TECHNISCHE UNIVERSITÄT MÜNCHEN

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Statik

Sensitivity analysis and regularization for shape optimization of coupled problems

Electra Stavropoulou

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. habil. Fabian Duddeck

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Univ.-Prof. Dr. rer. nat. Nicolas R. Gauger,
   Technische Universität Kaiserslautern

Die Dissertation wurde am 30.09.2014 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 09.03.2015 angenommen.

Abstract

In the last few decades there has been a great interest on improvement and enhancement of industrial designs in order to reduce costs and resources. For this reason computer aided engineering tools are increasingly applied. Numerical shape optimization automates this design process by use of computational methods. In this work, novel sensitivity analysis and regularization methods are developed for node-based shape optimization with applications to fluid and fluid-structure interaction problems.

In order to deal efficiently with the large number of design variables, gradient-based optimization methods are chosen and the sensitivity information is evaluated using the adjoint method. More precisely, emphasis is given to the adjoint sensitivity of the fluid-structure interaction problem. The coupled sensitivity analysis is the only way to achieve exact gradients since neglecting the interaction can lead to enormous inaccuracies. This is analyzed and successfully validated through numerical examples. Solution of the adjoint coupled system is a challenging task mainly because of the complexity of the coupling operators. Here, the adjoint problem is solved in a partitioned way similar to the primal problem and verification of the method shows very good agreement with finite differences as reference.

Moreover, special attention is given to the shape parametrization of the design which is decisive for the evolution of the shape. In this study, a node-based shape description is chosen in which the shape is described solely by its discretization, in order to have the widest possible design space. A drawback of node-based shape optimization is that the noisy sensitivity field results in non smooth shapes and element distortions. Two methods are developed to deal with this issue: the Vertex-Morphing and the In-plane regularization method. The Vertex-Morphing method is an explicit method which produces smooth designs while keeping the design features of the initial shape. The In-plane regularization deals with the mesh irregularities by applying an artificial stress field on the design mesh and can be used also as a general purpose mesh smoothing tool.

The overall design chain is presented and the robustness and accuracy of the methods is shown through the shape optimization of industrial cases.

Contents

Chapter 1

Motivation and objectives

1.1 Introduction

Optimal shape design receives great attention in aerospace, marine and automotive industry as well as in architecture and civil engineering. The aim of optimal shape design is to determine the shape that minimizes a functional which describes mechanical characteristics of the design. Nowadays, there is a tendency to automate this design process and bring the designs to their limits regarding efficiency and performance in order to reduce costs and resources. This work aims in design optimization of fluid and fluid-structure interaction problems.

Fluid-structure interaction occurs when an elastic structure interacts with a fluid flow. The flow exerts a pressure on the structure which results in a structural deformation. This deformation affects the fluid flow by changing its boundary. The problem is in general transient and there are already good tools developed for the analysis and numerical modeling of such a problem. Among others one can refer to the work of Farhat et al. [29, 30, 32], Felippa et al. [37], Bletzinger et al. [43, 72, 109], Wall et al. [42, 73], Löhner et al [76]. Turek et al. [61], Tezduyar et al. [117, 118], Quarteroni et al. [41], and Degroote [21].

The problem is in general involved. Apart from the modeling of the single fields, the exchange of information between the fluid and the structure, the stability of the overall problem, the treatment of the discretization and the way that the coupled system is solved are some challenges in this topic. The problem becomes even more involved when gradient-based optimization is of interest. In this case the sensitivity of a function which depends on the coupled state variables is required. This work investigates such a sensitivity analysis performed in the adjoint manner.

Observing the optimization workflow from a geometrical point of view, a key element in this process is the parametrization of the shape. Inspired from engineering design, Computer Aided Geometric Design (CAGD) methods have been commonly used to represent the shape for both structure [13, 27, 68] and fluid optimization problems [62, 104]. In this instance, the design parameters of the CAD model are the design variables of the optimization problem which control the shape. These parameters can be for instance the position of control points in case of NURBS or morphing boxes. The limited number of design variables allows for application of zero-order methods like evolutionary strategies [4] and genetic algorithms [48]. These methods do not require gradient information and are preferred in early optimization stages. Moreover, they are the only choice when gradient information is not available. However a shortcoming of CAD parametrization is that the low number of design variables restricts the design space and the resulting shape has always the same geometrical characteristics as the initial one.

This motivates the use of enriched design spaces with larger number of design variables. In contrast to CAD methods, the node-based shape optimization approach regards the design space to be as large as possible by considering the location of each surface point of the discrete design as a control parameter (design variable). Hence, the discretization of the design surface is also used to describe the geometry and provides the vertex coordinates as design variables for the optimization problem. This method suffers from mesh dependent results and non-smooth shape derivatives which almost always lead to unphysical and jagged shapes [89]. This work focuses on the regularization of the problem by presenting ways to obtain smooth shapes while maintaining mesh quality.

In the sequence, the two topics of interest are further introduced.

1.1.1 Adjoint sensitivity for coupled problems

For node-based shape parametrization, zero-order methods are not suitable due to the large number of design variables. Therefore a gradient-based algorithm shall be used. In such a case, the evaluation of the gradient information, i.e. the sensitivity analysis is a crucial step.

Among others, adjoint methods provide sensitivity values on every discretization element (cell, point, etc.) almost by the same cost as the primal problem [64, 112]. While these methods are extensively used and validated for the sensitivity analysis of structural [11, 50, 121] and fluid [46, 91, 103] problems, the gradient evaluation with a coupled problem as a constraint [33, 82, 111] is a relatively new field of research. The challenge here is the interpretation and understanding of the monolithic adjoint discrete system. More specifically, this coupled system has terms which are always computed implicitly in the primal problem. For the adjoint problem however, one has to derive these terms, in order to be able to solve the adjoint coupled equations in a partitioned way.

1.1.2 Regularization for node-based shape optimization

As mentioned before an issue in node-based optimization is that the problem is ill-posed and therefore shape and mesh irregularities are to be expected. For this reason, various projection methods have been suggested which smooth the shape derivatives or equivalently the shape variation field. An implicit smoothing method based on the Sobolev gradient is described in [63, 88, 89]. In this method the smooth gradients are obtained as a solution of a differential equation which penalizes the high curvatures of the sensitivity field. It performs very well and is widely used particularly in aerodynamic shape optimal design. This implicit smoother can be included in the calculation of the gradients as pre-conditioner. Furthermore, a method used in structural problems and proposed in [105] imposes shape and mesh regularity criteria as constraints to the original optimization problem. In this way, the mesh and the shape are controlled well, but the complexity of the problem is increased. Alternatively, explicit filtering methods smooth the shape derivative field in a separate computational step after sensitivity analysis. This type of filter can be implemented easily for black-box adjoint solvers. These methods have shown success both in structural optimization [39, 40, 74] as well as in aerodynamic optimization [116] applications.

The above mentioned methods act only on the “out-of-plane” direction of each design node. This is because by neglecting the discretization error, only the normal variations of the surface can change the shape. However, in discretized surfaces, updating the shape in the normal direction without considering the tangential direction leads to high distortion of the surface elements [66, 74, 105]. The quality of the mesh limits the flexibility in shape deformation, and decreases the accuracy of the results. Therefore, an “in-plane” regularization step is also required in combination with the shape derivative filter in order to ensure the quality of the surface discretization while morphing the shape [114].

The two procedures described above: the out-of-plane and the in-plane treatment of the surface can be performed consecutively or simultaneously. When applied consecutively, first the sensitivity smoothing is done, and the geometry is updated. After that and in a separate step, the mesh of the design surface is corrected with the In-plane regularization method described in [114].

Alternatively, these mechanisms can be combined in one simultaneous update step including both normal and tangential directions. This is the chosen strategy in the method developed in this work. The method is called Vertex-Morhing, and is a node-based shape parametrization technique presented in [56]. The method controls the normal and tangent update directions synchronously by introducing a control field which is discretized with as many nodes as of the geometry space. This field is linked to the geometry by a linear map. This definition bridges the idea of node-based and CAD-based shape optimization, since the role of the discretized control field is identical to the one of the control points in CAD parametrization. The main difference is that here, much more control points are used and moreover the location of the control points coincides with the surface points. Additionally, when a perfect conservation of the mesh quality is desired e.g. very large shape variations, the developed implicit In-plane regularization [114] is applied.

1.2 Objectives of the present work

The aim of this thesis is to develop a partitioned methodology for node-based shape optimization of the fluid-structure interaction (FSI) problem. The focus is on the following crucial aspects: first, the coupled sensitivity analysis for FSI and second, the regularization methods needed for such node-based optimization. Further on, the various methods are merged into a modular optimization framework capable of optimizing the shape of industrial complex geometries.

More precisely:

- The monolithic formulation of the adjoint coupled problem is presented and from this a partitioned problem is derived. Focus is on the definition of the various terms of the monolithic system and the formulation of the partitioned single field equations. The verification of the obtained sensitivities is also of interest.
- The Vertex-Morphing method is shown in order to treat the noisy design updates and further focus is given to the In-plane regularization method used for maintaining a good quality discretization during optimization.

The present thesis is organized as follows:

Chapter 2 examines how the sensitivity analysis can be performed for structure, fluid and coupled problems. The sensitivity analysis is the kernel of every gradient-based optimization problem. In this work the sensitivity analysis is performed using the adjoint method. For the structural problem the semi-analytical sensitivity analysis is explored [11] while for the fluid problem the continuous adjoint method is chosen [96]. In the coupled fluid-structure interaction problem the two single field formulations are combined and the adjoint FSI problem is solved in a partitioned way similar to the solution of the primal problem. Emphasis is given to the construction of the partitioned solution out of the monolithic FSI problem.

Having an accurate sensitivity field, a gradient-based node-based shape optimization can be performed and *Chapter 3* provides all the necessary regularization methods required for this task. More precisely, the Vertex-Morphing method is exploited for regularization of the problem. Additionally, the In-plane regularization method is presented in the sequence to provide further improvement on the surface mesh required for applications with a very large shape deformation. Various In-plane regularization cases are presented which establish the method as a general mesh smoothing and mesh refining tool.

Chapter 4 presents two CFD shape optimization cases where the aforementioned regularization methods are successfully applied. Here, attention is given to the performance of the Vertex-Morphing and In-plane regularization methods in large scale industrial problems.

In *Chapter 5*, the adjoint sensitivity analysis and the presented regularization methods are combined for the solution of a three-dimensional shape optimization problem of a flexible pipe. First, the computed gradients of the coupled problem are verified with finite differences. Furthermore, the importance of a coupled sensitivity analysis is outlined by comparing it with a sensitivity calculation in which the structure is considered to be rigid. As it will be shown, the sensitivities are dramatically different proving that performing shape optimization on a flexible pipe by considering it rigid might lead the optimizer even in different directions. In the sequence, the computed sensitivities are used in the shape optimization of the flexible part. Here, the regularization methods described in Chapter 3 are successfully applied.

In *Chapter 6* the overall summary of the work is given. Moreover, directions for further research are addressed.

Chapter 2

Sensitivity analysis

2.1 Introduction

In general, sensitivity analysis determines the impact that variations of the input parameters of a system have on its output parameters. Sensitivity analysis plays an important role in uncertainty quantification, error analysis, inverse and identification studies and numerical optimization. In the context of gradient-based shape optimization it is used to assess the gradients of the response functions i.e. objectives and constraints which drive the optimization.

For steady state problems the response function J depends on the design variable s and on the response variable w. In sequence, however, the response variable w depends on the design variable s as well, i.e. $J = J(s, w(s))$. In parallel, the system is governed by a set of differential equations $R(s, w(s)) = 0$, with $\mathbf{R}(\mathbf{s}, \mathbf{w}(\mathbf{s})) = \mathbf{0}$ being its discrete form, where now $\mathbf{s} = [s_1, .., s_n]$ and $\mathbf{w} = [w_1, .., w_m]$ are the vector of the design variable and state variable, respectively. n and m is the number of design and state variables, respectively. The goal of sensitivity analysis in this case is to determine the total derivative $dJ/d\mathbf{s}$.

A simple way to compute this derivative is with the finite difference approach. For instance, a first order approximation using this approach is

$$\frac{dJ}{ds_i} = \frac{J(s_i + \Delta s_i) - J(s_i)}{\Delta s_i} + o(\Delta s_i). \tag{2.1}$$

Resulting from a Taylor series expansion, the truncation error of this approximation is of order $o(\Delta s_i)$ and thus, the error decreases as the finite difference step Δs_i is decreasing. However, subtracting the floating point numbers $J(s_i + \Delta s_i)$ and $J(s_i)$ introduces a cancellation error which increases as the step Δs_i is decreasing. Consequently, there is an optimal step size that gives the least error. This fact should be considered during evaluation of gradients with finite differences and an additional effort should be made to estimate the optimal step size.

Furthermore, the calculation of $dJ/d\mathbf{s}$ using the forward finite difference scheme of equation (2.1) requires $n+1$ evaluations of the response function J which in turns involves $n+1$ solutions of the system $\mathbf{R}(\mathbf{s}, \mathbf{w}) = \mathbf{0}$. This is the main reason why finite differences are not preferred for problems with many design variables even though their implementation is straightforward and can be used even with black-box solvers.

