Gradient projection, constraints and surface regularization methods in adjoint shape optimization

Pavlos P. Alexias and Eugene De Villiers

Abstract This paper deals with the treatment of various problems that are present in adjoint-based shape optimization applications, in which a parameterization of the surface is absent. A general implicit smoothing algorithm is used to reduce high-frequency noise which might be present in the gradients that are calculated using a continuous adjoint solver. The implicit smoother allows the definition of patches on the shape that need to remain fixed during shape optimization and automatically secures surface gradient continuity between constrained and deformable patches. Along with the gradient smoothing, a surface mesh regularization algorithm is presented and used to support high-quality elements and mesh uniformity during each optimization step. In the end, the capability and the effectiveness of the method are demonstrated in various industrial test cases.

1 Introduction

In the context of gradient-based numerical optimization, the adjoint method is the most cost-effective way to calculate the gradients of an objective function with respect to the design variables. This is due to the fact that the cost of the adjoint gradient calculation is practically independent of the number of the design variables of the optimization problem \[3, 4, 1\]. Design variable independence allows the exploration of richer design spaces and consequently convergence to better optimum solutions without increasing the computational cost. In gradient-free methods, like evolutionary algorithms (EA), the curse of dimensionality limits optimization w.r.t. many design variables due to significant cost increases.

Pavlos P. Alexias
Engys S.R.L., Via del Follatoio 12, Trieste 34148, Italy, e-mail: p.alexias@engys.com

Eugene De Villiers
Engys Ltd., Studio 20, RVPB, John Archer Way, London, SW 18 3SX, U.K e-mail: e.devilliers@engys.com
An important aspect of shape optimization applications is the selection of the design space and, accordingly, the selection of the design variables. A rational choice would be to keep the shape’s design parameters as design variables, but in CAD-free applications the existence of an analytical parametrization of the surface is not available for most applications. To deal with this problem, there are two main approaches to follow: 1) free-form deformation (FFD); in which the shape is enclosed by a hull object (e.g. a lattice) and 2) the node-based approach; in which each surface node of the computational mesh is considered as a design variable. In free-form deformation the shape’s deformation is translated into the deformation of the hull based on interpolation functions. There are various free form deformation techniques in the literature which are utilized in shape optimization problems. They use several types of parametrization basis functions to describe the hull object such as B-Splines, NURBS [13], Radial Basis Functions (RBFs) [15] or Harmonic Coordinates [8].

A notable drawback of the FFD methods is that the optimum solution is affected strongly by the user input. A different choice of geometry for the hull object or a different number of design parameters may result in different solutions, which can negatively impact the effectiveness of the method and reduce its general utility.

In the absence of parametrisation, the surface nodes of the CFD mesh can be used as design variables. The latter approach offers the richest possible design space (for the given spatial discretization) allowing the generation of better optimum solutions and the implementation of high-complexity constraints. However, any numerical noise in the adjoint derivatives, combined with the fact that each surface node is being perturbed independently from its neighbours, can create oscillations and irregularities. To avoid reducing the smoothness of the surface and negatively impacting the convergence of the optimization problem it is necessary to generate a smooth representation of the gradient. In FFD methods this smoothing takes place naturally through the given parametrization projecting the gradients from the mesh surface onto the parametric subspace. In node-based approaches the most established methods are the implicit smoothing technique [5], also also known as Sobolev gradient smoothing, and an explicit technique which uses Gaussian filter kernels [11].

In the present paper, the implicit technique will be used, in a modified form, that provides the advantages of Stück’s and Rung’s explicit technique in terms of translating the smoothing parameters into meaningful design variables. In addition, a mesh surface regularization algorithm will be presented and used effectively to prevent the degeneration and the appearance of low quality elements caused by big surface deformations. The gradients computed by the adjoint method are w.r.t. the point normal directions of the surface, making it necessary to re-order the points in the tangent direction to keep the mesh’s surface elements uniformity and quality. The algorithm is based on the maximization of a mesh quality metric using analytical derivative expressions of geometric mesh quantities.
2 The continuous adjoint method

A brief overview of the continuous adjoint method [16, 14] will be presented in this section. Without loss of generality, assuming a laminar flow of an incompressible fluid, the Navier-Stokes equations can be written as,

$$
R^p = -\frac{\partial v_j}{\partial x_j} = 0
$$

(1)

$$
R^r_i = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j} \left[ \nu \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{\partial p}{\partial x_i} = 0 \quad i = 1, 2, 3
$$

