor Vergata (UTV), Italy, e-mail: corrado.groth@uniroma2.it, biancolini@ing.uniroma2.it

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 nodes of the interior mesh. The former include the normal displacement of surface wall nodes [14], the control points of Bézier–Bernstein or NURBS curves or surfaces and CAD parameters [15, 17]. The latter include volumetric B-splines or NURBS [10], Radial Basis Functions (RBFs) [6, 2], the harmonic coordinates method [7], etc. The great advantage of the 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].

Optimization methods can be separated into two main categories, i.e. stochastic and gradient-based ones. Stochastic optimization methods, with Evolutionary Algorithms (EAs) as their main representative, are extremely versatile and have the ability to compute global optima but suffer from a computational cost that scales with the number of design variables, making their use impractical for large scale optimization problems. On the other hand, gradient-based optimization methods require a higher 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, [4, 12], have been developed. In this work, a continuous adjoint method that takes into consideration the differentiation of the turbulence model PDE is used to increase the accuracy of the computed sensitivities of the drag force objective function w.r.t. the shape modification parameters [13]. The continuous adjoint solver has been implemented on an in-house version of the OpenFOAM© software.

The above-mentioned tools are combined in order to form an automated optimization loop, targeting the minimization of the drag force exerted on the surface of a generic car model. In specific, a configuration of the DrivAer car model, [5], developed by the Institute of Aerodynamics and Fluid Mechanics of TU Munich, is studied. The constituents of the optimization loop were combined under the RBF4AERO project. Funded in the Aeronautics and Air Transport (AAT) research thematic area of the EU Seventh Framework Programme, the RBF4AERO Project 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, such a numerical 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 aircrafts constituting components. To demonstrate

the general validity and the effective usage of the RBF4AERO platform in the industrial field, one of its capabilities envisaging the adjoint-morphing coupling is, herein, described for a car aerodynamics optimization problem.

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 development is based on the incompressible Navier-Stokes equations.

2.1 Flow Equations

The mean flow equations together with the Spalart–Allmaras turbulence model PDE, [16], comprise the flow or primal system of equations that reads

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

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

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

where v_i are the components of the velocity vector, 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, $\widetilde{\nu}$ the Spalart–Allmaras model variable and Δ the distance from the wall boundaries. Details about the turbulence model constants, source terms and boundary conditions can be found in [16].

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 **n** is the outward facing normal unit vector.

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

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

where $\delta \Phi / \delta b_n$ is the total derivative of any quantity Φ while $\partial \Phi / \partial b_n$ is its partial derivative. These are related by

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

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. 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 objective function

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

where u_i are the components of the adjoint 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. This assumption leads to reduced gradient accuracy, possibly even to wrong sensitivity signs, [19]. To avoid making the "frozen turbulence" assumption, the Spalart–Allmaras model PDE has been differentiated, see [19]. A review on continuous adjoint methods for turbulent flows can be found in [13].

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

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

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. This development can be found in [19] and [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. 6 are set to zero, giving rise to the field adjoint equations

$$R^q = -\frac{\partial u_j}{\partial x_j} = 0 \tag{7a}$$

$$R_i^v = u_j \frac{\partial v_j}{\partial x_i} - \frac{\partial (v_j u_i)}{\partial x_j} - \frac{\partial \tau_{ij}^a}{\partial x_j} + \frac{\partial q}{\partial x_i} + \widetilde{\nu}_a \frac{\partial \widetilde{\nu}}{\partial x_i} - \frac{\partial}{\partial x_l}\left(\widetilde{\nu}_a \widetilde{\nu} \frac{\mathscr{C}_Y}{Y} e_{mjk} \frac{\partial v_k}{\partial x_j} e_{mli}\right) = 0 \tag{7b}$$