In contrast to the finite difference method, analytical methods can offer more efficient and reliable solutions to the problem. Applying the chain rule, the total derivative of the response with respect to the design variable is

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \frac{\partial J}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{s}}. \tag{2.2}$$

Usually, the derivatives involving the response function J can be computed analytically. Furthermore, the derivative of the state variables with respect to the design variables, $d\mathbf{w}/d\mathbf{s}$ can be evaluated by applying the chain rule to the derivative of the state equation $\mathbf{R}(\mathbf{s}, \mathbf{w}(\mathbf{s})) = \mathbf{0}$

$$\frac{d\mathbf{R}}{d\mathbf{s}} = \frac{\partial \mathbf{R}}{\partial \mathbf{s}} + \frac{\partial \mathbf{R}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{s}} = \mathbf{0}. \tag{2.3}$$

Substituting equation (2.3) to (2.2) and assuming that $\partial \mathbf{R}/\partial \mathbf{w}$ is not singular results

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} - \frac{\partial J}{\partial \mathbf{w}} \frac{\partial \mathbf{R}}{\partial \mathbf{w}}^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{s}}. \tag{2.4}$$

In the above equation, $\partial \mathbf{R}/\partial \mathbf{w}$ is the Jacobian matrix of the problem. According to the sequence that the evaluations are performed in this equation two methods can be distinguished.

In the direct approach, the evaluations in equation (2.4) are carried out from right to left, which means that first the term

$$\frac{\partial \mathbf{R}}{\partial \mathbf{w}}^{-1} \frac{\partial \mathbf{R}}{\partial \mathbf{s}} \tag{2.5}$$

is determined as the solution of the following system

$$\frac{\partial \mathbf{R}}{\partial \mathbf{w}} \frac{d\mathbf{w}}{d\mathbf{s}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{s}}. \tag{2.6}$$

The right-hand side of this equation consists of n columns which means that n linear system solutions are required. After solution of this system and substitution of $d\mathbf{w}/d\mathbf{s}$ in equation (2.4), n vector multiplications of size m are needed.

In contrast, in the adjoint approach, the evaluations are carried out from left to right and the following, so called dual, system is solved

$$\frac{\partial \mathbf{R}}{\partial \mathbf{w}}^{T} \boldsymbol{\alpha} = -\frac{\partial J}{\partial \mathbf{w}}^{T}. \tag{2.7}$$

The parameter $\boldsymbol{\alpha}$ is the adjoint variable of the problem. As it can be observed from equation (2.7), the Jacobian of the dual problem is the transpose of the Jacobian of the original or primal problem while the right-hand side of the system is dependent on the number of response functions. In case that the sensitivity of more responses is needed, the system (2.7) needs to be solved as many times as the number of responses. Finally, the adjoint variable is substituted back into equation (2.4),

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \alpha^{T} \frac{\partial \mathbf{R}}{\partial \mathbf{s}}. \tag{2.8}$$

Here, n vector multiplications of size m are required.

From all the above it is clear that the choice between direct and adjoint approach depends on the number of design variables and responses.

In problems where the number of responses is more than the number of design variables the discrete approach is preferable. Otherwise the adjoint method is more efficient. In this work the adjoint method is preferred since the number of design variables is vastly more than the number of responses.

Alternatively, the adjoint variables can be identified as the Lagrange multipliers related to the state equations as equality constraints of an extended problem where the design and the state variables are treated independently.

Derivation of the adjoint equations can be done by one of the following approaches: In the discrete approach, the adjoint equation is derived using a discrete state equation, which is the formulation presented in this section. Contrary, in the continuous approach the adjoint equation is formulated upon the linearized primal continuous equation and then primal and adjoint equations are discretized separately. According to the application and the existing numerical implementation of the primal problem one or the other method can be preferred. Moreover, it is noteworthy that the discrete equation systems produced by the continuous and discrete approaches are identical only if the discretization schemes used for discretizing the continuous equations match the ones used for the primal discrete equation and the derived discrete adjoint equation. By all means, by refining the discretization regardless of the schemes, the two approaches tend to the same solution.

A review of the methods used for the sensitivity analysis for linear elastic structural problems can be found in [122] and the references therein. In these problems, the equation of equilibrium of the structure acts as an equality constraint and the response function depends on the structural displacements. Among the various continuous [17, 50] and discrete [121] methods, the discrete method of semi-analytic sensitivity analysis is one of the most favored because of its efficiency and simplicity in implementation. Within this method the derivatives of element data are approximated with finite differences which introduces a truncation and a cancellation error in the computation of elemental derivatives. These errors are amplified when these derivatives are used for further computations [6]. There exist various methods to eliminate these errors among which the most significant can be found in [11, 16, 77].

Concerning compressible and incompressible flow problems the basic formulation can be found in [46] and a review of the various adjoint implementations can be found in [115]. Unlike the structural problem, for the fluid problem there is an ongoing discussion on the advantages and disadvantages of the various discrete and continuous methods [91]. Furthermore, even though the adjoint equations where derived initially for the potential, Euler and compressible Navier-stokes equations [5, 64, 103] there is a great work done also for incompressible flows both on the field of continuous [96, 112, 132] and discrete [92, 93] adjoints.

In multidisciplinary sensitivity analysis the objective is to evaluate the gradient of a system's response which is governed by a set of coupled equations describing the interaction of the physical fields involved. The analytical formulation of such a problem is presented in [111]. Generally, in this field the bibliography is significantly less but there are indeed some interesting publications in the field of steady-state sensitivity analysis for a fluid-structure interaction problem. Among others one can refer to the work of Maute et al. [84, 85] who presented a three-field formulation for the adjoint problem similar to the solution of the primal problem presented in [30] using the discrete adjoint method for both the structure and the fluid field. Moreover, Martins et al. [82, 83] combined the continuous adjoint method on the fluid side with the discrete on the structural side giving emphasis on the construction of the different components of the Jacobian matrix of the problem. Fazzolari et al. in [33] presents the coupled problem in its continuous form and applies the coupled terms as boundary conditions to the continuous equations. An approximation of the fully nonlinear Jacobian of the coupled problem is shown in [78]. Finally, a one-dimensional demonstration of the time-dependent problem and the physical interpretation of the adjoint problem compared to the primal one is introduced in [24].

In the present work, the discrete formulation is used for the structural problem and the continuous formulation is used for the fluid one. A combination of the two approaches is applied for the sensitivity analysis of a coupled fluid-structure interaction problem. In what follows, the single-field adjoint sensitivity formulations for the structural and fluid problems are presented in Section 2.2. More precisely, in Section 2.2.1 the semi-analytic sensitivity analysis for a linear structural problem is described and in Section 2.2.2 the continuous adjoint equations for the steady-state incompressible Navier-Stokes are shown. Using these two single field adjoint equations the coupled adjoint sensitivity analysis is presented in Section 2.3.

2.2 Single-field adjoint sensitivity

In the following sections the single field sensitivity analysis formulation is presented for the structural and fluid problem, in Sections 2.2.1 and 2.2.2, respectively. This formulation is used in Section 2.3 to demonstrate the sensitivity analysis for a steady-state coupled fluid-structure interaction problem.

2.2.1 Structural adjoint sensitivity

Hereafter, the equilibrium equations for a linear elastic structure are presented in their continuous and discrete form. Based on the discrete form, the sensitivities for a general response function are derived using the adjoint method.

In what follows, a body of volume Ω_s with a smooth boundary Γ_s is considered. On this body a prescribed body force b_i per unit volume defined on Ω_s as well as a traction $\hat{t}_i$ on each boundary surface $\Gamma_{s_t} \subset \Gamma_s$ is applied. On the remaining part of the surface Γ_{s_u} a displacement $\hat{u}_i$ is prescribed. The displacement u_i on Ω_s that the body will exhibit under the above conditions is of interest.

Furthemore, let ϵ be the infinitesimal strain tensor defined as

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \tag{2.9}$$

and σ_{ij} be the stress tensor related to the strain tensor through the constitutive equation

$$\sigma_{ij} = E_{ijkl}\epsilon_{kl}. \tag{2.10}$$

The forth order tensor E_{ijkl} is the elasticity tensor satisfying major and minor symmetries as well as positive-definiteness.

The strong form of the boundary-value problem presented above is the following

$$\begin{aligned} \frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0 &\quad \text{in } \Omega_s \\ u_i = \hat{u}_i &\quad \text{on } \Gamma_{s_u} \\ \sigma_{ij} n_j = \hat{t}_i &\quad \text{on } \Gamma_{s_t} \end{aligned} \tag{2.11}$$

The derivation of the above equation can be found in [35, 60, 120]. The discrete form of the above problem is the following

$$\mathbf{R}(\mathbf{s}, \mathbf{u}) = \mathbf{K}(\mathbf{s})\mathbf{u}(\mathbf{s}) - \mathbf{f}(\mathbf{s}) = \mathbf{0}, \tag{2.12}$$

considering that boundary conditions are already incorporated into the system. Here, the state variable $\mathbf{w}$ is the structural displacement $\mathbf{u}$. The matrix $\mathbf{K}$ is the stiffness matrix of the problem and depends on the design variable $\mathbf{s}$. The vector $\mathbf{f}$ is the external loading defined on each degree of freedom and depends also in many cases on the design variable $\mathbf{s}$.

The response function in such a case depends also on the structural displacements,

$$J = J(\mathbf{s}, \mathbf{u}(\mathbf{s})) \tag{2.13}$$

and for instance it can be the strain energy, the displacement or the stress at a point, the mass or the volume of the structure or the eigenfrequency.

According to equation (2.7) the adjoint system under these considerations is

$$\mathbf{K}^T \boldsymbol{\alpha} = -\frac{\partial J}{\partial \mathbf{u}}^T. \tag{2.14}$$

After evaluation of the adjoint displacements, equation (2.8) gives the total gradient of the response

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \alpha^T \mathbf{P}^*, \tag{2.15}$$

where

$$\mathbf{P}^* = \frac{\partial \mathbf{K}}{\partial \mathbf{s}} \mathbf{u} - \frac{\partial \mathbf{f}}{\partial \mathbf{s}} \tag{2.16}$$

is the Pseudo load “vector”. Each column of this matrix represents a pseudo load case defined for each design variable $s_i, i = 1 \ldots n$.

Consequently, evaluation of the total derivative of the response funtion requires solution of the primal equation (2.12), solution of the adjoint equation (2.14) and evaluation of (2.15). Since the response function is given usually analytically the partial derivatives of the response function can be computed straightforward. The involved parts to be evaluated are the terms $\partial\mathbf{K}/\partial\mathbf{s}$ and $\partial\mathbf{f}/\partial\mathbf{s}$. Specially, the term $\partial\mathbf{K}/\partial\mathbf{s}$ requires an assembly of the stiffness matrix for each design variable. In semi-analytic sensitivity analysis these quantities are computed with finite differences on the element level and such a costly assembly is avoided. For instance, for the evaluation of the derivative of the stiffness matrix $\mathbf{K}$ with respect to the design variable s_i a forward finite difference scheme could be used

$$\frac{\partial\mathbf{K}}{\partial s_i} \approx \frac{\mathbf{K}(\mathbf{u}, s_i + \Delta s) - \mathbf{K}(\mathbf{u}, s_i)}{\Delta s_i}. \tag{2.17}$$

The semi-analytic method overcomes the element specific cumbersome derivations required for the evaluation of the derivative of the stiffness matrix. This makes the sensitivity evaluation robust and independent of the element formulation. However, this analysis shows high truncation errors when applied to structural problems modeled with finite elements [6]. The exact semi-analytic sensitivity analysis overcomes these accuracy problems by introducing correction factors based on product spaces of rigid body rotation vectors [11].

2.2.2 Fluid adjoint sensitivity

In what follows, the derivation of the continuous adjoint Navier-Stokes equations and the formulation of the gradients with respect to the design shape are presented. The derivation of the adjoint equations follows the one by Othmer [96, 98], Stück [115] and Soto and Löhner [112, 113].

After presenting the primal equations and the response function in their integral form, the derivation is done by applying the augmented Lagrangian method. Using this method, the adjoint problem of the Navier-Stokes equations is formulated and respective boundary conditions are derived in the subsequent sections. Finally, a surface formulation based on [96] for computing shape design gradients, without a dependence on volume mesh sensitivities (which is one of the main advantages of the continuous adjoint method over the discrete one) is covered in the last section.

Navier-Stokes equation and response function

The steady incompressible Navies-Stokes state equations together with the conservation of mass describe the motion of a Newtonian incompressible fluid as follows

$$\begin{aligned}\boldsymbol{R_v} &= \boldsymbol{v} \cdot \nabla \boldsymbol{v} + \nabla p - \nabla \cdot 2\nu\varepsilon(\boldsymbol{v}) = \mathbf{0} \\ R_p &= \nabla \cdot \boldsymbol{v} = 0,\end{aligned} \tag{2.18}$$

with the first and second equation being the momentum and continuity equation, respectively. Variable $\boldsymbol{v}$ is the fluid velocity, p denotes the density-normalized pressure with units kg m^{-1} s^{-2}/kg $\mathrm{m}^{-3} = \mathrm{m}^2/\mathrm{s}^2$, ν the kinematic viscosity and

$$\varepsilon(\boldsymbol{v}) = \frac{1}{2}(\nabla \boldsymbol{v} + (\nabla \boldsymbol{v})^T) \tag{2.19}$$

the symmetric strain rate tensor.

Equation (2.18) is nonlinear because of the convective term $\boldsymbol{v} \cdot \nabla \boldsymbol{v}$ of the momentum equation. Moreover, there is no explicit transport equation for the pressure which appears only on the momentum equation which is a transport equation for the velocity. An idea to overcome this problem is to find an equation for the pressure which makes the velocity field solenoidal. For this, an additional equation which links the pressure to the velocities is required. This equation is the Poisson equation for the pressure and is solved iteratively together with the momentum equation. In this way the velocity field is projected through the Poisson equation in such a way which satisfies continuity. Among the various projection methods, the SIMPLE algorithm is applied in this work and it is described in detail in Appendix B.1.