(2)

where $v_i$ is the velocity in direction of the Cartesian coordinates, $\nu$ is the kinematic viscosity and $p$ is the static pressure field divided by the fluid density $\rho$. 1 Defining a general objective function that contains both surface and volume integrals as

$$
F = \int_S F_S dS + \int_{\Omega} F_{\Omega} d\Omega
$$

(3)

then $F$ is augmented by the state equations leading to

$$
F_{aug} = F + \int_{\Omega} u_i R^r_i d\Omega + \int_{\Omega} q R^p d\Omega
$$

(4)

Here, $u_i$ and $q$ are the adjoint variables of the flow velocity and the static pressure respectively. Note that since the Navier-Stokes equations are satisfied, it holds that $F_{aug} = F$. Next comes the differentiation of the augmented objective function using the Green-Gauss theorem to pass from volume integrals to surface integrals. Zeroing the partial derivatives of flow field values results in the adjoint equations and adjoint boundary conditions:

$$
R^a_i = \frac{\partial u_i}{\partial x_i} - \frac{\partial F_{\Omega}}{\partial p} = 0
$$

(5)

$$
R^r_i = -v_j \frac{\partial u_j}{\partial x_j} + u_j \frac{\partial v_i}{\partial x_j} - v \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{\partial q}{\partial x_i} + \frac{\partial F_{\Omega}}{\partial v_i} = 0
$$

(6)

After solving the adjoint equations, in similar way with the primal Navier-Stokes equations, the final sensitivities values w.r.t. the design variables $b_n$ are,

$$
G = \frac{\delta F_{aug}}{\delta b_n} = -\int_S \left[ v \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - q n_i \right] \frac{\partial v_j}{\partial x_k} \frac{\partial x_k}{\partial b_n} dS
$$

(7)

---

1 On the aforementioned equations and on what follows, the Einsteins summation convention applies for the lower-case indices, unless declared differently.
3 Implicit Smoothing

The absence of a parametric description for the geometry or the existence of a low-quality computational mesh or partially converged primal solution (eq. 1), may result in the intrusion of noise into the adjoint sensitivities. In the node-based framework, the surface sensitivities are translated directly to node displacements and this will transfer the noise from the sensitivities on the surface. This noise intrusion can lead to impractical, non-manufacturable shapes and make our optimization problem difficult to converge (or even diverge). It is thus necessary to create a smooth gradient representation to cut-off any undesired oscillation.

3.1 Implicit Smoothing Equation

Based on the Sobolev gradient projection introduced by Jameson et al. [3], the smoothed gradient field \( \bar{G} \) is calculated from the initial gradient \( G \) through the diffusion-like equation

\[
\bar{G} - \epsilon \nabla^2 \bar{G} = G
\]  

(8)

where \( \epsilon \) is the smoothing coefficient which defines the smoothness of the final gradient representation. This generates a free variable for the optimization problem and thus the choice of the smoothing coefficient is case-specific. There are various suggestions in the literature for an optimum choice of \( \epsilon \) [9, 7]. In the present paper, based on the work of Stück and Rung [11], in which they use an equivalent explicit technique, the coefficient \( \epsilon \) will be translated into a maximum allowed oscillation radius on the shape’s surface. This can be done by taking the fundamental solution of the diffusion equation of a point source (for 2 variables)

\[
\Phi(x, \epsilon) = \frac{1}{4\pi \epsilon} e^{-x^2/4\epsilon}
\]  

(9)

where it can be noticed that the above equation is identical to the Gauss function with a standard deviation of \( \sigma = \sqrt{2\epsilon} \). This, allows the definition of a smoothing radius equivalent to three times the standard deviation of the Gauss bell in which every oscillation with a radius smaller than the defined will be severed.

Solving an elliptic equation implicitly will result in the spreading of sensitivity information inside the whole computational domain. However, only 0.3% of the information from a point source will pass outside of the 3\( \sigma \) smoothing radius. For all intents and purposes the smoothing will thus be acting locally inside the predefined radius.
3.2 Constraints and continuity

Smoothing the sensitivities through the solution of a PDE allows a straight-forward definition of constrained areas on the optimized shape. Zeroing the sensitivities on the non-moving patches and defining a fixed value boundary condition on the edges between deformable and constrained patches will result in having non-zero sensitivities only on the deformable patches. Even though applying a fixed value boundary condition on the common edges will prevent the smoother from generating sensitivities on constrained patches, there is no control on the way the shape transits from a constrained to a deformed patch. In cases where there are big sensitivity magnitudes in the vicinity of common edges, strong discontinuities will arise. Since it is important to maintain a smooth transition between the constrained and the deformable patches, the sensitivities with a geodesic distance from the fixed boundaries smaller than half of the smoothing radius are set to zero. In this way, the points close to the transition zone will be exactly on the Gauss bell curve maintaining $C^2$ continuity.

