6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

SOFT HANDLE TRIGGERING: A CAD - FREE PARAMETERIZATION TOOL FOR ADJOINT - BASED OPTIMIZATION METHODS

Athanasios G. Liatsikouras[1,2], Guillaume Pierrot[3]

[1] ESI Software Germany GmbH
Kruppstr. 90, ETEC H4, 3.OG, 45145, Essen, Germany

[2]National Technical University of Athens
School of Mech. Eng., Parallel CFD & Optimization Unit, Athens, Greece
thanasis.liatsi@gmail.com

[1] ESI - Group
99 Rue des Solets, 94513 Rungis, France
guillaume.pierrot@yahoo.fr

Key words: Smoothing, Handles/Parameters, Numerical noise, CAD-Free Parameterization, Adjoint-based Optimization, Finite Transformation Rigid Motion Mesh Morpher.

Abstract. In gradient–based optimization methods, such as in adjoint–based methods, after having computed the sensitivities, the necessary shape changes should be applied. The sensitivity vector is often affected by numerical noise, due to the limited resolution of the discretization schemes, thus it cannot be used directly to change the shape, because the resulting shape might contain noisy surfaces which may be infeasible to be manufactured. In this work, a CAD–free parameterization tool is proposed, based on [1], which aims to enforce smoothness to the resulting shape. A subset of the nodes belonging to the surface are selected as handles/parameters and are responsible for controlling the surface changes. The displacement field of the surface nodes attempt to match the target displacement field of the handles, whilst ensuring that, smoothness requirements are enforced. After having changed the boundaries, the volume mesh should also be adapted to the updated geometry so as to proceed with the optimization. One way to achieve this is by re–meshing, but this process is time–consuming and difficult to automate. An efficient alternative is to adapt the existing mesh to the updated boundaries by using a mesh morpher. Herein, the Finite Transformation Rigid Motion Mesh Morpher [2] is used to adapt the mesh to the necessary shape changes. The efficiency of the proposed parameterization tool is demonstrated as a standalone tool and as constituent of a discrete adjoint solver.

1 INTRODUCTION

During the last years, a significant improvement has been achieved in the field of the CFD–based aerodynamic shape optimization. Engineers are interested in obtaining the optimal configuration of a shape over all possible transformations with respect to (w.r.t.) some measure of the cost, also known as (a.k.a.) objective function. The main constituents needed for performing a fully automated shape optimization loop include the flow solver, a shape parameterization method, an optimization technique and a way to adapt the computational mesh to the updated boundaries. Typical examples of shape optimization might involve the design of ducts or manifolds for minimum power or pressure losses, cars with optimal combination of drag and lift, aircraft for maximum lift etc.

Optimization methods can be distinguished into two main categories, i.e. stochastic (or gradient–free) and gradient–based techniques. Stochastic optimization methods, like evolutionary algorithms, are very efficient when the number of design variables is reasonably small, since the curse of dimensionality limits the optimization process. This makes them impractical in large–scale problems. On the other hand, the gradient–based optimization methods, with the adjoint method [3, 4] being the most well–known representative for computing the gradient w.r.t. the design variables, require more effort to develop and maintain, but the biggest advantage of such methods is that the cost per optimization cycle is independent from the number of design variables. The adjoint method can further be separated into two sub–categories; the discrete [5] and the continuous [6] adjoint method. Each of them has their share of challenges. In this work, an in–house discrete adjoint solver has been used to compute the gradient of the cost function w.r.t. the design variables and an in–house steady–state flow solver [7]. The latter is used to solve the Navier–Stokes equations for incompressible flows and it is based on a mixed hybrid finite volume scheme (MHFV).

One important decision to be taken in the optimization process, is the selection of the design space, or in other words, the design variables. Since in CAD–free, in contrast to CAD–based parameterization techniques, there is no analytical expression that can describe the shape to be studied, a way to control the shape should be available. A very well–known way to do that is with the use of the so–called free–form deformation (FFD). In the latter, the shape to be optimized is enclosed by a cloud of points (a.k.a. lattice) which constitute the design variables. A basis function (such as radial basis functions [8], harmonic coordinates [9], volumetric B–splines [10] or NURBS etc.) is used to interpolate the movement of each node of the point cloud to the nodes of the surface mesh of the shape to be optimized. Such methods require user intervention and experience to form an appropriate point of cloud that surrounds the shape. Moreover, a different selection of point clouds might lead to different optimum solution, which means that the optimum geometry is affected by the user input.