The response functional depends on the state variables $(\boldsymbol{v}, p)$ and on the design variable s. It is integrated over the fluid domain Ω and the boundary $\Gamma = \partial\Omega$ as follows

$$\begin{aligned}J = J(s, \boldsymbol{v}, p) &= J_\Gamma(s, \boldsymbol{v}, p) + J_\Omega(s, \boldsymbol{v}, p) \\ &= \int_\Gamma j_\Gamma d\Gamma + \int_\Omega j_\Omega d\Omega.\end{aligned} \tag{2.20}$$

A typical response function used for internal flow problems is the power loss or power dissipation function described in [96] as

$$J := -\int_{\Gamma} (p + \frac{1}{2}v^2)\boldsymbol{v} \cdot \boldsymbol{n}, \tag{2.21}$$

where $\boldsymbol{n}$ is the outward normal vector to the boundary Γ.

A force component like drag or lift can be also defined as response function and is presented in [113] as

$$J := -\int_{S} \boldsymbol{n} \cdot (2\nu\varepsilon(\boldsymbol{v}) - p\boldsymbol{I}) \cdot \boldsymbol{d}, \tag{2.22}$$

where $\boldsymbol{d}$ is the direction of the force and S is the surface that the force is applied. By definition these response functions are defined on the boundary of the fluid domain which as it is shown in the sequel, results only in contributions to the boundary conditions of the adjoint equations. The flow uniformity at the outlet and a target velocity distribution can be further examples of response functions.

The Lagrange method

In order to calculate the sensitivities of the response function with respect to the design variables, the Lagrange method is applied to the continuous problem presented in the previous section. Initially, the response function is augmented as follows

$$L = J + \int_{\Omega} \boldsymbol{\Psi} \cdot \boldsymbol{R}. \tag{2.23}$$

The variable $\boldsymbol{\Psi} = (\boldsymbol{\psi_v}, \psi_p)$ is the continuous adjoint variable of the problem which consists of the adjoint velocity $\boldsymbol{\psi_v}$ and the adjoint pressure ψ_p while $\boldsymbol{R} = (\boldsymbol{R_v}, R_p)$ is the vector of the momentum and continuity equations.

Variation of $L = L(\boldsymbol{s}, \boldsymbol{v}, p)$ yields to

$$\begin{aligned} \delta L &= \delta L_{\boldsymbol{s}} + \delta L_{\boldsymbol{v}} + \delta L_p \\ &= (\frac{\partial J}{\partial \boldsymbol{s}} + \int_{\Omega} \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{s}})\delta \boldsymbol{s} \\ &+ (\frac{\partial J}{\partial \boldsymbol{v}} + \int_{\Omega} \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{v}})\delta \boldsymbol{v} + (\frac{\partial J}{\partial p} + \int_{\Omega} \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial p})\delta p. \end{aligned} \tag{2.24}$$

Choosing $\boldsymbol{\Psi} = (\boldsymbol{\psi_v}, \psi_p)$ such that

$$\begin{aligned}\delta L_{\boldsymbol{v}} + \delta L_p = (&\frac{\partial J}{\partial \boldsymbol{v}} + \int_\Omega \boldsymbol{\psi_v} \cdot \frac{\partial \boldsymbol{R_v}}{\partial \boldsymbol{v}} + \int_\Omega \psi_p \cdot \frac{\partial R_p}{\partial \boldsymbol{v}})\delta \boldsymbol{v} \\ + (&\frac{\partial J}{\partial p} + \int_\Omega \boldsymbol{\psi_v} \cdot \frac{\partial \boldsymbol{R_v}}{\partial p} + \int_\Omega \psi_p \cdot \frac{\partial R_p}{\partial p})\delta p = 0,\end{aligned} \tag{2.25}$$

the sensitivity of the response functional with respect to the design variable can be computed as

$$\delta L = \frac{\partial J}{\partial \boldsymbol{s}} \delta \boldsymbol{s} + \int_\Omega \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{s}} \delta \boldsymbol{s}. \tag{2.26}$$

Equation (2.25) initiates the continuous adjoint problem and its respective boundary conditions. Computing the adjoint variables based on (2.25), the surface sensitivities of equation (2.26) can be evaluated.

Variation of the Navier-Stokes equations

For setting up the adjoint problem of equation (2.25) the variation of the Navier-Stokes equations with respect to the state variables is required. Taking the variation of the momentum and continuity equations with respect to the velocity and pressure yields respectively

$$\begin{aligned}\delta_{\boldsymbol{v}} \boldsymbol{R_v} &= \delta \boldsymbol{v} \cdot \nabla \boldsymbol{v} + \boldsymbol{v} \cdot \nabla \delta \boldsymbol{v} - \nabla \cdot 2\nu\varepsilon(\delta \boldsymbol{v}), \\ \delta_{\boldsymbol{v}} R_p &= \nabla \cdot \delta \boldsymbol{v}, \\ \delta_p \boldsymbol{R_v} &= \nabla \delta p, \\ \delta_p R_p &= 0.\end{aligned} \tag{2.27}$$

Here, the variation of the kinematic viscosity ν is neglected. This is indeed true as the viscosity of the fluid in certain temperatures remains constant. However, in turbulence modeling based on the eddy viscosity assumption ν represents both the molecular viscosity and the sub-grid turbulence effects. In that case, assuming the variation of the kinematic viscosity equal to zero brings inconsistency to the adjoint flow problem compared to the primal one. This assumption is known as the frozen turbulence assumption.

Adjoint Navier-Stokes equations

Substituting the variations of the Navier-Stokes equations of the previous section to the adjoint problem of equation (2.25) yields to

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{v}}\delta \boldsymbol{v} + \int_\Omega \boldsymbol{\psi_v} \cdot (\delta \boldsymbol{v} \cdot \nabla \boldsymbol{v}) + \int_\Omega \boldsymbol{\psi_v} \cdot (\boldsymbol{v} \cdot \nabla \delta \boldsymbol{v}) \\ &- \int_\Omega \boldsymbol{\psi_v} \cdot (\nabla \cdot 2\nu\varepsilon(\delta \boldsymbol{v})) + \int_\Omega \psi_p \nabla \cdot \delta \boldsymbol{v} \\ &+ \frac{\partial J}{\partial p}\delta p + \int_\Omega \boldsymbol{\psi_v} \cdot \nabla \delta p = 0.\end{aligned} \tag{2.28}$$

In order to separate the variations, integration by parts and the divergence theorem which is described in Appendix A is applied on each integral term of equation (2.28) as follows

$$\int_\Omega \boldsymbol{\psi_u} \cdot (\delta \boldsymbol{v} \cdot \nabla \boldsymbol{v}) = \int_\Gamma (\delta \boldsymbol{v} \cdot \boldsymbol{n})(\boldsymbol{\psi_v} \cdot \boldsymbol{v}) - \int_\Omega \delta \boldsymbol{v} \cdot (\nabla \boldsymbol{\psi_v} \cdot \boldsymbol{v}), \tag{2.29a}$$

$$\int_\Omega \boldsymbol{\psi_v} \cdot (\boldsymbol{v} \cdot \nabla \delta \boldsymbol{v}) = \int_\Gamma (\boldsymbol{v} \cdot \boldsymbol{n})(\delta \boldsymbol{v} \cdot \boldsymbol{\psi_v}) - \int_\Omega (\boldsymbol{v} \cdot \nabla \boldsymbol{\psi_v}) \cdot \delta \boldsymbol{v}, \tag{2.29b}$$

$$\begin{aligned}\int_\Omega \boldsymbol{\psi_v} \cdot (\nabla \cdot 2\nu\varepsilon(\delta \boldsymbol{v})) = &\int_\Gamma 2\nu \boldsymbol{\psi_v} \cdot \varepsilon(\delta \boldsymbol{v}) \cdot \boldsymbol{n} - \int_\Gamma (2\nu \boldsymbol{n} \cdot \varepsilon(\boldsymbol{\psi_v})\delta \boldsymbol{v} \\ &+ \int_\Omega \delta \boldsymbol{v}(\nabla \cdot (2\nu\varepsilon(\boldsymbol{\psi_v}))),\end{aligned} \tag{2.29c}$$

$$\int_\Omega \psi_p \nabla \cdot \delta \boldsymbol{v} = \int_\Gamma \delta \boldsymbol{v} \cdot \psi_p \boldsymbol{n} - \int_\Omega \delta \boldsymbol{v} \cdot \nabla \psi_p, \tag{2.29d}$$

$$\int_\Omega \boldsymbol{\psi_v} \cdot \nabla \delta p = \int_\Gamma (\boldsymbol{\psi_v} \cdot \boldsymbol{n})\delta p - \int_\Omega \delta p \nabla \cdot \boldsymbol{\psi_v}. \tag{2.29e}$$

The first integration by part presented in equation (2.29a) is not really crucial since the term depends only on the variation of the velocity $\delta \boldsymbol{v}$ and not on any gradient of it. This idea was successfully applied by Stück [115]. For the integration by parts of this term the relation $\nabla \cdot \delta \boldsymbol{v} = 0$, derived in the previous section, is applied and for the relation in (2.29b) the continuity equation $\nabla \cdot \boldsymbol{v} = 0$ is used. It should be noted that $(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = \boldsymbol{v} \cdot \nabla \boldsymbol{v}$.

Substituting (2.29) to (2.28) and separating the domain integrals from the boundary integrals results in

$$
\begin{aligned}
&\int_\Omega (-\nabla \boldsymbol{\psi_v} \cdot \boldsymbol{v} - \boldsymbol{v} \cdot \nabla \boldsymbol{\psi_v} - \nabla \cdot 2\nu\varepsilon(\boldsymbol{\psi_v}) - \nabla\psi_p + \frac{\partial j_\Omega}{\partial \boldsymbol{v}}) \cdot \delta\boldsymbol{v} \\
&+ \int_\Omega (-\nabla \cdot \boldsymbol{\psi_v} + \frac{\partial j_\Omega}{\partial p})\delta p \\
&+ \int_\Gamma (\boldsymbol{n}(\boldsymbol{\psi_v} \cdot \boldsymbol{v}) + (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{\psi_v} + 2\nu\boldsymbol{n} \cdot \varepsilon(\boldsymbol{\psi_v}) + \psi_p \boldsymbol{n} + \frac{\partial j_\Gamma}{\partial \boldsymbol{v}}) \cdot \delta\boldsymbol{v} \\
&- \int_\Gamma 2\nu\boldsymbol{\psi_v} \cdot \varepsilon(\delta\boldsymbol{v}) \cdot \boldsymbol{n} \\
&+ \int_\Gamma (\boldsymbol{\psi_v} \cdot \boldsymbol{n} + \frac{j_\Gamma}{\partial p})\delta p = 0.
\end{aligned}
\tag{2.30}
$$

This equation has to be valid for every $\delta\boldsymbol{v}$ and δp. Consequently, the domain integrals give the adjoint equations of the steady-state incompressible Navier-Stokes for a general response function

$$
\begin{aligned}
-\nabla \boldsymbol{\psi_v} \cdot \boldsymbol{v} - \boldsymbol{v} \cdot \nabla \boldsymbol{\psi_v} &= \nabla \cdot 2\nu\varepsilon(\boldsymbol{\psi_v}) + \nabla\psi_p - \frac{\partial j_\Omega}{\partial \boldsymbol{v}} \\
\nabla \cdot \boldsymbol{\psi_v} &= \frac{\partial j_\Omega}{\partial p}\delta p,
\end{aligned}
\tag{2.31}
$$

and from the boundary integrals the adjoint boundary conditions are derived. In the special case that the response function is only a surface integral over the domain boundary, the adjoint equation is independent of the response. Then for evaluation of the adjoint variables of several response functions only a modification in the boundary conditions is needed.

The derived adjoint system is linear and has a similar structure to the primal state. According to Löhner [113] the term $\boldsymbol{v} \cdot \nabla \boldsymbol{\psi_v}$ can be neglected resulting to a first order linearization of the Navier-Stokes problem. Then the adjoint problem has the same convection-diffusion behavior as the primal one with the difference that the adjoint velocity is convected upstream compared to the primal flow. Even though the adjoint problem is linear, the problem caused by the lack of an explicit transport equation for the pressure remains, as already described in Section 2.2.2. For this reason, a SIMPLE-based algorithm is used again for the solution of the adjoint problem. The procedure is described in detail in Appendix B.2.

Adjoint Boundary conditions

In this section the adjoint boundary conditions will be shown for the power loss and the total force at a specific direction. By definition, these response functions are defined on the boundary of the fluid domain which results only in contributions to the boundary conditions of the adjoint equations as described before.