$$R^{\widetilde{\nu}_a} = -\frac{\partial (v_j \widetilde{\nu}_a)}{\partial x_j} - \frac{\partial}{\partial x_j}\left[\left(\nu + \frac{\widetilde{\nu}}{\sigma}\right)\frac{\partial \widetilde{\nu}_a}{\partial x_j}\right] + \frac{1}{\sigma}\frac{\partial \widetilde{\nu}_a}{\partial x_j}\frac{\partial \widetilde{\nu}}{\partial x_j} + 2\frac{c_{b2}}{\sigma}\frac{\partial}{\partial x_j}\left(\widetilde{\nu}_a \frac{\partial \widetilde{\nu}}{\partial x_j}\right) + \widetilde{\nu}_a \widetilde{\nu} \mathscr{C}_{\widetilde{\nu}} + \frac{\partial \nu_t}{\partial \widetilde{\nu}}\frac{\partial u_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + (-P + D)\,\widetilde{\nu}_a = 0 \tag{7c}$$

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 the adjoint stress tensor. Eq. 7c 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. This development is presented in detail in [19, 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 optimization problems, governed by the RANS turbulence models with wall functions, was initially presented in [20], 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 for flow solvers based on cell-centered finite-volume schemes, for the Spalart–Allmaras , [13], and $k-\omega$ *SST*, [8], turbulence models.

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

$$\frac{\delta F_{aug}}{\delta b_n} = T_{SD}^{WF} - \int_{S_{W_p}} SD_1 \frac{\partial \tau_{ij}}{\partial x_m} n_j t_i^I n_m n_k \frac{\delta x_k}{\delta b_n} dS - \int_{S_{W_p}} SD_1 \tau_{ij} \frac{\delta (n_j t_i^I)}{\delta b_n} dS$$
$$+ \int_{S_{W_p}} SD_{2,i} v_{\langle t \rangle}^I \frac{\delta t_i^I}{\delta b_n} dS - \int_{S_{W_p}} SD_{2,i} \frac{\partial v_i}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS$$
$$- \int_{S_{W_p}} \left[\left(\nu + \frac{\widetilde{\nu}}{\sigma} \right) \frac{\partial \widetilde{\nu}_a}{\partial x_j} n_j + \frac{\partial F_{S_z}}{\partial \widetilde{\nu}} n_z \right] \frac{\partial \widetilde{\nu}}{\partial x_m} n_m n_k \frac{\delta x_k}{\delta b_n} dS$$
$$- \int_{S_{W_p}} \left(-u_{\langle n \rangle} + \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k n_l n_m \right) TS_1 dS - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^I t_m^I TS_2 dS$$
$$- \int_{S_{W_p}} \left(\frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k (t_l^{II} t_m^I + t_l^I t_m^{II}) \right) TS_3 dS - \int_{S_{W_p}} \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{lm}} n_k t_l^{II} t_m^{II} TS_4 dS$$
$$+ \int_{S_{W_p}} (u_i R_i^v + q R^p + \widetilde{\nu}_a R^{\widetilde{\nu}}) \frac{\delta x_k}{\delta b_n} n_k dS \tag{8}$$

where

$$SD_1 = -u_{\langle t \rangle}^I + \phi_{ij} t_i^I n_j + \phi_{ij} n_i t_j^I \ , \ SD_{2,i} = \tau_{a.ij} n_j - q n_i + \frac{\partial F_{S_{W_p,k}}}{\partial v_i} n_k \ , \ \phi_{ij} \quad = \frac{\partial F_{S_{W_p,k}}}{\partial \tau_{ij}} n_k$$

Functions TS_1 to TS_4 can be found in [13] while term T_{SD}^{WF} results from the differentiation of the law of the wall.

The deformation velocities, $\delta x_k / \delta b_n$, included in eq. 8 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 is first introduced providing details of its application in mesh morphing field; 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 and the adjoint sensitivity is explained.

3.1 RBFs Background

RBFs are powerful mathematical functions able to interpolate data defined at discrete points only (source points) in a *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 calculation of the scalar parameters (sought coefficients) 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 calculated, 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.

Typical RBFs with global and compact support are shown in Table 1. RBFs are scalar functions with the scalar variable r, which is the Euclidean norm of the distance between two points defined in a generic n-dimensional space.