A way to alleviate such problems in CAD–free optimization, with which this paper is dealing with, is to consider as design variables all the nodes of the surface mesh of the shape to be optimized. Such decision leads to the richest possible design space but it has some challenges. In the adjoint method, in which the goal is to compute the gradient

(a.k.a. sensitivity vector) w.r.t. the design variables, it is very often to contain numerical noise. Mainly, this occurs due to the limited resolution in the discretization schemes and thus, cannot be used directly to change/deform the boundaries. Moreover, since each node is free to move independently from its neighbours, oscillations are generated in the surface which might be unacceptable during the manufacturing process. For instance, a noisy blade surface might lead to numerical problems in subsequent optimization loops.

The aim of this work is to eliminate the aforementioned problems by developing the Soft Handle Triggering tool (SHT), a CAD–free parameterization tool, which will be responsible for enforcing smoothness to the resulting shape. A subset of the nodes of the surface mesh are selected as parameters (in our context we call them handles) and they are controlling the shape changes. In addition, after having changed the boundaries, a way to propagate this movement to the computational mesh is required. An efficient way to do so, is by using a mesh morphing tool. In this work, the Finite Transformation Rigid Motion Mesh Morpher (FT–R3M) [2] is used which has been proven capable of undergoing extreme deformations, whilst ensuring a good resulting grid quality. The parameterization tool, proposed in this work, is not capable of working independently and a mesh morpher that relies on the minimization of a functional is needed to support it. FTR3M is a functional–minimization–based mesh morpher, in which the target is to minimize an energy functional related to the deformation energy between the initial and the final geometry. Therefore, the smoothing of the boundaries takes place at the same time with the mesh morphing. More details are written in chapter 3.

2 MESH MORPHING

As mentioned above, the SHT tool needs a mesh morpher that relies on the minimization of a functional in order to be able to support it. The SHT tool is responsible for deforming the boundaries, which aims to enforce smoothness to the resulting shape and, a mesh morpher undertakes to adapt the computational mesh to the updated boundaries. The idea of the SHT tool is to augment the morpher's cost function with a supplementary penalization energy. Thus, the SHT can work with any morpher that obeys the minimization of a functional. In the literature, there are several mesh morphers with this property that could be used. A brief overview of such mesh morphers follows.

2.1 Linear elasticity

In the linear elasticity method [11], the *Hooke's* law with linearized hyper–elasticity at small strains is used, in order to handle the rotation and the mesh anisotropy. In this approach, the computational mesh is considered as a deformable body of an elastic material. The volume mesh is adapted to the shape changes by minimizing the deformation energy of the whole mesh. This energy to be minimized is given from the following expression

$$E_M = \frac{1}{2}\int_{\Omega}\Big[\lambda[Tr(\epsilon)]^2 + 2\mu Tr(\epsilon^2)\Big]d\vec{u} \tag{1}$$

where Ω is the domain in which the equations are defined, λ and μ are the and Lamé constants and Tr is the trace of a matrix. Moreover, the strain displacement is computed

as

$$\epsilon = \frac{1}{2}\big(\nabla u^T + \nabla u\big) \tag{2}$$

2.2 Laplacian

In this approach, also known as "Laplacian coordinates" [12, 13], the new/updated position of the nodes that belong to the interior mesh of the computational domain are given by minimizing the following energy

$$E_M = \frac{1}{2}\int_{\Omega} ||\nabla \vec{u}||^2 d\vec{u} \tag{3}$$

with Ω being the domain of integration. This is an efficient mesh morphing technique when the transformation of the body does not involve big rotations.

2.3 Finite transformation rigid motion mesh morpher

The FT–R3M [2] is an in–house mesh morphing tool of ESI group, developed by the same authors. FT–R3M is a mesh-less method and tool which gracefully propagates the movement of the boundaries (surface changes) to the internal nodes of the volume mesh, ensuring an as-rigid-as-possible motion. It does not require any inertial quantities or cell connectivities related to the mesh. Nodes are grouped into "stencils" which are required to deform in an *as-rigid-as-possible* way. The idea is to minimize an energy functional between the initial and the final state of the shape to be deformed which is given by the following equation

$$\begin{cases} E_M = \sum\limits_{s\in\mathbb{S}} w_s \sum\limits_{(i,j)\in s} \mu_{s,ij} \big|\big| R_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i) \big|\big|^2 \\ \\ s.t.\ \ R_s R_s^T = I_n,\ \forall s \in \mathbb{S} \end{cases} \tag{4}$$

where, $\mathbb{S}$ is the set of the stencils (point clouds), $\vec{X}_j$ and $\vec{x}_j$ are the position vectors of node j in the initial and the final state respectively and R_s is the rotation matrix of stencil s. Furthermore, w_s is a scalar weight per stencil that stresses the importance of some stencils as higher than some others' and $\mu_{s,ij}$ a weight per edge that accounts for mesh anisotropy. It has been proved in [2] that, FT–R3M is very robust and can handle mesh anisotropies and rotations very efficiently.