In the case of the power loss response function (2.21), the variation with respect to the flow variables is

$$\frac{\partial J}{\partial \boldsymbol{v}}\delta \boldsymbol{v} + \frac{\partial J}{\partial p}\delta p = -\int_\Gamma [(p + \frac{1}{2}v^2)\boldsymbol{n} - (\boldsymbol{v}\cdot\boldsymbol{n})\boldsymbol{v}]\delta\boldsymbol{v} - \int_\Gamma \boldsymbol{v}\cdot\boldsymbol{n}\delta p \tag{2.32}$$

and if the response function is the force at a defined direction (2.22) then

$$\frac{\partial J}{\partial \boldsymbol{v}}\delta \boldsymbol{v} + \frac{\partial J}{\partial p}\delta p = -\int_S \boldsymbol{n}\cdot 2\nu\varepsilon(\delta\boldsymbol{v})\cdot\boldsymbol{d} + \int_S \boldsymbol{n}\cdot\boldsymbol{d}\delta p. \tag{2.33}$$

Furthermore, the boundary terms of equation (2.30) from which the boundary conditions are derived are the following

$$\begin{aligned}\int_\Gamma \big[&(\boldsymbol{\psi_v}\cdot\boldsymbol{v})(\boldsymbol{n}\cdot\delta\boldsymbol{v}) + (\delta\boldsymbol{v}\cdot\boldsymbol{\psi_v})(\boldsymbol{v}\cdot\boldsymbol{n}) + 2\nu(\delta\boldsymbol{v}\cdot\varepsilon(\boldsymbol{\psi_v}))\cdot\boldsymbol{n}\\ &- 2\nu\boldsymbol{\psi_v}\cdot\varepsilon(\delta\boldsymbol{v})\cdot\boldsymbol{n} + \psi_p\delta\boldsymbol{v}\cdot\boldsymbol{n} + \boldsymbol{\psi_v}\delta p\cdot\boldsymbol{n}\\ &+ \frac{\partial j_\Gamma}{\partial\boldsymbol{v}}\cdot\delta\boldsymbol{v} + \frac{j_\Gamma}{\partial p}\delta p\big].\end{aligned} \tag{2.34}$$

At the inlet Γ_{in} and wall Γ_w boundaries the velocity $\boldsymbol{v}$ is prescribed and consequently $\delta\boldsymbol{v} = \boldsymbol{0}$. Moreover a zero-gradient boundary condition is applied for the pressure p. Concerning the force response functional, it is enough to set

$$\boldsymbol{\psi_v} = \boldsymbol{0} \text{ on } \Gamma_{in,w}\setminus S \tag{2.35}$$

and

$$\boldsymbol{\psi_v} = -\boldsymbol{d} \text{ on } S. \tag{2.36}$$

In the case of the power loss response function as shown in [96], it holds that

$$\boldsymbol{\psi_v} = \boldsymbol{v} \text{ on } \Gamma_{in,w}. \tag{2.37}$$

At the outlet Γ_{out}, for the primal equations, the pressure and the normal gradient of the velocity are set to zero. For the force in a specific direction response setting

$$\boldsymbol{\psi}_{\boldsymbol{v}} = \mathbf{0} \text{ on } \Gamma_{out} \tag{2.38}$$

fulfills the boundary conditions of equation (2.34). Concerning the power dissipation, according to Othmer [96] by decomposition of (2.34) into tangential and normal components the boundary condition is as follows

$$\begin{aligned} \psi_p &= \boldsymbol{\psi}_{\boldsymbol{v}} \cdot \boldsymbol{v} + (\boldsymbol{\psi}_{\boldsymbol{v}} \cdot \boldsymbol{n})(\boldsymbol{v} \cdot \boldsymbol{n}) + \nu(\boldsymbol{n} \cdot \nabla)(\boldsymbol{\psi}_{\boldsymbol{v}} \cdot \boldsymbol{n}) - \frac{1}{2}\boldsymbol{v}^2 - \frac{1}{2}(\boldsymbol{v} \cdot \boldsymbol{n})^2, \\ 0 &= (\boldsymbol{v} \cdot \boldsymbol{n})(\boldsymbol{\psi}_{\boldsymbol{v}t} - \boldsymbol{v}_t) + \nu(\boldsymbol{n} \cdot \nabla)\boldsymbol{\psi}_{\boldsymbol{v}t}, \end{aligned} \tag{2.39}$$

where $\boldsymbol{v}_t$ and $\boldsymbol{\psi}_{\boldsymbol{v}t}$ are the tangential components of the velocity and the adjoint velocity, respectively.

Adjoint Surface Sensitivities

After computing the adjoint fields $\boldsymbol{\psi}_{\boldsymbol{v}}$ and ψ_p, the sensitivity of the response function can be evaluated from equation (2.26). This evaluation requires the variation of the Navier-Stokes equation with respect to the design variable $\boldsymbol{s}$ which is related to the variation of the Navier-Stokes equations with respect to the state variables through the total variation:

$$\delta \boldsymbol{R} = \frac{\partial R}{\partial \boldsymbol{s}}\delta \boldsymbol{s} + \frac{\partial R}{\partial \boldsymbol{v}}\delta \boldsymbol{v} + \frac{\partial R}{\partial p}\delta p = \mathbf{0}. \tag{2.40}$$

The last two terms of the above equation appear in equation (2.25) which after the integration by parts presented in (2.29) gives equation (2.30). So, substituting these two terms in (2.26) and applying the same ideas, the sensitivity equation is

$$\begin{aligned} \delta L &= \frac{\partial J}{\partial \boldsymbol{s}}\delta \boldsymbol{s} - \int_\Omega \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{v}}\delta \boldsymbol{v} - \int_\Omega \boldsymbol{\Psi} \cdot \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{p}}\delta \boldsymbol{p} \\ &= \frac{\partial J}{\partial \boldsymbol{s}}\delta \boldsymbol{s} - \int_\Gamma (\boldsymbol{n}(\boldsymbol{\psi}_{\boldsymbol{v}} \cdot \boldsymbol{v}) + (\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{\psi}_{\boldsymbol{v}} + 2\nu \boldsymbol{n} \cdot \varepsilon(\boldsymbol{\psi}_{\boldsymbol{v}}) + \psi_p \boldsymbol{n}) \cdot \delta \boldsymbol{v} \\ &+ \int_\Gamma 2\nu \boldsymbol{\psi}_{\boldsymbol{v}} \cdot \varepsilon(\delta \boldsymbol{v}) \cdot \boldsymbol{n} - \int_\Gamma \boldsymbol{\psi}_{\boldsymbol{v}} \cdot \boldsymbol{n} \delta p = 0. \end{aligned} \tag{2.41}$$

The volume integrals disappear because the adjoint fields satisfy the adjoint Navier-Stokes equations of (2.31) with the assumption that the objective function is defined only on the fluid boundary. However, the variations of $\boldsymbol{v}$ and p need to be expressed as a variation of the design variable $\boldsymbol{s}$. Using a first order Taylor series expansion it can be shown [59, 89] that

$$\delta \boldsymbol{v} = -\boldsymbol{n}\nabla \boldsymbol{v}\delta s \text{ and } \delta p = -\boldsymbol{n}\nabla p \delta s. \tag{2.42}$$

Substituting (2.42) into (2.41) results to an expression with boundary integral depending on the variation δs. Since this variation is zero everywhere except on the wall, the boundary integrals of (2.41) restrict on this boundary. Furthermore, $\boldsymbol{\psi_v} = \boldsymbol{0}$ because of the boundary conditions derived in the previous section. All the above lead to the following expression for the sensitivity

$$\frac{\delta L}{\delta s} = \frac{\partial J}{\partial s}\delta s - \int_{\Gamma_w} (2\nu \boldsymbol{n} \cdot \varepsilon(\boldsymbol{\psi_v}) + \psi_p \boldsymbol{n}) \cdot (\boldsymbol{n} \cdot \nabla)\boldsymbol{v}. \tag{2.43}$$

Further simplifications detailed in [96] lead to the final expression for the surface sensitivities

$$\frac{\partial L}{\partial s} = -A\nu(\boldsymbol{n} \cdot \nabla)\boldsymbol{\psi_{v_t}} \cdot (\boldsymbol{n} \cdot \nabla)\boldsymbol{v_t}, \tag{2.44}$$

with A the area influenced by the movement of the design variable s.

2.3 Coupled adjoint sensitivity

In this section the sensitivity analysis for a fluid-structure interaction problem is presented and the respective formulation is derived. First the primal problem is discussed and then its adjoint form is shown. Finally, the evaluation of the coupled sensitivities is presented.

2.3.1 Steady-state fluid-structure interaction

When a steady incompressible Newtonian fluid defined in Ω_f interacts with an elastic structure defined in Ω_s, the equilibrium equations of both fields should be satisfied, i.e.

$$\mathbf{R}(\mathrm{s}, \mathbf{w}(\mathrm{s})) = \begin{cases} \mathbf{S}(\mathrm{s}, \mathbf{w}(\mathrm{s})) = \mathbf{0} \\ \mathbf{F}(\mathrm{s}, \mathbf{w}(\mathrm{s})) = \mathbf{0} \end{cases}, \tag{2.45}$$

where $\mathbf{S}(\mathbf{s}, \mathbf{w}(\mathbf{s})) = \mathbf{0}$ is the structural response of equation (2.12) and $\mathbf{F}(\mathbf{s}, \mathbf{w}(\mathbf{s})) = (\mathbf{M}(\mathbf{s}, \mathbf{w}(\mathbf{s})), \mathbf{C}(\mathbf{s}, \mathbf{w}(\mathbf{s})))$ are the momentum and continuity equations (2.18) represented here in their discrete form. The fluid problem in each equilibrium state should be set in the unknown domain defined by the displacement of the deformed structure. The state variable $\mathbf{w} = (\mathbf{u}, \mathbf{v}, \mathbf{p})$ of such a problem is the vector structural displacements as well as the fluid velocities and pressures. The boundary conditions of such a problem are inherited from the single field problems at $\partial\Omega_s$ and $\partial\Omega_f$, respectively with the additional condition that

$$\boldsymbol{\sigma_s} \cdot \boldsymbol{n} = -\boldsymbol{\sigma_f} \cdot \boldsymbol{n} \quad \text{on} \quad \partial\Omega_s \cap \partial\Omega_f. \tag{2.46}$$

For a mathematical definition of the steay-state fluid structure interaction problem as well as the existence of a smooth solution the reader can refer to [49].

In such a problem the response function J can be fluid or structural related, so $J = J(\mathbf{s}, \mathbf{w}(\mathbf{s}))$. The problem is nonlinear and it is often solved with the Newton-Raphson method. So, linearization of equation (2.45) around an equilibrium point or the previous iteration k results to a linear system of equations

$$\begin{bmatrix} \dfrac{\partial \mathbf{F}}{\partial(\mathbf{v}, \mathbf{p})} & \dfrac{\partial \mathbf{F}}{\partial \mathbf{u}} \\ \dfrac{\partial \mathbf{S}}{\partial(\mathbf{v}, \mathbf{p})} & \dfrac{\partial \mathbf{S}}{\partial \mathbf{u}} \end{bmatrix} \begin{bmatrix} \Delta(\mathbf{v}, \mathbf{p})_k^T \\ \\ \Delta \mathbf{u}_k \end{bmatrix} = - \begin{bmatrix} \mathbf{F}(\mathbf{v}_k, \mathbf{p}_k, \mathbf{u}_k) \\ \\ \mathbf{S}(\mathbf{v}_k, \mathbf{p}_k, \mathbf{u}_k) \end{bmatrix}, \tag{2.47}$$

with $\Delta(\mathbf{v}, \mathbf{p})_k = (\mathbf{v}, \mathbf{p})_{k+1} - (\mathbf{v}, \mathbf{p})_k$ and $\Delta \mathbf{u}_k = \mathbf{u}_{k+1} - \mathbf{u}_k$. The matrices in the main diagonal of equation (2.47) are the Jacobian matrices of the single-field problems and the off-diagonal terms are responsible for the coupling of the two fields, evaluated at iteration k.

The above problem can be solved in a monolithic manner as one system of nonlinear equations [41, 44, 53, 57, 117, 118] or in a partitioned way by splitting it into two subsystems which interact with each other [30, 37, 73, 76, 101, 102, 124, 131].

When the problem is solved in a partitioned manner and there is a significant interaction of the two fields, the Gauss-Seidel loop between the two fields converges slowly and in many cases does not converge at all. For this problem many algorithms have been developed which stabilize and accelerate the problem. The Aitken method was one of the first methods developed for this reason [61, 90] and among all, the interface GMRES and the interface quasi-Newton methods [22, 23] have shown great performance in partitioned problems using black-box field solvers.

Another aspect in the fluid-structure interaction problem is the update of the fluid mesh during the fluid-structure interaction simulations. Usually, during an FSI simulation the structure is described in a Lagrangian way while the fluid is described using the Eulerian description. In the Langrangian description the node of the computational domain is linked to a material point during the computation while in the Eulerian description the material particles are moving through a fixed computational grid. When deformations of the common boundary of the two fields occur like in a fluid-structure interaction problem the disretizations of both fields should be adapted in a way which retains the quality of the computation. Usually this adaptation is essential in the fluid fixed Eulerian grid, which mesh is usually more prone to distortions. The Arbitrary Lagrangian-Eulerian (ALE) formulation combines the Lagrangian and the Eulerian formulation by keeping their best features [25, 55] and allows the mesh to move arbitrarily with its own velocity. However, the ALE description requires a mesh-update procedure. The various update procedures that are commonly used are discussed in Section 3.4.3 of the next chapter.

Nevertheless, the mesh update procedures are in general limited to relatively small shape changes and result to meshes with bad element quality when the deformations or rotations of the structural boundary are getting large. In these cases the fixed-grid methods appear to give more reliable results. In this class of problems the fluid remains fixed while the deforming structural boundary is not conforming the fluid domain. The immersed boundary method [28, 87, 100] and the fictitious domain method [47] are the most prominent among this class of problems. There are also applications which combine the ALE method with the fixed-grid approach [127].

In this work, the fluid-structure interaction problem is solved in a partitioned way. Moreover, since the problem is steady, the fluid-structure interaction boundary has zero velocity and the ALE adaptation in the

Navier-Stokes equations is not required. However, the methods used for the mesh update in ALE computations are applied in order to correct the mesh in each nonlinear fluid-structure interaction iteration.

2.3.2 An adjoint fluid-structure interaction

In this section the adjoint system of the aforementioned steady fluid-structure interaction problem is derived. The primal variables $\mathbf{w} = (\mathbf{v}, \mathbf{p}, \mathbf{u})$ are the velocities and pressures of the fluid and the structural displacements. The response function J of such a problem can be fluid and/or structure related, so $J = J(\mathbf{s}, \mathbf{w}(\mathbf{s}))$. According to (2.7), the adjoint system of such a problem is

$$\begin{bmatrix} \dfrac{\partial \mathbf{F}}{\partial (\mathbf{v}, \mathbf{p})} & \dfrac{\partial \mathbf{F}}{\partial \mathbf{u}} \\ \dfrac{\partial \mathbf{S}}{\partial (\mathbf{v}, \mathbf{p})} & \dfrac{\partial \mathbf{S}}{\partial \mathbf{u}} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})} \\ \\ \boldsymbol{\psi}_{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial J}{\partial (\mathbf{v}, \mathbf{p})} \\ \dfrac{\partial J}{\partial \mathbf{u}} \end{bmatrix}. \tag{2.48}$$

The unkowns of the above problem are the vector of adjoint velocity and pressure $\boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})}$ and the adjoint displacement $\boldsymbol{\psi}_{\mathbf{u}}$. The right hand side consists of the partial derivatives of the response function with respect to the state variable $\mathbf{w}$ and for the individual fields its evaluation can be done analytically or with finite differences as described in Sections 2.2.1 and 2.2.2.