The geodesic distance calculation is performed by solving the eikonal equation

$$|\nabla \phi(x)| = \frac{1}{f(x)}, \quad x \in \Omega$$

subject to boundary condition $\phi|_{\partial \Omega}$. Solving for $\phi(x)$ will result in the shortest time needed to travel from the boundary $\partial \Omega$ to any point $x$ inside $\Omega$ with a velocity $f(x)$. In the special case where $f(x) = 1$ the solution $\phi(x)$ is the shortest distance of $x$ from $\partial \Omega$.

The effect and the importance of securing a smooth transition is demonstrated in figure 1. Smoothing is performed on a uniform displacement with superimposed noise (figure 1a), while respecting a circular constraint. Solving the equation 8 without transition treatment, will result on the shape depicted in figure 1b which does not have a continuous surface gradient at the common edge. Figure 1c demonstrates how a smooth transition can be achieved at the constrained boundary by zeroing the displacement values in elements with a distance from the constraint patch smaller than half of the smoothing radius.

4 Numerical Implementation

In the present paper, the code, the solvers and the applications have been developed under the open source framework of OPENFOAM. The sensitivity smoothing pertains to the solution of a two-dimensional partial differential equation (PDE) in a three-dimensional surface. In the case of a homogenous surface mesh with known topology, the equation can be solved moving between the Euclidian and the curvilinear space through covariant and contravariant functions. In the general case of unstructured grids this method will not work without obtaining an implicit surface representation [6]. In the present paper, a different approach will be followed that
Fig. 1: Differences in the shapes when solving the implicit smoothing equation (eq. 8) with (1c) and without (1b) ensuring a smooth transition from the constrained to the deformable patch.
Gradient projection, constraints and surface regularization methods

allows the solution of any PDE via the so called finite area approach. The method is implemented in the same way as the finite volume method in OPENFOAM, with the difference that the discretized elements are polygonal faces with individual local coordinate systems. The surfaces curvature enters the calculations through tensor transformations of every face centroid into a global coordinate system [10].

5 Surface mesh regularization

In the node-based method, the gradients of the objective function are computed w.r.t. the normal directions of the nodes on the surface. Moving continuously towards the normal direction, even with a smooth gradient representation, will eventually lead to the generation of low-quality surface mesh elements, tangled faces or high non-uniformity. To avoid this, a separate movement in the tangent direction is necessary for the restoration of the mesh quality.

In this section a mesh optimization algorithm aimed at the maximization of an element-wise quality metric will be presented. The goal is to calculate the new set of surface node positions that maximize this metric, assuming that the movement of the surface nodes in the tangent direction will not have a significant impact. The aforementioned assumption can be rationalised in the context of iterative surface motion, where the point locations are updated repeatedly.

The quality metric is defined for every surface element as

$$\mu = \alpha \frac{S_e}{P_e}$$

where $S_e$ is the surface of the element, $P_e$ is the perimeter and $\alpha$ is a normalization factor. The goal is to reposition the vertices of the surface mesh such that the faces will have the maximum metric value. In order to do so, it is necessary to calculate the derivative of the metric w.r.t. the position of the points that constitute a face. An extensive analysis of the differentiation of such kind of quality metrics goes beyond the scope of this paper. For a detailed description on the differentiation of geometric quantities in discrete geometry refer to Alexias and De Villiers [17].

If a point $P$ on the surface, is surrounded by $N$ faces, the total derivative of that point is the arithmetic mean of the sensitivities from all contributing faces.

$$\delta = \frac{1}{N} \sum_{n=1}^{N} \frac{d\mu}{dP}$$

Having obtained the sensitivity derivatives of the objective function, an optimization step is performed through a quasi-Newton method as

$$P^{\text{new}} = P^{\text{old}} + \lambda H^{-1} \cdot \delta$$
where $H$ is the Hessian matrix approximation using a limited memory BFGS method [2].

An example of the surface mesh regularization algorithm is demonstrated in figure 2 on a simple surface deformation case. Applying a displacement field on a flat surface may result in the creation of a surface mesh with distorted and anisotropic elements. Using mesh regularisation during the deformation results in a higher quality mesh with greater uniformity. The impact of the optimization algorithm on the mesh quality can be independently quantified by examining an alternative mesh quality metric, like aspect ratio. Figure 3 depicts the aspect ratio before and after the regularization. Taking into account that the optimum aspect ratio value is one, it is clear that the optimisation procedure significantly improves this metric, leading to a higher quality surface mesh.