Radial Basis Functions(RBF) with global support	$\varphi(r), r = \|r\|$
Spline type (R_n)	$r^n, nodd$
Radial Basis Functions(RBF) with compact support	$\varphi(r) = f(\xi), \xi \leq 1, \xi = \frac{r}{R_{sup}}$
Wendland C^0 (C0)	$(1-\xi)^2$
Wendland C^2 (C2)	$(1-\xi)^4(4\xi+1)$
Wendland C^4 (C4)	$(1-\xi)^6\left(\frac{35}{3}\cdot\xi^2+6\xi+1\right)$

Table 1 Typical RBF functions.

An interpolation function composed of a radial basis φ and a polynomial h of order $m-1$, where m is said to be the order of φ, introduced with the aim to guarantee the compatibility for rigid motions, is defined as follows if N is the total number of contributing source points

$$s(x) = \sum_{i=1}^{N} \gamma_i \varphi\left(\left\|x - x_{k_i}\right\|\right) + h(x) \tag{9}$$

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, that is

$$s(x_{k_i}) = g_i, 1 \leq i \leq N \tag{10}$$

$$\sum_{i=1}^{N} \gamma_i q(x_{k_i}) = 0 \tag{11}$$

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 $m \leq 2$ [1], a linear polynomial can be used

$$h(x) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z \tag{12}$$

The subsequent exposition 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

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P^T} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \tag{13}$$

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

$$M_{ij} = \varphi\left(\left\|x_{k_i} - x_{k_j}\right\|\right), 1 \leq i \leq N, 1 \leq j \leq N \tag{14}$$

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 source points in the other three columns

$$\mathbf{P} = \begin{pmatrix} 1 & x_{k_1} & y_{k_1} & z_{k_1} \\ 1 & x_{k_2} & y_{k_2} & z_{k_2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N} & y_{k_N} & z_{k_N} \end{pmatrix} \tag{15}$$

RBF interpolation works for scalar fields. For the smoothing problem, each component of the displacement field prescribed at the source points is interpolated as follows

$$\left\{ \begin{array}{l} s_x(x) = \sum\limits_{i=1}^{N} \gamma_i^x \varphi\left(\left\|x - x_{k_i}\right\|\right) + \beta_1^x + \beta_2^x x + \beta_3^x y + \beta_4^x z \\ s_y(x) = \sum\limits_{i=1}^{N} \gamma_i^y \varphi\left(\left\|x - x_{k_i}\right\|\right) + \beta_1^y + \beta_2^y x + \beta_3^y y + \beta_4^y z \\ s_z(x) = \sum\limits_{i=1}^{N} \gamma_i^z \varphi\left(\left\|x - x_{k_i}\right\|\right) + \beta_1^z + \beta_2^z x + \beta_3^z y + \beta_4^z z \end{array} \right\} \tag{16}$$

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 and it is suitable for parallel implementation. In fact, once the solution is known and shared in the mem-

ory of each calculation 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. Furthermore, despite its meshless nature, 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.

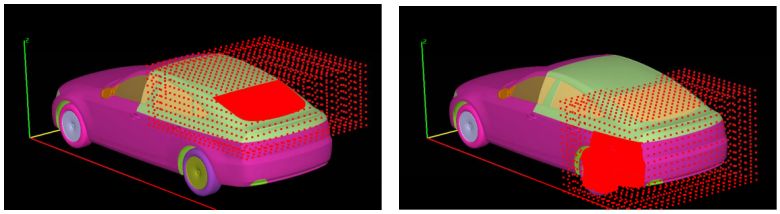
3.2 RBF Morph Tool

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 dataset (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. A detailed description of the usage of RBF Morph for the external aero optimization of the Volvo XC60 is given in [9] where the 50:50:50 approach demonstrates how 50 different shape variations can be explored using an high fidelity 50 millions cells mesh in less than 50 wall clock hours.

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 axis are present in the RBF set-up).