Concerning the Jacobian matrix of the problem, its diagonal components $\partial \mathbf{F}/\partial(\mathbf{v}, \mathbf{p})$ and $\partial \mathbf{S}/\partial \mathbf{u}$ are the Jacobian matrices of the single fields. Assuming that the fluid flow is as described in Section 2.58 and is evaluated with finite volumes with one unknown per cell for the pressure and one for each velocity component, the matrix $\partial \mathbf{F}/\partial(\mathbf{v}, \mathbf{p})$ is of size $4N_c$ where N_c is the number of cells. The structural problem is solved with finite elements and its stiffness matrix is of size N_{dofs} which is the number of degrees of freedom per node times the number of nodes. The off-diagonal components of the Jacobian matrix are the coupling components of the system showing how the residual equation of a single field is changing by changing the state of the other field. In the sequence, these coupling terms are discussed and the way to evaluate them is described.

The change of the structural residual at each degree of freedom of the structure by varying the velocity and pressure of the flow is described by the matrix $\partial\mathbf{S}/\partial(\mathbf{v},\mathbf{p})$ which is of size $N_{dofs} \times 4N_c$. Since a change of the velocity field does not have a direct effect on the structural residual $\mathbf{S} = \mathbf{K}(\mathbf{s})\mathbf{u}(\mathbf{s}) - \mathbf{f}(\mathbf{s},\mathbf{p})$, there is no direct dependence of $\mathbf{S}$ on the fluid velocities. So,

$$\frac{\partial\mathbf{S}}{\partial\mathbf{v}} = \mathbf{0}. \tag{2.49}$$

Considering that the stiffness matrix and the displacement vector do not relate to a change of the pressure field, the only direct dependence of the structural equation to the fluid state variables is through the force vector $\mathbf{f}(\mathbf{s},\mathbf{p})$, representing the total forces applied on the structure. The total force is the sum of all external forces applied on the structure, like self weight and the forces applied from the fluid. The later ones are evaluated by interpolating the surface pressure field exerted by the fluid on the structural nodes and this interpolation rule gives the direct relation between forces and pressures. Numerically, taking a zero-gradient boundary condition for the pressure on the design surface, only the pressures of the adjacent cells to the design surface have non-zero interpolation coefficients and contribute to the interpolation of their own nodes. In case of non-conforming meshes the interpolation rule relating the two meshes has also to be taken into account.

For the simple 2-dimensional case of figure 2.1 the force $\mathbf{f_3}$ can be computed by the following interpolation scheme

$$\mathbf{f}_3 = \frac{1}{2}\mathbf{A_2}\mathrm{p}_2 + \frac{1}{2}\mathbf{A_3}\mathrm{p}_3, \tag{2.50}$$

then the partial derivative of $\mathbf{f_3}$ with respect to p_2 would simply be $1/2\mathbf{A_2}$.

As a result, by separating the domain velocities and pressures from the boundary ones and the boundary degrees of freedom of the structure from the volume ones, the matrix $\partial\mathbf{S}/\partial(\mathbf{v},\mathbf{p})$ can be further simplified

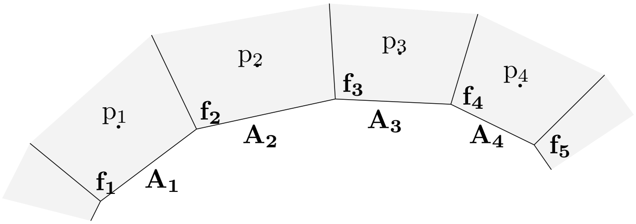


Figure 2.1: A simple 2-dimensional fluid mesh.

as

$$\frac{\partial \mathbf{S}}{\partial(\mathbf{v},\mathbf{p})} = \begin{array}{c|cc} & 4N_{c\Omega} & 4N_{c\Gamma} \\ \hline N_{dofs\Gamma} & \mathbf{0} & \mathbf{C}_S \\ N_{dofs\Omega} & \mathbf{0} & \mathbf{0} \end{array} = \begin{array}{cc} 4N_{c\Omega} & 4N_{c\Gamma} \\ \hline \mathbf{0} & [\mathbf{0} \quad \mathbf{C}_{S_p}] \\ \mathbf{0} & \mathbf{0} \end{array} . \tag{2.51}$$

As described with the simple 2-dimensional example the above matrix is non-zero only for the entities which relate the boundary degrees of freedom of the structure and the pressures of the cells of the boundary $\Gamma = \partial\Omega_s \cap \partial\Omega_f$. For linear problems, where the integration is done on the initial undeformed surface, the matrix $\mathbf{C}_{S_p}$ is constant during the primal and adjoint state since it depends only on the initial geometry.

The off-diagonal term $\partial\mathbf{F}/\partial\mathbf{u}$ describes the change of the flow equations caused by a change of the structural displacements and its size is $4N_c \times N_{dofs}$,

$$\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \begin{array}{c|c} & N_{dofs} \\ \hline 3N_c & \dfrac{\partial \mathbf{M}}{\partial \mathbf{u}} \\ N_c & \dfrac{\partial \mathbf{C}}{\partial \mathbf{u}} \end{array} . \tag{2.52}$$

The partial derivatives of the momentum and continuity equations with respect to the structural displacements can be calculated with a forward finite difference scheme as follows

$$\frac{\partial \mathbf{M}(\mathbf{s},\mathbf{u},\mathbf{v},\mathbf{p})}{\partial \mathbf{u}} \approx \frac{\mathbf{M}(\mathbf{s},\mathbf{u}+\Delta\mathbf{u},\mathbf{v},\mathbf{p}) - \mathbf{M}(\mathbf{s},\mathbf{u},\mathbf{v},\mathbf{p})}{\Delta\mathbf{u}} \tag{2.53}$$

$$\frac{\partial \mathbf{C}(\mathbf{s},\mathbf{u},\mathbf{v},\mathbf{p})}{\partial \mathbf{u}} \approx \frac{\mathbf{C}(\mathbf{s},\mathbf{u}+\Delta\mathbf{u},\mathbf{v},\mathbf{p}) - \mathbf{C}(\mathbf{s},\mathbf{u},\mathbf{v},\mathbf{p})}{\Delta\mathbf{u}}. \tag{2.54}$$

The momentum and continuity equations, $\mathbf{M}$ and $\mathbf{C}$, are zero at $(\mathbf{s},\mathbf{u},\mathbf{v},\mathbf{p})$ since at this point the system is in equilibrium. For the evaluation of the equations at the point $(\mathbf{s},\mathbf{u}+\Delta\mathbf{u},\mathbf{v},\mathbf{p})$ no additional fluid solution is required and the residual change is evaluated for the disturbed geometry while the velocity $\mathbf{v}$ and pressure $\mathbf{p}$ remain constant.

Alternatively, the term $\partial\mathbf{F}/\partial\mathbf{u}$ is possible to be calculated analytically if the fluid code is available. In [1] the term is decomposed as follows

$$\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \frac{\partial \mathbf{F}}{\partial \mathbf{X}}\frac{\partial \mathbf{X}}{\partial \mathbf{u}}, \tag{2.55}$$

where $\mathbf{X}$ is the volume mesh. The first term of the right-hand side of (2.55) is evaluated in [128] with matrix-vector multiplications via automatic differentiation or an analytic differentiation of the Navier-Stokes equations. The second term can be understood only in its discrete form and expresses the influence that the surface displacement has on the position of the volume vertices. It is obtained by differentiating the mesh update tool of the fluid solver.

A change of the volume degrees of freedom of the structure does not result to a change of the flow equations and for these degrees of freedom the above derivatives are zero. Moreover, for a linearized problem an infinitesimal change on the structural boundary will have an effect only on the boundary cells. In practice, the finite disturbance is applied to the fluid domain using a mesh motion algorithm. Depending on the mesh motion algorithm which is used this matrix can be dense or sparse and respective efficient storage algorithms are required. Regarding only a local change and by separating again the boundary from the domain terms the derivative matrix of equation (2.52) results to

$$\frac{\partial \mathbf{F}}{\partial \mathbf{u}} = \left[\begin{array}{c:c} \mathbf{0} & \mathbf{0} \\ \hdashline \begin{matrix}\mathbf{C}_{f_M} \\ \mathbf{C}_{f_C}\end{matrix} & \mathbf{0} \end{array}\right], \tag{2.56}$$

where the matrices $\mathbf{C}_{f_M}$ and $\mathbf{C}_{f_C}$ are the non-zero entities of the derivatives of equations (2.53) and (2.54), respectively.

After substitution of equations (2.51) and (2.56) to (2.48) the following adjoint equation system is obtained

$$\left[\begin{array}{c:c}
\left(\frac{\partial \mathbf{F}}{\partial(\mathbf{v},\mathbf{p})}\right)^T & \begin{array}{c:c} \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{C}_S^T & \mathbf{0} \end{array} \\ \hdashline
\begin{array}{c:c} \mathbf{0} & \mathbf{C}_f^T \\ \hdashline \mathbf{0} & \mathbf{0} \end{array} & \left(\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right)^T
\end{array}\right]
\left[\begin{array}{c}
\boldsymbol{\psi}_{\mathbf{v}_\Omega} \\ \boldsymbol{\psi}_{\mathbf{p}_\Omega} \\ \hdashline \boldsymbol{\psi}_{\mathbf{v}_\Gamma} \\ \boldsymbol{\psi}_{\mathbf{p}_\Gamma} \\ \hdashline \boldsymbol{\psi}_{\mathbf{u}_\Gamma} \\ \hline \boldsymbol{\psi}_{\mathbf{u}_\Omega}
\end{array}\right]
=
\left[\begin{array}{c}
\frac{\partial J}{\partial(\mathbf{v},\mathbf{p})_\Omega} \\ \hdashline \frac{\partial J}{\partial(\mathbf{v},\mathbf{p})_\Gamma} \\ \hdashline \frac{\partial J}{\partial \mathbf{u}_{\Gamma,\Omega}}
\end{array}\right], \tag{2.57}$$

where the evaluation of the matrices $\mathbf{C}_f = [\ \mathbf{C}_{f_M} \quad \mathbf{C}_{f_C}\ ]$ and $\mathbf{C}_S$ is described earlier in this section.

Expanding the matrix, the following system of equations are obtained

$$\left(\frac{\partial \mathbf{F}}{\partial(\mathbf{v},\mathbf{p})}\right)^T \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})} = \left[\begin{array}{c} \frac{\partial J}{\partial(\mathbf{v},\mathbf{p})_\Omega} \\ \frac{\partial J}{\partial(\mathbf{v},\mathbf{p})_\Gamma} \end{array}\right] - \left[\begin{array}{c} \mathbf{0} \\ \mathbf{C}_S^T \end{array}\right] \boldsymbol{\psi}_{\mathbf{u}}. \tag{2.58}$$

$$\left(\frac{\partial \mathbf{S}}{\partial \mathbf{u}}\right)^T \boldsymbol{\psi}_{\mathbf{u}} = \frac{\partial J}{\partial \mathbf{u}_{\Gamma,\Omega}} - \left[\begin{array}{c} \mathbf{0} \\ \mathbf{C}_f^T \end{array}\right] \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})}. \tag{2.59}$$

Equation (2.58) is the discretized form of the adjoint Navier-Stokes equations (2.31) with the last term being an additional source term in the equation bringing the structural adjoint influence on the fluid problem. Moreover, equation (2.59) is the structural adjoint equation (2.14) with the last term being the additional coupling term showing the impact of the adjoint flow to the structural problem. As a result it is enough to evaluate the original single field adjoint problems by adding the additional source terms.

For the solution of the above system a partitioned approach is followed and the equations (2.58) and (2.59) are solved iteratively in a staggered manner. The procedure is described in algorithm (1) and is very similar to the one followed for the solution of the primal steady state fluid-structure interaction problem.

Algorithm 1 Solution of the adjoint fluid-structure interaction problem

```
 1: calculate $\mathbf{C}_f^T$ and $\mathbf{C}_S^T$
 2: initialize $\boldsymbol{\psi}^0_{(\mathbf{v},\mathbf{p})}$, $\boldsymbol{\psi}^0_{\mathbf{u}}$
 3: $k = 0$
 4: while (!converged) do
 5:     predict $\boldsymbol{\psi}^{k+1}_{\mathbf{u}}$
 6:     k=k+1
 7:     calculate the source term $\begin{bmatrix} \mathbf{0} \\ \mathbf{C}_S^T \end{bmatrix} \boldsymbol{\psi}^{k+1}_{\mathbf{u}}$
 8:     calculate $\boldsymbol{\psi}^{k+1}_{(\mathbf{v},\mathbf{p})}$ from equation (2.58)
 9:     calculate the source term $\begin{bmatrix} \mathbf{0} \\ \mathbf{C}_f^T \end{bmatrix} \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})}$
10:     calculate $\boldsymbol{\psi}^{k+1}_{\mathbf{u}}$ from equation (2.59)
11:     check convergence
12: end while
```

As described in algorithm (1) starting with a converged fluid-structure interaction result first the matrices $\mathbf{C}_f^T$ and $\mathbf{C}_S^T$ are evaluated. These matrices depend only on the primal state and can be calculated once before the adjoint fluid-structure interaction loop. After the initialization of the unknown variables $\boldsymbol{\psi}^0_{(\mathbf{v},\mathbf{p})}$, $\boldsymbol{\psi}^0_{\mathbf{u}}$ the adjoint fluid-structure interaction loop begins. In step (5) of algorithm (1) the new adjoint displacement is predicted. This prediction can be made using any of the available algorithms used in fluid-structure interaction problems presented in [20]. Then the source terms of equations (2.58) and (2.59) are evaluated and the problem is solved till convergence is reached.