6 Applications

In this section, the algorithms and the methods, presented in this paper, are combined to achieve a fully automated optimization process with minimal user intervention. The efficacy of the method will be demonstrated in two large-scale industrial test cases consisting of internal and external flow respectively.

6.1 Power losses minimization

The first application aims at the minimization of the power losses of an air S-bend duct. The flow is laminar with a Reynolds number of 350, having a structure computational mesh comprising 700 thousand cells. Figure 4 illustrates the duct geometry and the parts of the duct that are allowed to move during the shape optimization. The objective function subject to minimization is

$$ F = \int_S \left( p + \frac{1}{2} v^2 \right) v_n dS $$  \hspace{1cm} (14)

Following a steepest decent optimization method every new position of the surface nodes is given by

$$ x^i = x^{i-1} - a \bar{G} \cdot n $$  \hspace{1cm} (15)

where $\bar{G}$ is the smoothed gradient, $a$ is the step of the steepest decent and $n$ is the normal direction of each point. Figure 5 illustrates the differences between smoothed and raw gradients when applying a smoothing radius of 1cm. The smoothed gradients satisfy the constraints and simultaneously allow smooth transition between the constrained and unconstrained patches.
Fig. 2: Surface mesh without (top) and with (bottom) mesh regularization after the surface displacement.

Fig. 3: Aspect ratio of surface mesh elements before (left) and after (right) mesh regularization.
Solving the eikonal equation will result in the calculation of the geodesic distance curves on the 3D deformable surface. To ensure a smooth transition a zero-band zone needs to be created and thus the points belonging to a distance smaller than half the smoothing radius (0.5 cm in the present case) will be assigned a zero gradient value. Figure 6 illustrates the geodesic distances on the deformable surface of the S-bend duct.

After performing the steepest decent optimization step, the surface optimization takes place: improving the surface mesh quality and maintaining the mesh uniformity. Thus, the final node locations are calculated through

$$x_{final}^i = x_i + \delta_{total}$$ (16)

where $\delta_{total}$ is the total displacement resulting from the mesh optimization algorithm. Figure 7 illustrates the difference between having a smooth transition with a surface mesh optimization and simplistic gradient smoothing. In the first case, the mesh has improved uniformity and higher quality faces. Most importantly, there is
no "step" in the surface at the interface. This “step” typically has a negative impact on the optimization problem as it creates inferior quality elements that hamper the convergence of the CFD solution.

Fig. 7: Differences between deformed shape using smooth transition and mesh optimization (left) and using only gradient smoothing (right).

Beyond the surface displacement algorithm, it is necessary to adapt the internal mesh points for re-solving the CFD equations. For this purpose, a Laplacian equation with an inverse distance diffusion coefficient is solved combined with a mesh optimization approach that guarantees a high-quality mesh, even during extreme deformations.

Figure 9 displays the pressure losses reduction history as a function of the optimization cycles. It can be seen that after 35 optimization steps the pressure losses were reduced by 17.1%. It is noticeable that on every optimization cycle the mesh quality remained exceptionally high allowing us to proceed through a series of complex shapes and deformations without the need of remeshing. This contributes a lot to the automation of the optimization procedure and the reduction of the computational time.
Fig. 8: Optimized sBend duct shape, achieving a 17.1% power losses reduction.

Fig. 9: Pressure losses reduction for every optimization step.

6.2 Drag force minimization

The second application aims at the minimization of the drag force of the DrivAer car model [12]. In detail, the fast-back configuration with a smooth underbody, with mirrors and wheels is used in this application. Only half of the car geometry is meshed and used for the simulation, with a computational grid comprising around 6 million cells. The flow is turbulent and modelled with the Spalart-Allmaras turbulence model with wall functions. Even though the flow around a car does not reach a steady state solution, a time invariant CFD model is used for the simulation. This simplifies the optimization procedure, avoiding barriers and difficulties which arise when we are dealing with unsteady adjoint equations.
As it can be seen from figure 11, illustrated with green color, only a portion of the rear part of the car is allowed to move during the optimization. After 8 optimization cycles using a steepest decent method the algorithm converged to a shape with a reduction in drag of more than 2% (figure 10). Figure 12 demonstrates the displacement field (after applying the smoothing algorithm) for the first optimization cycle. It can be seen that the highest deformation lies on a thin strip on the rear part of the trunk of the car. Observing the figure 13, which compares the initial and the optimum shape, the generation of a spoiler, by lowering the trunk, leads to an increased pressure on the rear part of the car contributing to the reduction of the drag force. This percentage of drag reduction (2.1%) may seem small but is significant considering that only a small portion of the rear car was subject to shape deformation during the optimization.