In this paper, FT–R3M has been used as a mesh morphing tool in which the SHT has been developed on top, in order to take advantage of its nice properties during mesh deformation. In section 3, more details are described of the contribution of SHT to FT–R3M.

3 SOFT HANDLE TRIGGERING CONTRIBUTION

The concept of the SHT tool is to augment the morpher's energy with a supplementary penalization energy. Let us denote by E_M the nominal energy of the mesh morpher (i.e.

FT–R3M) and by E_S the additional penalization energy. The final, augmented energy ($E_{aug} = E_M + E_S$), which is then required to be minimized is

$$\begin{cases} E_{aug} = \sum_{s\in\mathbb{S}} w_s \sum_{(i,j)\in s} \mu_{s,ij} \left|\left| R_s(\vec{X}_j - \vec{X}_i) - (\vec{x}_j - \vec{x}_i) \right|\right|^2 + \sum_{k\in\mathbb{H}} \alpha\lambda_k \left|\left| (\vec{x}_k - \vec{p}_k) \right|\right|^2 \\ \\ s.t.\ R_s R_s^T = I_n,\ \forall s \in \mathbb{S} \end{cases} \tag{5}$$

where $\mathbb{H}$ is the set of handles/parameters, $\vec{p}_k$ the target displacement of the handle node k and α is a scalar. For the time being, let us assume that the target displacements of the handles are given. Later on, it will be explained how this quantity is computed within the adjoint–based optimization loop.

The first step is to select the handles/parameters. The handles are a set of replicated surface nodes. Each handle has a specific target displacement, $\vec{p}_k$ and a weight coefficient, λ_k which normalizes the penalization energy w.r.t. the nominal energy of the morpher. The λ_k coefficient is a product of a weight w_s and the weight $\mu_{s,ij}$. The scalar α is a positive scalar which favours the penalization energy w.r.t. the morpher's energy in order to enforce stronger smoothness conditions. It is worth mentioning that the target displacement is not necessarily identical with the actual displacement, $\vec{x}_i$ of the underlying node. The actual displacement of a surface node is a compromise in between the target displacement of the corresponding handle and the smoothness requirements. Thus, the shape changes are driven from the target displacement of the handles and the computational mesh adaptation is done by the mesh morpher at the same time.

To solve such a system, the *Newton's method* [14] is used. If we denote by $\vec{b}$ the set of degrees of freedom, namely the nodal displacements $\vec{x}_i$ and the rotation matrices of each stencil R_s, then the system to be solved is the following

$$\mathbf{H}\delta\vec{b} = -\frac{\partial E_{aug}}{\partial \vec{b}}, \quad \mathbf{H} = \nabla^2 E_{aug} = \frac{\partial^2 E_{aug}}{\partial \vec{b}^2} \tag{6}$$

where $\mathbf{H}$ is the Hessian matrix of the augmented cost function E_{aug}.

4 GRADIENT COMPUTATION W.R.T. THE HANDLES

Since a discrete adjoint solver is used, the gradient of the objective function (J) is computed w.r.t. the nodal coordinates ($\vec{X}$), namely $dJ/d\vec{X}$. In order to keep a consistency within the optimization loop, the gradient w.r.t. parameters/handles must be computed ($dJ/d\vec{p}$). To do so, let us denote by G the intrinsic relation between the node coordinates and the parameters/handles. This is written as

$$G = G(\vec{X}, \vec{p}) = 0 \tag{7}$$

For the purpose of computing the gradient with respect to the parameters, a geometric adjoint system with Eq. (7) as master/slave equation, is solved. By differentiating the latter, we have

$$\left[\frac{\partial G}{\partial \vec{p}}\right]^T + \left[\frac{d\vec{X}}{d\vec{p}}\right]^T \left[\frac{\partial G}{\partial \vec{X}}\right]^T = 0 \tag{8}$$