After the evaluation of the adjoint variables $(\boldsymbol{\psi}_{\mathbf{u}}, \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})})$ the sensitivity of the objective function with respect to the design variable $\mathbf{s}$ is computed according to equation (2.8) as follows

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \boldsymbol{\psi}_{(\mathbf{v},\mathbf{p})}^{T} \frac{\partial \mathbf{F}}{\partial \mathbf{s}} + \boldsymbol{\psi}_{\mathbf{u}}^{T} \frac{\partial \mathbf{S}}{\partial \mathbf{s}}. \tag{2.60}$$

When the response function J is not defined on the design boundary as in the case of power loss defined in (2.21), the first partial derivative of equation (2.60) is zero. The evaluation of the second term was described in Section 2.2.2 and in this work the formulation presented by Othmer [96] is used.

Expanding the last term of the sensitivity in (2.60) gives

$$\boldsymbol{\psi}_{\mathbf{u}}^{T} \frac{\partial \mathbf{S}}{\partial \mathbf{s}} = \boldsymbol{\psi}_{\mathbf{u}}^{T} \left(\frac{\partial \mathbf{K}}{\partial \mathbf{s}} \mathbf{u} - \frac{\partial \mathbf{f}}{\partial \mathbf{s}} \right). \tag{2.61}$$

This equation is zero in case the design boundary does not coincide with the fluid-structure interface. When the two surfaces coincide it can be evaluated using semi-analytic sensitivity analysis described in Section 2.12.

2.3.3 The SIMPLE algorithm for the adjoint coupled problem

In this section the discrete formulation of equation (2.58) is presented in its continuous form in order to clarify how the additional coupling term can be added to the continuous adjoint Navier-Stokes equations. Then the solution procedure for the adjoint Navier-Stokes equation for the coupled problem is shown.

The discrete adjoint Navier-Stokes equations with the additional coupling term of equation (2.58) can be written in a continuous form in

Cartesian coordinates as

$$-\frac{\partial \psi_{vj}}{\partial x_i} v_j - v_j \frac{\partial \psi_{vi}}{\partial x_j} = \frac{\partial}{\partial x_j} 2\nu \varepsilon_{ij}(\psi_v) + \frac{\partial \psi_p}{\partial x_i} \tag{2.62a}$$

$$\frac{\partial \psi_{vi}}{\partial x_i} = -C_{S_p} d. \tag{2.62b}$$

For sake of simplicity, a response function defined only on the boundary of the computational domain is considered. On the above equation, the additional coupling term appears only in equation (2.62b) since there is no direct dependence of the structural residual on the fluid velocities, as already discussed in equation (2.49) of the previous section. The additional coupling term is constant during the solution of the fluid equations at each iteration of the adjoint fluid-structure interaction problem since it is the result of the multiplication of the adjoint structural displacements with the derivative of the structural residual with respect to the pressure.

Due to the lack of an explicit adjoint pressure transport equation a SIMPLE-based algorithm is followed for the numerical solution of the above equation system. The notation used is the one of Ferziger and Peric [38] and the solution strategy is similar to the one followed for the solution of the primal and adjoint single field problems presented in detail in Appendix B. Starting with a pair of adjoint velocities and pressures $({\psi_v}_i^{m-1}, \psi_p^{m-1})$ which satisfy equation (2.62), the semi-discretized momentum equation for the adjoint velocities is

$$\alpha_P {\psi_v}_{i,P}^{m*} + \sum_l \alpha_l {\psi_v}_{i,l}^{m*} = -\Delta\Omega \left(\frac{\partial {\psi_p}^{m-1}}{\partial x_i} \right)_P - \Delta\Omega \left(v_j \frac{\partial {\psi_v}_j^{m-1}}{\partial x_i} \right)_P . \tag{2.63}$$

The index P is the current index of the velocity while l refers to its neighboring nodes for each discretized equation. Here, the second term of equation (2.62a) is treated explicitly.

As a result the adjoint velocity at node P is

$${\psi_v}_{i,P}^{m*} = \tilde{\psi_v}_{i,P}^{m*} - \frac{\Delta\Omega}{\alpha_P} \left(\frac{\partial {\psi_p}^{m-1}}{\partial x_i} \right)_P , \tag{2.64}$$

where

$$\tilde{\psi_{v}}_{i,P}^{m*} = \frac{1}{\alpha_P}\left(-\sum_l \alpha_l {\psi_{v}}_{i,l}^{m*} - \Delta\Omega\left(v_j \frac{\partial {\psi_{v}}_j^{m-1}}{\partial x_i}\right)_P\right). \tag{2.65}$$

The $*$ symbol on the above equations denotes that the resulting adjoint velocities will not satisfy equation (2.62b). Thus, a “projection” or “correction” of this field is required such that

$$\frac{\partial {\psi_{v}}_i^m}{\partial x_i} = -C_{Sp}d. \tag{2.66}$$

The pair $({\psi_{v}}_i^m, \psi_p^m)$ which satisfies the above equation should also satisfy equation (2.64). As a result it holds that

$${\psi_{v}}_{i,P}^m = \tilde{\psi_{v}}_{i,P}^{m*} - \frac{\Delta\Omega}{\alpha_P}\left(\frac{\partial {\psi_p}^m}{\partial x_i}\right)_P. \tag{2.67}$$

By substituting equation (2.67) to (2.66), a discrete equation for the adjoint pressure is obtained

$$\frac{\partial}{\partial x_i}\left(\frac{\Delta\Omega}{A_p}\left(\frac{\partial {\psi_p}^m}{\partial x_i}\right)_P\right) = \left(\frac{\partial \tilde{\psi_{v}}_i^{m*}}{\partial x_i}\right)_P - \Delta\Omega\left(C_{Sp}d\right)_P. \tag{2.68}$$

After solution of the above equation the adjoint velocity field which satisfies (2.66) is evaluated from equation (2.67). The new pair of adjoint variables $({\psi_{v}}_i^m, \psi_p^m)$ does not satisfy the momentum equation. Consequently, an iterative procedure is initiated until the pair satisfies both equations.

Chapter 3

Node-based shape optimization

In this chapter the methods required for regularization of the ill-posed node-based shape optimization problem are presented. The chapter is organized as follows: Section 3.1 introduces the optimization problem in its general form. The various out-of-plane regularization methods are briefly presented in Section 3.2 and the proposed Vertex-Morphing method is discussed in Section 3.3. At the end of the chapter (Section 3.4), the implicit In-plane regularization method which is utilized for retaining the quality of the surface mesh during optimization is presented.

3.1 The optimization problem

The following discrete optimization problem is considered

$$\begin{array}{llcll} & J(\mathbf{s},\mathbf{w}(\mathbf{s})) & \rightarrow & \min & \\ \text{s.t.} & \mathbf{R}(\mathbf{s},\mathbf{w}(\mathbf{s}) & = & \mathbf{0} & , \\ \text{and} & g_j & = & g_j(\mathbf{s},\mathbf{w}) \leq 0 & , \forall j \in 1,..,n_j \\ & h_k & = & h_k(\mathbf{s},\mathbf{w}) = 0 & , \forall k \in 1,..,n_k \\ & s_l^{\text{lower}} \leq & s_l & \leq s_l^{\text{upper}} & , \forall s_l \in \mathbb{R}, l = 1,..,n, \end{array} \tag{3.1}$$

where J is the objective function and $\mathbf{s}$, $\mathbf{w}$ the discrete design and state variables respectively described in the previous chapter. The problem is constrained by the state equation $\mathbf{R}$, which can be for instance the equation describing a linear-elastic structure or an incompressible flow. It can also have additional n_j inequality and n_k equality constraints defined as g and h, respectively. The design variables can be also bounded by lower s^{lower} and upper s^{upper} bounds.

The methods and derivations presented in this chapter are independent of the nature of the state equation as well as the type of the objective function and constraints. Any gradient-based optimization algorithm can be applied and the gradients can be evaluated as in (2.2.1) for a linear elastic structural state, or as in (2.2.2) for an incompressble Newtonian steady-state flow problem, or as in (2.3) for a steady-state fluid-structure interaction problem using the adjoint method. Generally, here the control of the design as well as the surface and volume mesh is discussed independent of the optimization method applied. The goal of the chapter is to show how the node-based shape during optimization problem can be regularized from a geometrical perspective and for this reason the optimization applications will be restricted for simplicity to unconstrained problems.

3.2 Out-of-plane regularization

The sensitivity field evaluated on the design surface is generally non-smooth. The non-smooth sensitivity field results in a noisy geometry update with small scale oscillations. This irregularity arises from the fact that the computed gradients are less regular than the parameterization and for this reason a projection or a smoothing step is required [64, 89, 106–108]. Generally, the methods used for smoothing the sensitivity field or the design update can be explicit or implicit. Regarding each method in an operator form, the explicit methods operate directly on the noisy field while the implicit ones operate on the smooth (unknown) field. In the sequence, an overview of the most commonly applied methods is presented constraining the derivations in the one-dimensional case.

The Sobolev gradient smoothing is a well established implicit filter commonly used in aerodynamic adjoint shape optimization [106, 108]. According to [106], the method delivers a smooth gradient field by approximating the Hessian of the problem with the Laplace-Beltrami operator

$$I - \epsilon^2 \frac{\partial}{\partial x}\frac{\partial}{\partial x}. \tag{3.2}$$

Using this operator is equivalent to employing a modified Sobolev inner product to describe the shape derivative.

More precisely, a Newton's step for the design variable s results in the following update

$$s_{n+1} = s_n - H_s^{-1} G, \tag{3.3}$$

where G and H_s are the gradient and the Hessian of the objective function J, respectively. By approximating the Hessian with the Laplace-Beltrami operator (3.2) the smooth descent direction $\bar{G}$ can be evaluated as the solution of the following PDE:

$$\bar{G} - \epsilon^2 \frac{\partial}{\partial x}\frac{\partial \bar{G}}{\partial x} = G, \tag{3.4}$$

with zero Neumann boundary conditions. In one-dimensional problems, the variable ϵ can be interpreted as an arbitrary positive value which penalizes the high curvatures detected by the curvature operator

$$\frac{\partial^2}{\partial x^2}. \tag{3.5}$$

The higher the ϵ, the higher the penalization. It should be noted that the same idea can be applied directly on the design update (δs). Equation (3.4) can be solved separately for the unknown $\bar{G}$ or can be applied as a preconditioner to the sensitivity equations.

By definition the obtained gradient is two classes smoother than the initial one. Considering the zero Neumann boundary condition of (3.4), for the variation of the objective it follows

$$\begin{aligned}\delta J &= \int G\delta s dx \\ &= \int (\bar{G} - \epsilon^2 \frac{\partial}{\partial x}\frac{\partial \bar{G}}{\partial x})\delta s dx \\ &= \int (\bar{G}\delta s + \epsilon^2 \frac{\partial \delta s}{\partial x}\frac{\partial \bar{G}}{\partial x})dx.\end{aligned} \tag{3.6}$$

The last relation can be seen as a modified Sobolev inner product (weighted by ϵ^2) and therefore this preconditioning is often referred to as a Sobolev smoothing. Moreover, the smooth gradient $\bar{G}$ is a Riesz-representative of the L^2 gradient G in the scalar product defined in (3.6) as discussed also in [106].

Then, the variation of the design δs can be obtained using the smooth sensitivity field as

$$\delta s = -\lambda \bar{G}, \tag{3.7}$$

with λ the step size. Therefore, it follows that the variation of the objective function J is the following

$$\delta J = \int G\delta s dx = -\lambda \int (\bar{G}^2 + \epsilon \left(\frac{\partial \bar{G}}{\partial x}\right)^2)dx, \tag{3.8}$$

which guarantees decrease on the objective J.

Explicit filters act directly on the raw sensitivity field or the geometry update by convoluting it with a kernel K as follows

$$\bar{G}(\boldsymbol{\xi_0}) = \int_\Gamma K(\boldsymbol{\xi})G(\boldsymbol{\xi_0} - \boldsymbol{\xi})d\boldsymbol{\xi}. \tag{3.9}$$

The local coordinate $\boldsymbol{\xi}$ is defined on the design surface Γ and K is selected to be the Gaussian kernel

$$K(\boldsymbol{\xi};\sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^D} e^{-\frac{|\boldsymbol{\xi}|^2}{2\sigma^2}}, \tag{3.10}$$

where D is the dimension of Γ and σ^2 the variance. The higher the variance, the wider the effect of the filter and thus the variance σ is often referred to as filter radius. For this type of filtering there is no linear system solution needed as the filter acts as an external operator.

In the sequence the relation between the presented implicit and explicit filters is shown in the one-dimensional case. The convolution of the raw sensitivity field with the Gaussian filter with $\sigma^2 = 2\beta t$ is the solution of the following linear, isotropic diffusion equation

$$\frac{\partial \bar{G}}{\partial t} = \beta \frac{\partial^2 \bar{G}}{\partial x^2} \tag{3.11}$$

with initial condition

$$\bar{G}(x,0) = G(x). \tag{3.12}$$

The variable β is the “diffusion” coefficient showing how fast is the diffusion of G over the pseudo-time t. According to [108, 116] a first order time discretization of equation (3.11) yields

$$\bar{G} - \beta t \frac{\partial^2 \bar{G}}{\partial x^2} = \bar{G}(0) \tag{3.13}$$

Comparing equations (3.4) and (3.13) results to the following relation

$$\sigma^2 = 2\epsilon. \tag{3.14}$$

Hence, a convolution filtering with a Gaussian kernel is first order equivalent to the implicit Sobolev-gradient method.