3.3 Coupling of RBF mesh morphing with adjoint sensitivities

Once the adjoint sensitivities are available as surface mesh information 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 result-



(a) rear window (b) boat tail

Fig. 1 Example of RBF points arrangement for the definition of two shape parameters. The rear window angle is controlled imposing a rigid rotation to the nodes on the window whilst preserving the shape of the roof and gently deforming the tail. The morphing volume is limited by a Box Encapsulation. Boat tail angle is changed applying a rotation around a proper axis of part of the rear car whilst preserving the shape of the wheel; also in this case the morphing action is limited by a box.

ing from its perturbation (keeping all the other constant). The sensitivity w.r.t. each given parameter is then obtained multiplying the surface perturbation field by the surface sensitivities. It is worth noticing 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 explored design point for which local sensitivities are required. The coupling can be used to enrich DOE based exploration for the parametric shape; in the industrial application presented herein, the parameters sensitivities are used in a local optimization method based on the gradient.

4 Optimization Algorithm

The gradient-based algorithm used to minimize the drag force is described in brief below:

1. Define the shape modification parameters, section 3.
2. Solve the flow equations, eqs. 1.
3. Compute the drag force value, $F_D = \int_{S_W} \left(-\tau_{ij} + p\delta_i^j\right) n_j r_i dS, \mathbf{r} = [1,0,0]^T$.
4. Solve the adjoint equations, eqs. 7.
5. Compute the deformation velocities and through them, the sensitivity derivatives, eq. 8.
6. Update the design variables by $\Delta b_i = -\eta \delta F / \delta b_i$, where η is a user-defined step.
7. Morph the car surface and displace the interior mesh nodes.
8. Unless the stopping criterion is satisfied, go to step 2.

5 Applications

In this section, the optimization algorithm presented in in section 4, in the form of an automated software, is used to minimize the drag force exerted on the surface of the DrivAer car model. In specific, the fast-back configuration with a smooth underbody, with mirrors and wheels (F_S_wm_ww) is used as a test case.

Six shape deformation variables (design variables) are defined in total. The part of the car surface parameterized by each of them and the corresponding deformation velocities are depicted in fig. 2.

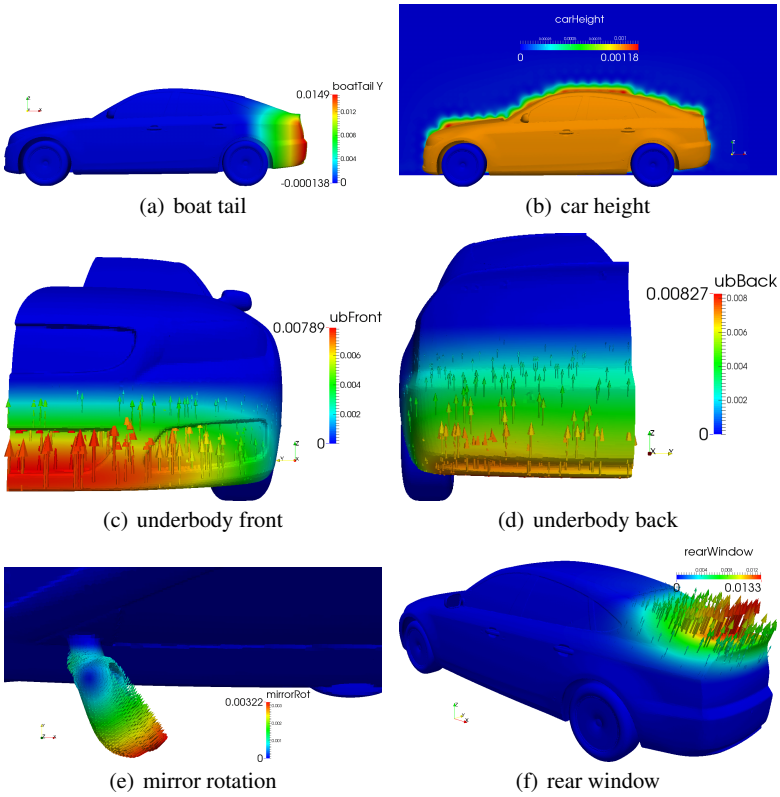
The minimization of the drag force is targeted by simultaneously varying all shape deformation parameters. A computational grid of approximately 3.8 million cells is used and turbulence is modeled by means of the Spalart–Allmaras model with wall functions. Even though the flow around a car varies in time, the steady state primal and adjoint PDEs are solved, to avoid the practical difficulties faced when solving the unsteady adjoint equations in medium and large scale computational grids [18], by proceeding backwards in time. Hence, the objective function cannot reach a constant value within each optimization cycle but oscillates around a "mean" value. The evolution of the objective function value during the flow solver iterations over the optimization cycles is shown in fig. 3.