Then, differentiating the objective function w.r.t. the parameters/handles yields

$$\left[\frac{dJ}{d\vec{p}}\right]^T = \left[\frac{\partial J}{\partial \vec{p}}\right]^T + \left[\frac{d\vec{X}}{d\vec{p}}\right]^T \left[\frac{\partial J}{\partial \vec{X}}\right]^T \tag{9}$$

Furthermore, since there is no explicit expression that relates the objective function with the parameters it yields that $\left[\partial J/\partial \vec{p}\right]^T = 0$. Now, introducing Φ, a Lagrangian operator, we can multiply it with Eq. (8) and add it to Eq. (9) we obtain

$$\left[\frac{dJ}{d\vec{p}}\right]^T = \left[\frac{d\vec{X}}{d\vec{p}}\right]^T \left[\frac{\partial J}{\partial \vec{X}}\right]^T + \left\{\left[\frac{\partial G}{\partial \vec{p}}\right]^T + \left[\frac{d\vec{X}}{d\vec{p}}\right]^T \left[\frac{\partial G}{\partial \vec{X}}\right]^T\right\}\Phi \tag{10}$$

By reformulating the augmented gradient with respect to the parameters, Eq. (10), we finally have

$$\left[\frac{dJ}{d\vec{p}}\right]^T = \left[\frac{d\vec{X}}{d\vec{p}}\right]^T \left\{\left[\frac{\partial J}{\partial \vec{X}}\right]^T + \left[\frac{\partial G}{\partial \vec{X}}\right]^T \Phi\right\} + \left[\frac{\partial G}{\partial \vec{p}}\right]^T \Phi \tag{11}$$

The derivative of the node coordinates with respect to the parameters, namely the term $\left[d\vec{X}/d\vec{p}\right]^T$, is the so-called "grid sensitivities". The grid sensitivities express the dependency of the each node coordinate with respect to a change in each parameters. The computational cost of this derivative is proportional to the number of parameters and since it is not desirable to compute this term, the expression that is multiplied with, is demanded to be equal to zero. As a consequence the adjoint equation is introduced, namely

$$\left[\frac{\partial G}{\partial \vec{X}}\right]^T X^* = -\left[\frac{\partial J}{\partial \vec{X}}\right]^T \tag{12}$$

where the Lagrangian operator Φ is replaced by the adjoint node coordinate vector, X^*. By solving the adjoint equation in Eq. (12), the adjoint node coordinate vector X^* is obtained and finally the gradient with respect to the parameters is computed as

$$\left[\frac{dJ}{d\vec{p}}\right]^T = \left[\frac{\partial G}{\partial \vec{p}}\right]^T X^* \tag{13}$$

From Eq. (13), it is shown that the gradient is independent of the number of handles, which make its computation very efficient. Then, after having computed the gradient w.r.t. the parameters, the latter is used from an optimization algorithm in order to compute the target displacement of each handle node.

5 ALGORITHM OF THE OPTIMIZATION FRAMEWORK

The gradient–based algorithm used for the aerodynamic shape optimization is described in brief below and its basic constituents are demonstrated in Fig. 1.

0. Generate a mesh and use a selection algorithm to choose a subset of the surface nodes as parameters/handles.

1. Solve the flow equations.
2. Solve the adjoint equations and obtain the sensitivity vector $dJ/d\vec{X}$.
3. Solve the adjoint system of the morpher, Eq. (12) and obtain $\vec{X}^{\star}$.
4. Compute the gradient w.r.t. the parameters/handles, Eq. (13).
5. Update the target displacement of the handles by $\vec{p}_i^{\,new}=\vec{p}_i^{\,old} \pm \eta\, dJ/d\vec{p}_i$, where η is a user–defined step.
6. Use FT–R3M and SHT to update the shape and the computational mesh.
7. Unless the stopping criterion is satisfied, go to step 1.

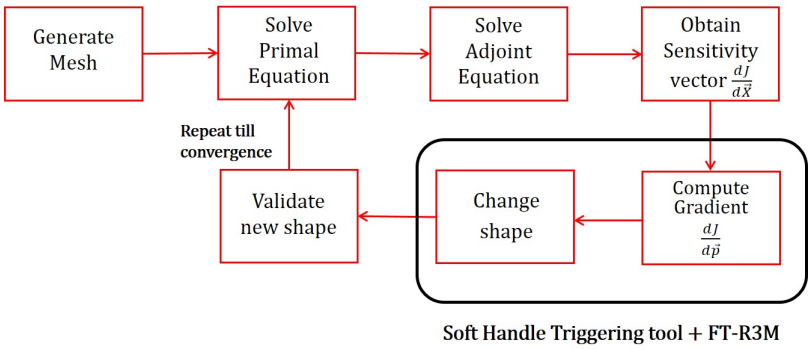


Figure 1: Basic constituents of the adjoint–based optimization workflow.