An optimal choice of ε and σ would correspond to a Newton’s step [107] but then an additional effort would be required for the determination of this optimal value. Alternatively, this projection can be understood as a design handle. The choice of σ determines the wave lengths appearing on the improved shape and can be decided upon the manufacturing constraints.

3.3 The Vertex-Morphing method

The smoothing methods presented in the previous section regularize the optimization problem in an extra step, decoupled from the optimization problem, by applying an operator on the sensitivity field. In this section, a consistent formulation, the Vertex-Morphing method [9, 56], is presented in which the smoothing is not done anymore as a "post-processing" step, but it is incorporated into the optimization problem.

In addition to the geometry field, a control field is introduced, on which the optimization problem is defined. As described in the next paragraphs, the control field is related to the geometry through a mapping operator.

More precisely, consider the one-dimensional optimization problem $J(x,u) \rightarrow min$ defined in $x(\xi) \in \Gamma : (-1,1) \rightarrow \mathbf{R}$ for which the state equation $R(x,u) = 0$ should be satisfied. The geometry is obtained from the design or control variable $s(\xi), s \in \Gamma$ by a linear transformation $\mathcal{A} : \Gamma \rightarrow \Gamma$,

$$x(\xi_0) = \mathcal{A}s = \int_{-1}^{1} A(\xi_0, \xi) s(\xi) d\xi. \tag{3.15}$$

If A is translation invariant then

$$x(\xi_0) = \int_{-1}^{1} A(\xi_0 - \xi) s(\xi) d\xi \tag{3.16}$$

and the geometry x is obtained by a convolution of the continuous control field s with a kernel A which can be considered as a generalization of the Gaussian kernel K presented in (3.10).

According to the above equation the variation of the geometry is

$$\delta x(\xi_0) = \int_{-1}^{1} A(\xi_0, \xi) \delta s(\xi) d\xi = \int_{-1}^{1} A(\xi_0 - \xi) \delta s(\xi) d\xi, \tag{3.17}$$

and the sensitivity of the objective function with respect to the control variable s is given by

$$\frac{dJ}{ds} = \frac{\partial J}{\partial s} + \frac{\partial J}{\partial x} \frac{dx}{ds}. \tag{3.18}$$

The first term of equation (3.18) is zero since the objective function does not depend directly on the control field. Finally, substituting equation (3.17) to (3.18) yields

$$\frac{dJ}{ds}(\xi_0) = \int_{-1}^{1} \frac{\partial J}{\partial x} A(\xi, \xi_0) d\xi. \tag{3.19}$$

The partial derivative $\partial J / \partial x$ implies that during this evaluation s is not varying. However, the state variables of the problem can still vary.

Consequently, this term can be evaluated at each point of the design surface using the adjoint method presented in Chapter 2.

The essential difference to the methods presented in the previous section is that in Vertex-Morphing the filtering operation is included in the optimization problem consistently through the chain rule. Thus the filtering appears twice during an optimization step. Once during the design update and once for the evaluation of the sensitivities. Moreover, using a generalization of the rule of equation (3.15) one can regularize simultaneously both the normal and tangential update, as described in [56].

In numerical shape optimization one deals with discretized geometries and therefore the control field is discretized as well. Let us first consider that the geometry and the control field have identical spatial discretizations, $\boldsymbol{x} = [x_1, \dots, x_n]$ and $\boldsymbol{s} = [s_1, \dots, s_n]$, respectively. In such a case and equivalent to equation (3.15), the mapping between $\boldsymbol{s}$ and $\boldsymbol{x}$ is a square matrix and the linking is as follows

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}. \tag{3.20}$$

If an equidistant spatial discretization is chosen and the linear map of equation (3.16), A is symmetric then the operator $\boldsymbol{A}$ is also symmetric and thus $\boldsymbol{A}^T = \boldsymbol{A}$. However, it is not necessary to have the same discretizations for $\boldsymbol{x}$ and $\boldsymbol{s}$. Based on the above definition, CAD geometry representation is nothing but a coarse discretization of the control field through a set of "control points" which describe the fine geometry mesh. There, the $\boldsymbol{A}$ operator is a rectangular matrix containing the weightings of the control points for each mesh point.

From equation (3.20) it follows that the variation $\delta\boldsymbol{x}$ and the sensitivity $\nabla_{\boldsymbol{s}} J$ with respect to $\boldsymbol{s}$ can be written as

$$\begin{aligned} \delta\boldsymbol{x} &= \boldsymbol{A}\delta\boldsymbol{s}, \\ \nabla_{\boldsymbol{s}} J &= \boldsymbol{A}^T \nabla_{\boldsymbol{x}} J. \end{aligned} \tag{3.21}$$

Again, here it becomes clear that the linear operator $\boldsymbol{A}$ appears twice during one optimization step: once during the design update and once in the evaluation of the sensitivities.

In order to study the effect of the linking rule of equation (3.20) on the optimization problem, the second order Taylor series approximation of the objective function is considered

$$\tilde{J} \approx J + (\nabla_{\boldsymbol{s}} J)^T \delta\boldsymbol{s} + \frac{1}{2}\delta\boldsymbol{s}^T \boldsymbol{H}_{\boldsymbol{s}} \delta\boldsymbol{s}, \tag{3.22}$$

where $\boldsymbol{H}_{\boldsymbol{s}}$ is the Hessian matrix with respect to $\boldsymbol{s}$. Furthermore, a Taylor series expansion of the stationary condition $\nabla_{\boldsymbol{s}}\tilde{J} = 0$ yields to the Newton's method for the update in $\boldsymbol{s}$

$$\delta\boldsymbol{s} = -\boldsymbol{H}_{\boldsymbol{s}}^{-1} \nabla_{\boldsymbol{s}} J. \tag{3.23}$$

Considering that

$$\begin{aligned} \nabla_{\boldsymbol{s}} J &= \boldsymbol{A}^T \nabla_{\boldsymbol{x}} J, \\ \boldsymbol{H}_{\boldsymbol{s}} &= \boldsymbol{A}^T \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{A} \end{aligned} \tag{3.24}$$

equation (3.23) becomes

$$\delta\boldsymbol{s} = -(\boldsymbol{A}^T \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{A})^{-1} \boldsymbol{A}^T \nabla_{\boldsymbol{x}} J. \tag{3.25}$$

Therefore, the geometry update results to

$$\delta\boldsymbol{x} = \boldsymbol{A}(-\boldsymbol{A}^{-1} \boldsymbol{H}_{\boldsymbol{x}}^{-1} \boldsymbol{A}^{-T}) \boldsymbol{A}^T \nabla_{\boldsymbol{x}} J = -\boldsymbol{H}_{\boldsymbol{x}}^{-1} \nabla_{\boldsymbol{x}} J, \tag{3.26}$$

which implies that the geometry update is not influenced by the control field $\boldsymbol{s}$ and the solution of the optimization problem remains unchanged. In other words, assuming that the problem is convex, there is no effect on the optimum solution from the choice of $\boldsymbol{A}$ and the operator $\boldsymbol{A}$ defines only a reparamerization of the geometry. Moreover, because of equation (3.26), this reparametrization takes effect only in combination with first order or quasi-Newton optimization algorithms as it cancels when the full second order information is used. Then, different choices of $\boldsymbol{A}$ would result in different iteration histories, which in case of a non-convex problem, may lead to different local minima.

In most engineering applications, the optimization problem is non-convex. The goal in this case is to find a local minimum, which satisfies the manufacturing and design requirements. Typically, individual local minima are defined by the characteristic wavelengths of the shape. It appears that different operators $\boldsymbol{A}$ are related to those minima through the filter radius. So, the filter radius can be used as a design handle to steer the optimizer to local minima with the appropriate design prerequisites.

In three-dimensional problems, where a three-dimensional design surface is to be optimized, the surface mesh quality is the main limiting factor and must be treated through an in-plane regularization step, such as the one presented in the next section. However, in three-dimensional Vertex-Morphing, the in- and out-plane regularization is done simultaneously and through a suitable choice of the mapping operator A, as described in [56]. As such, large design updates can be obtained with no additional effort. The performance of the method in mesh quality improvement in three-dimensional industrial examples is demonstrated in the next chapter.

Moreover, a crucial point in industrial shape optimization is the preservation of the design characteristics, such as sharp corners and edges. Filter-based methods such as Vertex-Morphing, perform the shape variation in wavelengths much larger than these features. As the result, design features are preserved after large shape changes in the design surface until late stages of the optimization. Clearly, after many optimization iterations the shape converges to its optimal which may eliminate design features.

Besides the theoretical aspects, the method is easy to implement. Due to its explicit nature, it can be applied even to black-box primal and dual solvers as an external module and no additional linear system solution is needed. Additionally, the construction of the linear operator $\boldsymbol{A}$ requires only the cloud of the points of the design surface without their connectivity. For the neighbor-search already developed efficient algorithms can be explored [79–81].

From the users point of view, there is no additional parameterization of the geometry required which is time consuming and requires experience from the user. The method applies directly on the existing discretization of the design surface which was already used for the primal calculation. In this way, a design space as large as possible is obtained while the only design handle is the filter radius of the linear operator. The filter radius is defined globally representing the smaller wavelengths desired during optimization.

3.4 In-plane regularization

3.4.1 Motivation

The Vertex-Morphing method presented in the previous section regularizes both the normal and tangential components of the design update simultaneously and its performance in industrial examples is shown in Chapter 4. Here, in order to evaluate the method only with respect to the quality of the resulting surface meshes during optimization the two-dimensional ducted flow of figure 3.1 is considered. The objective is the power dissipation defined in Section 2.21 and the curved part of the upper wall of the duct is the design surface. As one can expect, the optimizer will update the surface such that the clogged section gets unblocked such that the flow passes freely through the duct. As a result the curved part will move upwards and the high curvatures will be smoothened out. Reduction of curvature is a very common but challenging task in shape optimization since this reduction will cause overlapping of elements. For sake of comparison three different cases were studied. In the first case, a smoothing on the sensitivity field is applied as described in Section 3.2. In the second case, the Vertex-Morphing method is applied and in the third one an additional In-plane regularization [114] is used to reinforce the quality of the mesh.

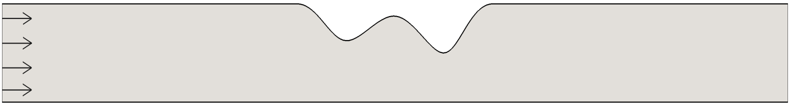


Figure 3.1: 2D ducted flow.

Figure 3.2 compares the mesh density conservation in the first 25 optimization iterations at which the clogged part of the duct is almost flat. The values are normalized, therefore a mesh density of 1 indicates the perfect mesh and 0 the collapse of the mesh. As one can observe the first case were no additional treatment of the tangential direction takes place, the optimization fails at the 11th iteration while the other two cases continue till the 25th iteration. The Vertex-Morphing method has almost the same computational cost as the smoothing of only the sensitivity field but it delivers higher quality of surface meshes and thus allows for more optimization steps. Adding In-plane regularization

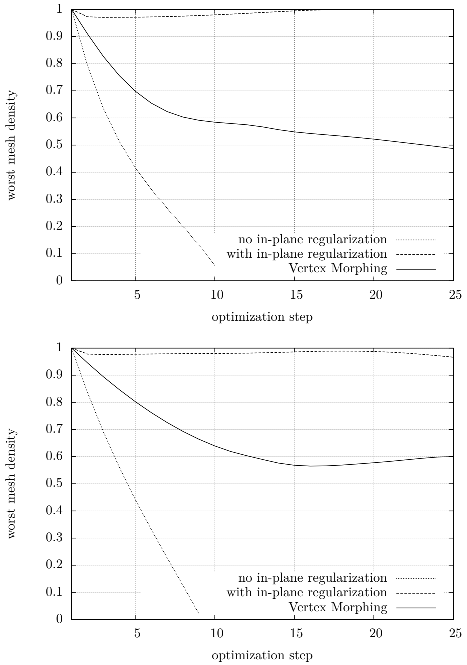


Figure 3.2: Comparison of the surface mesh density conservation for two filter radii (the filter of the bottom graph is 1.6 times larger of the top one).

will keep the surface mesh quality almost at its best. However, an additional effort has to be given as described in the rest of this chapter.

In what follows, a global method which regularizes the surface mesh to a desired condition is presented. In this method, an artificial stress field is applied on the surface or on the volume mesh and a global linear system for the equilibrium is solved. The applied stress adapts the shape of each element towards a desired predefined template geometry and at the end a globally smooth mesh is achieved. In this way, both the shape and the size of each element are controlled. The method can be applied on both structured and unstructured grids since there is no additional assumption on the mesh topology.

The remainder of this section is organized as follows: Section 3.4.2 presents the additional in-plane regularization term with which the optimization problem of Section 3.1 is augmented. In Section 3.4.3, a short overview is given on the methods used so far for mesh smoothing and mesh quality control with emphasis to elliptic smoothers since the proposed method is also elliptic. Finally, the In-plane regularization method is discussed in Section 3.4.4 and the role of the fundamental components of the method is demonstrated.

3.4.2 The augmented optimization problem

In general, in node-based shape optimization the shape is described only by the discretization and no other geometrical link is established. Therefore, the coordinates of the surface nodes are considered to be the design variables of the optimization problem, since changing the position of the internal nodes will not alter the shape. More precisely, an update of the position of a surface point can be decomposed into two components, normal and tangential to the surface, as follows:

$$\boldsymbol{x_l} = \tilde{\boldsymbol{x_l}} + x_{l,n} \cdot \boldsymbol{n} + x_{l,t} \cdot \boldsymbol{t} = s_l \cdot \boldsymbol{n} + r_l \cdot \boldsymbol{t}, \quad l = 1, \ldots, n_s, \qquad (3.27)$$

where $\tilde{\boldsymbol{x_l}}$ is the initial position, n_s is the number of surface nodes and $\boldsymbol{n}$ and $\boldsymbol{t}$ are the unit update vectors normal and tangential to the surface at node l.