The cumulative deformation magnitude after 15 optimization cycles, which led to a reduction by more than 7% in the mean drag value, is shown in fig. 4. The pressure field plotted over the initial and optimized geometries is depicted in fig. 5.

As expected, the area with the highest deformation is located, in the rear part of the car. In specific, two major trends are present. The first one is to lower the height of the rear window and to form a sort of a spoiler at the end of the trunk. This creates an area of increased pressure at the bottom of the rear window and despite the increased pressure on top of the formed spoiler, a resultant force that pushes the car forward is generated, fig. 6. The second trend is to create a "boat tail" effect (see fig. 4(c)) which leads to an increased pressure in the back side of the car, contributing thus to drag reduction.

6 Conclusions

In this paper, the continuous adjoint method and an RBF-based morpher, combined into an automated optimization software in the context of a research project funded by the EU, were used as the constituents of a gradient-based optimization algorithm, targeting the drag minimization of the DrivAer generic car model. A significant reduction in the drag value was observed after 15 optimization cycles which required approximately 16 hours on 64 Intel(R) Xeon(R) CPUs E5-2630 @2.30GHz. The utilization of the RBF-based shape modification parameters allowed the design of a smooth and manufacturable car shape.



(a) boat tail (b) car height

(c) underbody front (d) underbody back

(e) mirror rotation (f) rear window

Fig. 2 DrivAer shape optimization: Part of car surface controlled by the six shape deformation parameters and the corresponding deformation velocities ($\delta x_k/\delta b_n$).

Acknowledgements

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.

References

1. Beckert, A., Wendland, H.: Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. Constructive Approximation **5**(2), 125–134 (2011)
2. Biancolini, M., Viola, I., Riotte, M.: Sails trim optimisation using CFD and RBF mesh morphing. Computers & Fluids **93**, 46–60 (2014)

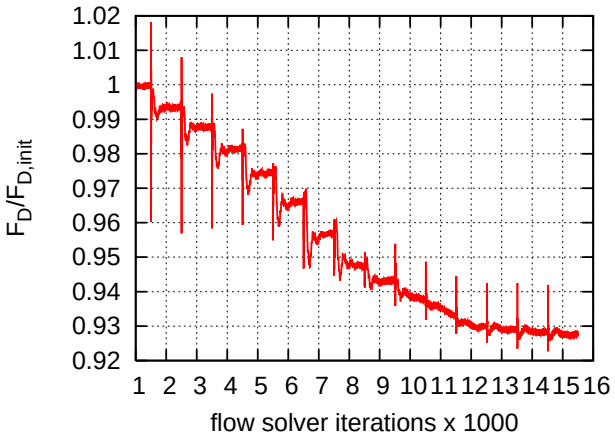


Fig. 3 DrivAer shape optimization: Evolution of the drag force value through the iterations of the flow solver over the optimization cycles, normalized with the mean value obtained using the baseline geometry. The flow solver run for 1000 iterations for each optimization cycle (a previously "converged" solution was used to initialize the optimization, so only 100 iterations were executed during the first optimization cycle). Kinks in the objective function value correspond to the first iterations after each shape update (new optimization cycle). A decrease of more than 7% percent can be observed in the "mean" drag value.