6 APPLICATIONS

In this section, the SHT tool with the FT–R3M are tested and demonstrated as standalone tools and as part of a discrete adjoint–based optimization loop.

6.1 Blade tip gap - Smoothing test case

The first case is dealing with the deformation of a 2D blade. In Fig. 2 the initial mesh and shape of the tip gap in the blade is shown. The mesh consists of approximately 33K nodes and 50K elements. The blade is being deformed and artificial numerical noise is applied on its movement, whilst the nodes marked in red in Fig. 2 are kept fixed. The purpose of this case is to validate the efficiency of the SHT as a standalone tool. In the study that follows, different values of the scalar α (see also section 3) are applied, to show the dependency of the resulting shape on this quantity (Fig. 3). All nodes that are free to move are considered as handles.

In Fig. 3, the efficiency of the SHT tool is demonstrated. The artificial noise that has been added to the deformation of the blade makes it unacceptable for manufacturing.

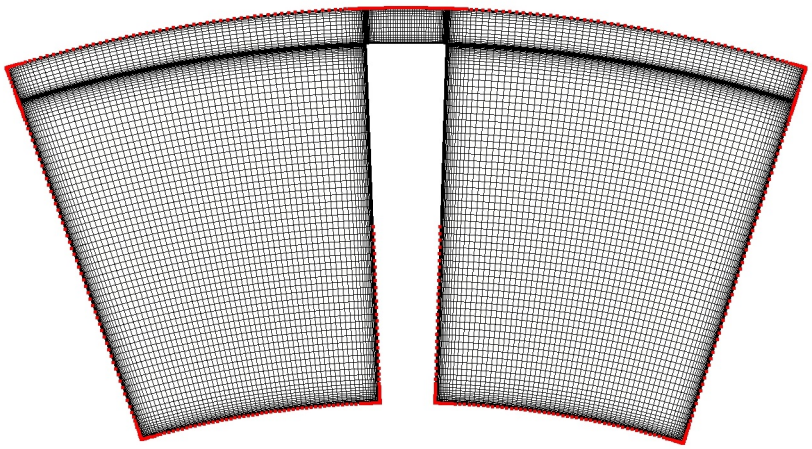
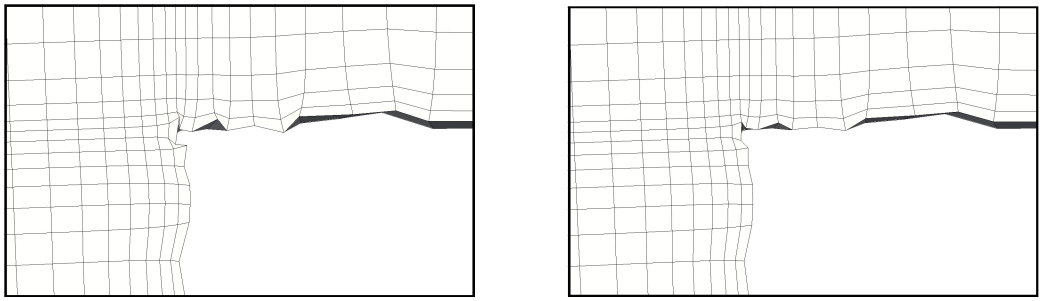


Figure 2: Blade tip gap case: Initial mesh and shape of the blade. Nodes marked in red are kept fixed, whereas the rest of the boundary nodes are being deformed.



(a) No use of the SHT tool

(b) $\alpha = 1$

(c) $\alpha = 0.1$

(d) $\alpha = 0.01$

Figure 3: Blade tip gap case: In this figure the resulting mesh is presented without and with the use of the SHT tool and for different values of the scalar balancing coefficient α.

After applying the SHT tool, the surfaces are smoothed and the shape obtains a manufacturable shape. This balancing coefficient α makes a compromise between the deformation energy of the FT–R3M and the supplementary penalization energy. The more noisy the displacement, the smaller this balancing coefficient should be. Of course, there are some limitations in the latter, because the smaller its value it is, the more ill–conditioned the

system to be solved becomes.