Fig. 10 Convergence history of the drag force for every optimization cycle. A reduction of more than 2% can be observed.

Fig. 11: DrivAer car geometry. With green color is the deformable part while the rest has to remain fixed during the optimization.
7 Conclusions

In this paper, a complete and fully automated framework has been developed to deal with gradient and surface treatment problems in the context of adjoint-based optimization. The gradients of an objective function w.r.t. the mesh surface nodes are computed using the continuous adjoint method. Those gradients are smoothed
via an implicit solver that removes any unnecessary oscillation and noise. Using the same smoothing framework in conjunction with geodesic distances resulted in the proper application of constraints while keeping a desirable level of continuity. As a final step, a quality based optimisation was performed on the surface. The methods efficacy was demonstrated by achieving significant performance improvements for both internal and external flow test cases.

Acknowledgements This work has been conducted within the IODA project (http://ioda.sems.qmul.ac.uk), funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

References

1. Pironneau O (1974) On optimum design in fluid mechanics. J. Fluid Mech., 64(1):97-110.
2. Nocedal J, (1980) Updating Quasi-Newton Matrices with Limited Storage, Mathematics of Computation.
3. Jameson A (1988) Aerodynamic design via control theory. Journal of Scientific Computing 3 (3), pp. 233-260.
4. Jameson A (1995) Optimum aerodynamic design using CFD and control theory. AIAA-1995-1729-CP
5. Jameson A. and Vassberg J.C. (2000) Studies of alternative numerical optimization methods applied to the brachistochrone problem, Computational Fluid Dynamics, 9:281-296.
6. M Bertalmio, L T Cheng, S Osher and G Sapiro (2001) Variational problems and partial differential equations Journal of Computational Physics, 174:2, pp. 759-780, TBW.
7. Gherman I, Schulz V (2005) Preconditioning of one-shot pseudo-timestepping methods for shape optimization. PAMM Proc. Appl. Math. Mech., 5(1):741-742.
8. Joshi P, Meyer M, Derose T, Green B and Sanojci T, (2007). Harmonic coordinates for character articulation. ACM Trans. Graph. 26, 3, 71.
9. Schmidt S, Ilic C, Gauger N, Schulz V, (2008) Shape gradients and their smoothness for practical aerodynamic design optimization. Preprint-Number SPP1253-10-03.
10. Z Takovic, H Jasak (2009) Simulation of Thin Liquid Film Flow Using OpenFOAM Finite Area Method. 4th OpenFOAM Workshop, Montreal, Canada.
11. Stück A and Rung T (2011) Filtered Gradients for Adjoint-based Shape Optimisation, 20th AIAA Computational Fluid Dynamics Conference 27 - 30 June 2011, Honolulu, Hawaii.
12. Heft, A., Indinger, T., Adams, N. (2012) Experimental and numerical investigation of the DrivAer model. ASME 2012, Symposium on Issues and Perspectives in Automotive Flows, pp. 4151. Puerto Rico, USA
13. 0. Martin, M.J., Andres, E., Lozano, C., Valero, E. (2014) Volumetric B-splines shape parametrization for aerodynamic shape design. Aerospace Science and Technology 37, 2636
14. Papoutsis-Kiachagias, E, 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.
15. E.M. Papoutsis-Kiachagias, S. Porziani, C. Groth, M.E. Biancolini, E. Costa, K.C. Giannakoglou(2015) Aerodynamic Optimization of Car Shapes using the Continuous Adjoint Method and an RBF Morpher. EUROGEN 2015, Glasgow, UK, September 14-16.
16. Georgios K. Karpourzas, Evangelos M. Papoutsis-Kiachagias, Thomas Schumacher, Eugene de Villiers, Kyriakos C. Giannakoglou, Carsten Othmer (2016) Adjoint Optimization for Vehicle External Aerodynamics, International Journal of Automotive Engineering, Vol.7, No. 1, pp.1-7.
17. Alexias P, De Villiers E. (2016) Sphericity: Mesh optimization for arbitrary element topology, VII ECCOMAS Congress, 5-10 June 2016, Crete Island, Greece.