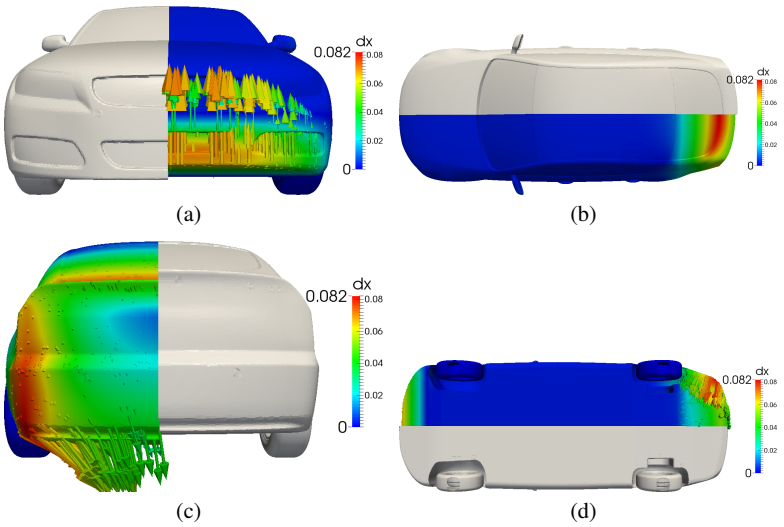


Fig. 4 DrivAer shape optimization: initial (starboard side) and optimized (port side) geometries. The latter is coloured based on the cumulative deformation of the car surface after 15 optimization cycles. The areas with the highest deformation and, thus, the higher impact on the objective value are the ones affected by the boat-tail and rear-window shape modification parameters.

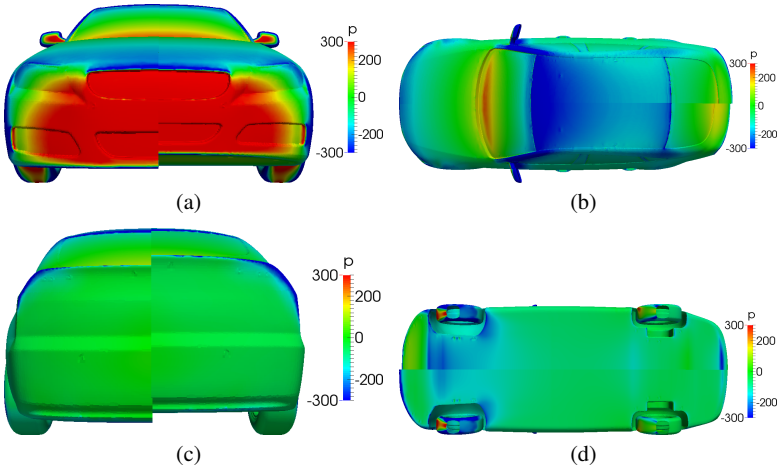


Fig. 5 DrivAer shape optimization: Pressure distribution over the initial (starboard side) and optimized (port side) geometries .