6.2 Adjoint–based optimization of an S–bend duct

This application is dealing with the S–bend duct, an air duct test case provided by VolksWagen AG. The optimization aims at minimizing J, where J stands for the total pressure losses across the duct. The latter is computed at the inlet/outlet, from the expression

$$J = \frac{\int \left(p + \frac{1}{2}\,\rho\vec{u}\right)\vec{u}\cdot\vec{n}dS}{\int \vec{u}\cdot\vec{n}dS} \tag{14}$$

where $\vec{u}$ is the velocity vector, p is the static pressure, ρ is the density of the fluid and $\vec{n}$ is the unit outward normal vector at the boundaries of the flow domain.

The baseline mesh consists of 480K nodes and 466K hexahedra. The flow is laminar with inlet velocity $u{=}0.1\,m/s$ (normal to the inlet boundary) and with Reynolds number approximately of $Re{=}400$, based on the hydraulic diameter of the inlet. In Fig. 4, the initial shape of the duct is presented. The blue part of the geometry remains fixed, whereas the central curved part (red region) is free to deform.

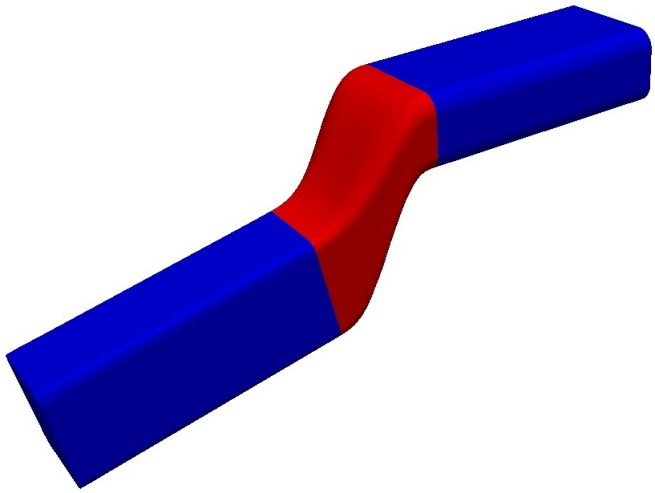
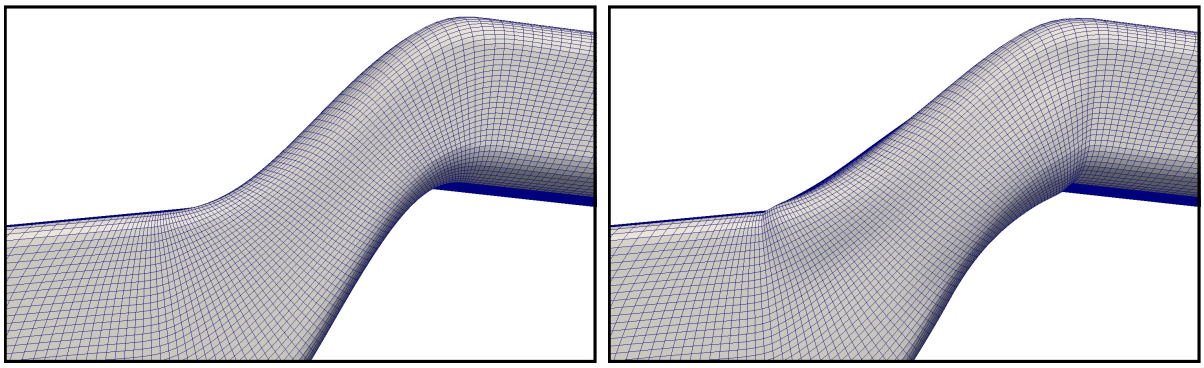


Figure 4: S–bend duct optimization: Initial shape of the duct. Regions marked in blue are not allowed to change, whereas the central curved part marked in red is free to deform.

After performing 30 optimization cycles, the total pressure losses are reduced approximately by 23.7% w.r.t. the initial. The new shape of the duct is shown in Fig. 5 in comparison to the initial one. The resulting/optimized shape is smooth and manufacturable and there is no "step" in the vicinity of the unconstrained and the constrained patches. In addition, in Fig. 6 the streamlines in the initial and the optimized shape are demonstrated and in Fig. 8 the convergence history of the optimization is plotted. It is noticed that the stagnation/recirculation area near the outlet of the duct has been limited in the optimized shape which is the main reason of the reduction of the pressure losses.



(a) Initial shape. (b) Optimized shape.

Figure 5: S–bend duct optimization: Close–up view in the central curved of the duct. In this figure the initial (fig. 5(a)) and the optimized (fig. 5(b)) shapes are compared. The volume of the optimized geometry has been increased during the optimization, leading to approximately 23.7% less total pressure losses.

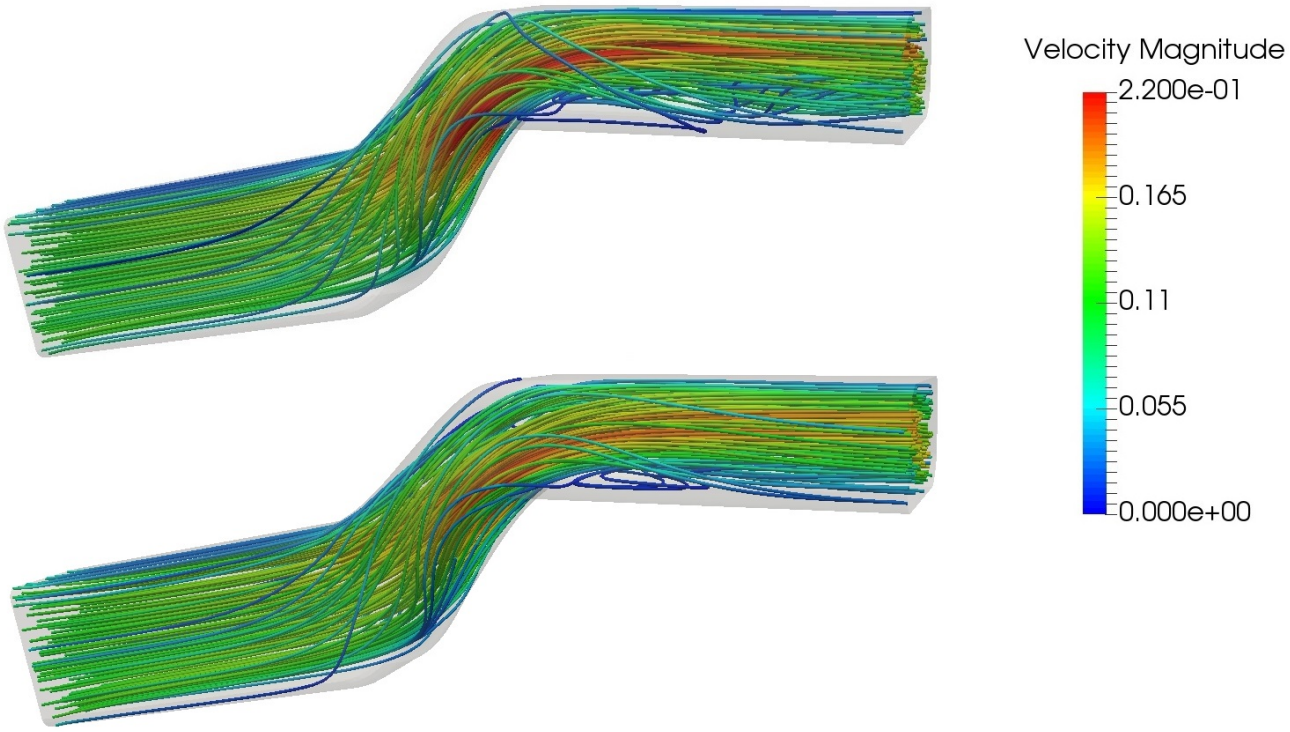


Figure 6: S–bend duct optimization: Streamlines in the initial (upper) and the optimized (bottom) shape of the duct. The intense flow recirculation area near the bottom side of the wall has been reduced in the optimized shape, which is the reason of the reduction in the total pressure losses across the duct.