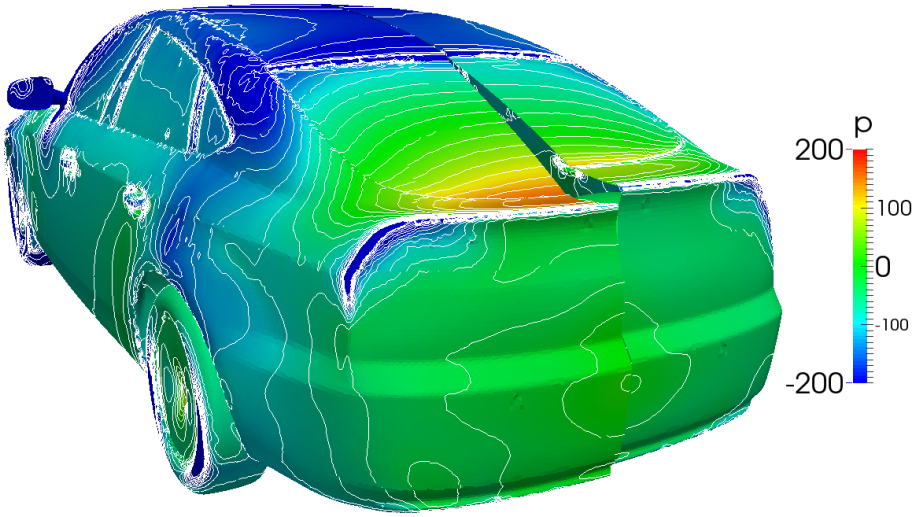


Fig. 6 DrivAer shape optimization: initial (right) and optimized (left) geometries, coloured based on pressure. Lowering the rear window, creating a spoiler at the end of the trunk and creating a boat-tail shape for the rear side lead to an increased pressure at the rear part of the car, contributing to drag reduction.

3. Biancolini, M.E.: 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), pp. 347–380
4. Giles, M., Pierce, N.: An introduction to the adjoint approach to design. Flow, Turbulence and Combustion **65**, 393–415 (2000)

5. Heft, A., Indinger, T., Adams, N.: Experimental and numerical investigation of the DrivAer model. In: ASME 2012, Symposium on Issues and Perspectives in Automotive Flows, pp. 41–51. Puerto Rico, USA (2012)
6. Jakobsson, S., Amoignon, O.: Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. Computers & Fluids **36**, 1119–1136 (2007)
7. Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T.: Harmonic coordinates for character articulation. Tech. rep.
8. Kavvadias, I., Papoutsis-Kiachagias, E., Dimitrakopoulos, G., Giannakoglou, K.: The continuous adjoint approach to the k–ω SST turbulence model with applications in shape optimization. Engineering Optimization, to appear **DOI:10.1080/0305215X.2014.979816** (2014)
9. Khondge, A., Sovani, S.: An accurate, extensive, and rapid method for aerodynamics optimization: The 50:50:50 method. SAE Technical Paper **01**(0174) (2012)
10. Martin, M.J., Andres, E., Lozano, C., Valero, E.: Volumetric B-splines shape parametrization for aerodynamic shape design. Aerospace Science and Technology **37**, 26–36 (2014)
11. Micchelli, C.: Interpolation of scattered data: Distance matrices and conditionally positive definite functions. Constructive Approximation **2**(1), 11–22 (1986)
12. Nadarajah, S., Jameson, A.: Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. AIAA Paper **25**(30) (2001)
13. Papoutsis-Kiachagias, E., Giannakoglou, K.: 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 (2014)
14. Papoutsis-Kiachagias, E., Kyriacou, S., Giannakoglou, K.: The continuous adjoint method for the design of hydraulic turbomachines. Computer Methods in Applied Mechanics and Engineering **278**, 612–639 (2014)
15. Robinson, T., Armstrong, C., Chua, H., Othmer, C., Grahs, T.: Optimizing parameterized cad geometries using sensitivities based on adjoint functions. Computer-Aided Design & Applications **9**(3), 253–268 (2012)
16. Spalart, P., Jou, W., Stretlets, M., Allmaras, S.: Comments on the feasibility of LES for wings and on the hybrid RANS/LES approach. In: Proceedings of the first AFOSR International Conference on DNS/LES (1997)
17. Thompson, P., Robinson, T., Armstrong, C.: Efficient CAD-based aerodynamic design optimization with adjoint CFD data. In: 21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences (2013)
18. Vezyris, C., Kavvadias, I., Papoutsis-Kiachagias, E., Giannakoglou, K.: Unsteady continuous adjoint method using POD for jet-based flow control. In: 11th World Congress on Computational Mechanics, ECCOMAS. Barcelona, Spain (2014)
19. Zymaris, A., Papadimitriou, D., Giannakoglou, K., Othmer, C.: Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. Computers & Fluids **38**(8), 1528–1538 (2009)
20. Zymaris, A., Papadimitriou, D., Giannakoglou, K., Othmer, C.: Adjoint wall functions: A new concept for use in aerodynamic shape optimization. Journal of Computational Physics **229**(13), 5228–5245 (2010)