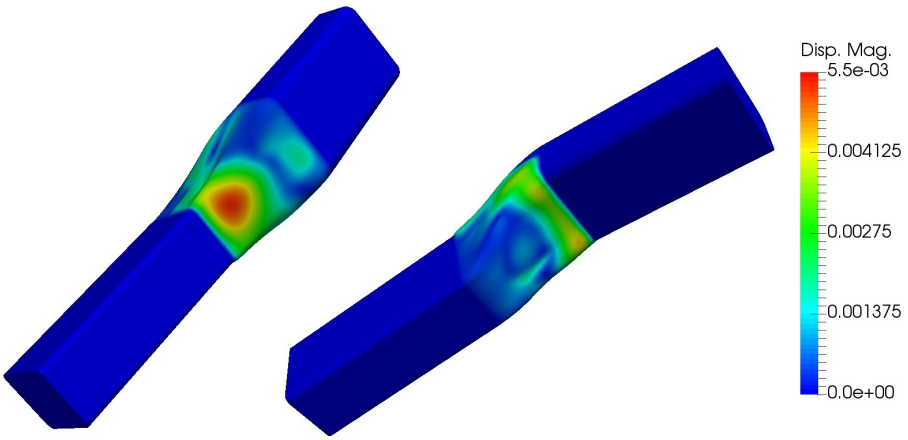


Figure 7: S–bend duct optimization: Optimal shape of the duct coloured based on the cumulative displacement on the upper side (left fig.) and the bottom side (right fig.). Flow from left to right.

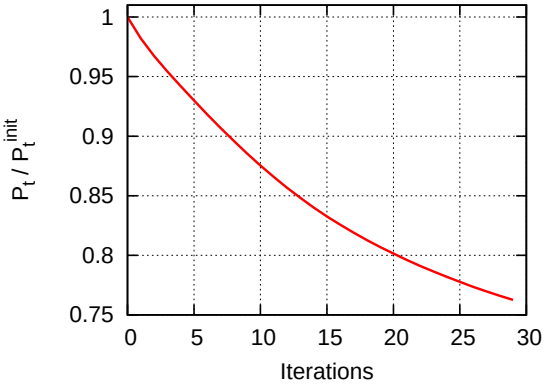


Figure 8: S–bend duct optimization: Convergence history of the optimization.

7 CONCLUSION

In this work, the SHT tool has been developed and presented together with FT–R3M, an in–house mesh morphing tool. A fully–automated CAD–free, adjoint–based shape optimization framework has been implemented which does not require any manual intervention. Since both FT–R3M and SHT are differentiated, the gradient consistency is maintained. The cases shown in this paper demonstrate the efficiency of both tools. Smoothness requirements on the surface are enforced and, their movement is gracefully propagated to the interior mesh.

8 ACKNOWLEDGEMENT

This work has been conducted within the IODA ITN on: "Industrial Optimal Design using Adjoint CFD". The first author is an Early Stage Researcher (ESR) in this project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska–Curie Grant Agreement No. 642959.

REFERENCES

[1] Z. Su, S. Wang, C. Yu, F. Liu, and X. Shi. A novel Laplacian based method for mesh deformation. *Journal of Information & Computational Science*, 7(4):877–883, 2010.

[2] A.G. Liatsikouras, G. Pierrot, G. Fougeron, and G.S. Eleftheriou. Finite transformation rigid motion mesh morpher. In *EUROGEN 2017*, Madrid, Spain, September 13-15 2017.

[3] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65(3-4):393–415, 2000.

[4] A. Jameson. Optimum aerodynamic design using CFD and control theory. *AIAA paper*, 1729:124–131, 1995.

[5] R. Roth and S. Ulbrich. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow, Turbulence and Combustion*, 90(4):763–783, 2013.

[6] E.M. Papoutsis-Kiachagias and K.C. Giannakoglou. Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications. *Archives of Computational Methods in Engineering*, 23(2):255–299, 2016.

[7] M. Oriani and G. Pierrot. A mixed hybrid finite volume scheme for incompressible Navier-Stokes. In *NAFEMS World Congress, San Diego (California)*, pages 21–24, 2015.

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

[9] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. In *ACM Transactions on Graphics (TOG)*, volume 26, page 71. ACM, 2007.

[10] M.J. 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.

[11] D.R. Lynch. Unified approach to simulation on deforming elements with applications to phase change problems. *Journal of Computational Physics*, 47(3):387–411, 1982.

[12] P. Hansbo. Generalized Laplacian smoothing of unstructured grids. *International Journal for Numerical Methods in Biomedical Engineering*, 11(5):455–464, 1995.

[13] S. Zhang, J. Huang, and D.N. Metaxas. Robust mesh editing using Laplacian coordinates. *Graphical Models*, 73(1):10–19, 2011.

[14] J. Nocedal and Stephen J.W. *Numerical Optimization*. Springer, New York, NY, USA, second edition, 2006.