Neglecting the discretization error and the finite step size of the optimization, small variations of $x_{l,n}$ in the "out of plane" direction will cause a change in the shape, while small variation of $x_{l,t}$ in the "in-plane" direction will only alter the discretization. In other words, $x_{l,n}$ is the shape relevant component s_l, while $x_{l,t}$ is the mesh relevant component r_l. Thus, only the normal component s_l is regarded as design variable and consequently in each optimization step only the out of plane direction is updated. However, during this process the quality of elements could deteriorate and severely distorted elements might appear. In extreme cases, the elements become degenerate and further progress of analysis is restricted. Hence, a correction in the "in-plane" direction is required.

For this reason, the optimization problem of Section 3.1 with response function J, constraints (g_j, h_k) and variable bounds $(s_l^{\text{lower}}, s_l^{\text{upper}})$ for each surface node l is modified as follows:

$$\begin{array}{llll} & \tilde{J}(\mathbf{s},\mathbf{w}(\mathbf{s})) & = & J(\mathbf{s},\mathbf{w}(\mathbf{s})) + \mathcal{R}(\mathbf{s},\mathbf{r}) & \rightarrow \min \\ s.t. & \mathbf{R}(\mathbf{s},\mathbf{w}(\mathbf{s}) & = & \mathbf{0} & , \\ and & g_j & = & g_j(\mathbf{s},\mathbf{w}) \leq 0 & ,\forall j \in 1,..,n_j \\ & h_k & = & h_k(\mathbf{s},\mathbf{w}) = 0 & ,\forall k \in 1,..,n_k \\ & s_l^{\text{lower}} \leq & s_l & \leq s_l^{\text{upper}} & ,\forall s_l \in \mathbb{R}, l = 1,..,n, \end{array} \tag{3.28}$$

where $\tilde{J}$ is the augmented objective function of the original objective J modified by a regularization term $\mathcal{R}$ responsible for retaining the mesh quality. The surface sensitivity of the augmented objective function $\tilde{J}$ is

$$\frac{d\tilde{J}}{d\boldsymbol{s}} = \frac{dJ}{d\boldsymbol{s}} + \frac{d\mathcal{R}}{d\boldsymbol{s}} \tag{3.29}$$

The augmented objective function should not alter the problem and since the only shape modifications occur in the normal to the surface direction, the term $\mathcal{R}$ should be modeled such that the shape derivative of the modified optimization problem is the same as the initial one, i.e.

$$\frac{d\tilde{J}}{d\boldsymbol{s}} = \frac{dJ}{d\boldsymbol{s}} \tag{3.30}$$

Here, in order to prevent any influence from the mesh regularization term on the original optimization problem, we define $\mathcal{R}$ such that its derivative with respect to the normal direction is zero. Also, in absence of discretization errors it holds in the tangent space

$$\frac{dJ}{d\boldsymbol{r}} = 0 \quad \text{and} \quad \frac{d\tilde{J}}{d\boldsymbol{r}} = \frac{dR}{d\boldsymbol{r}}. \tag{3.31}$$

Of course after discretization, the effects of $\boldsymbol{s}$ and $\boldsymbol{r}$ are not as clearly separated anymore and consequences due to discretization have to be accepted. Still the regularization term $\mathcal{R}$ controls the mesh quality. This term and its modeling is the main focus of the following sections.

3.4.3 Overview of mesh quality control methods

Generally, there are two main classes of problems in the field of mesh quality control: the mesh quality improvement and the mesh motion problems. In mesh smoothing, the goal is to improve the quality of a 2D or 3D mesh [52, 70, 119, 129], whereas in mesh motion problems the resulting 2D or 3D mesh is seeked after moving the 1D or a 2D boundary, respectively [18, 19, 65]. In both fields the ideas are similar. As the mesh motion solvers advance, bigger boundary updates can take place. These larger modifications of the design surface give rise to the mesh distortion problem of the boundary. For this reason, in-plane regularization methods are required which deal with retaining the surface mesh quality during the evolution of the shape [74, 105].

Mesh smoothing and grid generation

Various methods have been developed for mesh smoothing (mesh relaxation) or grid generation for complex surfaces and volume domains. Usually, in order to reduce this complexity, the physical domain where the computations need to be performed is transformed to a simpler region commonly called the logical space. The logical space can be a unit square in case of a two-dimensional problem or a unit cube for a three-dimensional problem. After defining a trivial grid in this region, the goal of these methods is to define a transformation from the logical to the physical space, as shown in figure 3.3. This transformation should have a non zero Jacobian. A zero Jacobian transformation can produce folded grids in reentrant corners and low quality elements.

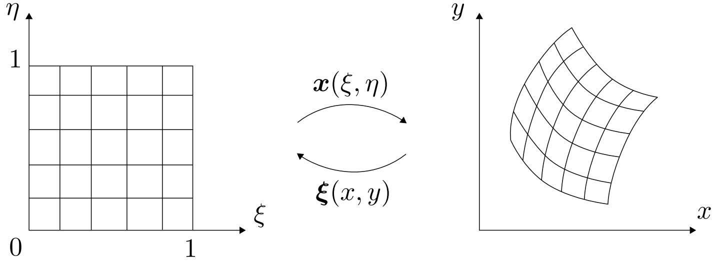


Figure 3.3: Transformation map $\boldsymbol{x}(\xi,\eta)$ from the logical (left) to the physical (right) space.

More precisely, according to [70] let U_k be the unit logical space of dimension k and Ω_k^n the physical region defined in the n-dimensional space. Then, the map from the logical to the physical space is defined as

$$X_k^n : U_k \to \Omega_k^n. \tag{3.32}$$

Hense, the problem of grid generation can be reduced to the following

Given the parametric boundary of the physical domain,

$$\partial x_k^n : \partial U_k \to \partial \Omega_k^n, \tag{3.33}$$

find the transformation map

$$X_k^n : U_k \to \Omega_k^n. \tag{3.34}$$

The proposed methods for this type of problems can be geometrically, mechanically or mathematically motivated. For instance the ideas behind the Delaunay triangulation and the octree methods arise from geometrical principles, while there are methods based on mechanical principles like the spring analogy. Moreover, in algebraic smoothing an interpolation between the given boundary curves is taking place. This type of grid generators are explicit in the sense that no linear system solution is needed and they can apply directly on an initial discretization. Furthermore, there are methods that solve a partial differential equation for the unknown transformation using with this way the general properties of the PDE equations . One can distinguish between hyperbolic, parabolic and elliptic grid generators that employ the advantages of each respective type of equation for generating a mesh.

Among them, the elliptic grid generators are the most commonly used. In this case elliptic equations are applied for smoothing the mesh or for propagating the motion of a boundary to the domain. A well known elliptic equation is the Laplace equation

$$\nabla^2 u = 0 \tag{3.35}$$

or the Poisson's equation

$$\nabla^2 u = f(x), \tag{3.36}$$

for a given function f. Elliptic equations describe usually steady-state problems and produce smooth solutions even for non-smooth boundaries.

In this class of methods, the Amsden-Hirt and the Thompson-Thames-Mastin grid generators are broadly used [69, 119] for both structured and unstructured grids.

In the elliptic Amsden-Hirt method, the transformation $\boldsymbol{x}$ between a square logical domain and the physical surface is sought and is considered to satisfy the Laplace equation

$$\nabla^2 x = x_{\xi\xi} + x_{\eta\eta} = 0, \quad \nabla^2 y = y_{\xi\xi} + y_{\eta\eta} = 0, \tag{3.37}$$

in 2D, with the parametric boundaries of the domain defining the boundary conditions of the above equation.

These equations are linear and satisfy the following variational form

$$I_{AH} = \frac{1}{2} \iint_0^1 (\boldsymbol{g}_1 \cdot \boldsymbol{g}_1 + \boldsymbol{g}_2 \cdot \boldsymbol{g}_2) dA \rightarrow \min_r, \tag{3.38}$$

where $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are the covariant base vectors of the physical domain

$$\boldsymbol{g}_1 = \frac{\partial \boldsymbol{x}}{\partial \xi} \quad \text{and} \quad \boldsymbol{g}_2 = \frac{\partial \boldsymbol{x}}{\partial \eta}, \tag{3.39}$$

and the integration is performed over the square logical domain. Minimization of this functional will lead to the Laplace equation (3.37) of the transformation map from the logical to the physical domain.

By solving this variational problem, the resulting transformation may have zero Jacobian and non-folded meshes in non-convex domains are therefore not guaranteed [70]. Solved in a local manner the equation reduces to the well know Laplacian smoother.

The Thompson-Thames-Mastin method guaranties non-zero Jacobians for the continuous problem by solving the same Laplacian problem for the inverse of the aforementioned transformation

$$\nabla^2 \xi = \xi_{xx} + \xi_{yy}, \quad \nabla^2 \eta = \eta_{xx} + \eta_{yy}. \tag{3.40}$$

Thus, the functional describing the method is the following:

$$I_{TTM} = \int_\Omega (\boldsymbol{g}^1 \cdot \boldsymbol{g}^1 + \boldsymbol{g}^2 \cdot \boldsymbol{g}^2) d\alpha \rightarrow \min_r, \tag{3.41}$$

where $\boldsymbol{g}^1$ and $\boldsymbol{g}^2$ are the contravariant base vectors of the physical domain

$$\boldsymbol{g}^1 = \frac{\partial \boldsymbol{\xi}}{\partial x} \quad \text{and} \quad \boldsymbol{g}^2 = \frac{\partial \boldsymbol{\xi}}{\partial y}, \tag{3.42}$$

and the integration is performed over the physical space. Unlike the Amsden-Hirt method, this equation leads to a nonlinear system of equations.

Castillo et al. [15] introduced a third target space which links the logical space and the physical space as in figure 3.4, where he introduced some desired properties of the final mesh. Hansen et al. in [51] extended this method for mesh smoothing of unstructured meshes by discretizing the resulting equation with finite elements and using a metric which incorporates influences from the neighboring elements.

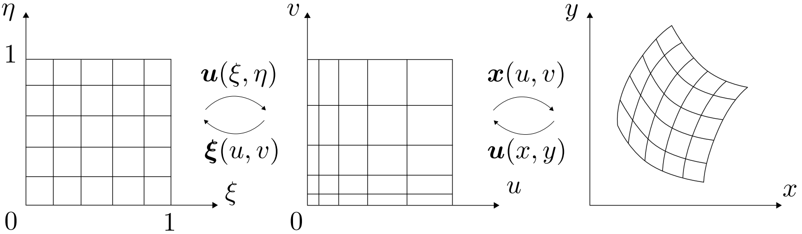


Figure 3.4: Transformation for logical (left) to target (middle) and then to physical space (right).

The regularization method presented in the Section 3.4.4 is also an elliptic method, which uses an idea similar to the target space in a local manner and individually for each element. Moreover, even though it is mechanically motivated it will be shown that in special cases, it reduces to the Amsden-Hirt method.

Mesh motion methods

In mesh motion problems the goal is to find a resulting volume or surface mesh after the surface of curve boundary is displaced. This mesh update is required by many applications like ALE flows with moving boundaries and shape optimization. There are several ways to perform a mesh update in such a case and the ideas are often similar to the ones for grid generation and mesh smoothing presented in the previous paragraph.

Farhat et al. [19, 31] developed a physically motivated approach which

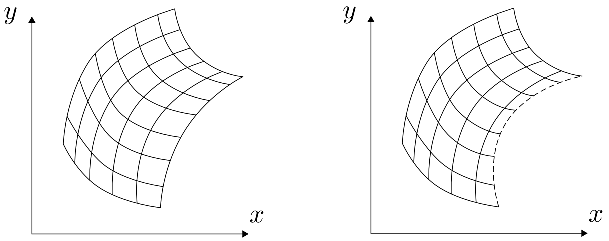


Figure 3.5: Starting from an initial mesh the goal is to find the resulting mesh after a boundary is displaced

introduces a net of linear and torsional springs in order to propagate the motion from the boundary to the fluid domain in moving boundary problems. Furthermore, in free-surface flows [7] and fluid-structure interaction applications [123], the fluid domain is regarded as a linear elastic body which deforms in the Lagrangian framework absorbing the boundary displacement. An algebraic method that does not require the connectivity of the nodes of the fluid domain and is based on the interpolation of the displacement of the boundary to the domain using radial basis functions was developed in [18]. In addition to the above methods, the diffusive property of the elliptic equations was used in various Laplacian methods and applied in unstructured grids and time dependent flows by Jasak et al. [65]. Here, according to [65], the Laplacian equation is solved for a defined variable diffusion γ

$$\nabla \cdot (\gamma \nabla \boldsymbol{u}) = 0, \tag{3.43}$$

where $\boldsymbol{u}$ is the velocity of each point. Thus, the updated position $\boldsymbol{x}_n$ of a point after mesh motion can be evaluated from the previous position $\boldsymbol{x}_{n-1}$ as

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \boldsymbol{u}\Delta t, \tag{3.44}$$

where Δt is the time step. The boundary conditions are inherited from the motion of the boundary and the equation is solved with finite elements using an iterative solver, [54].

This method is also used in this work for the update of the fluid domain during shape optimization as seen in the applications of the next chapter.

3.4.4 The In-plane regularization method

The proposed mesh regularization method is inspired by form-finding which is a method to determine the free-form equilibrium shape of membrane and shell structures subjected to a certain stress field [10, 75, 130]. In other words, assuming a stress field applied on the resulting structure, the displacement field which brings the system to equilibrium is seeked. Consequently, since no body forces and surface tractions are regarded, the weak form of equilibrium in the context of geometrical nonlinear analysis reduces to the internal virtual work done by the predefined stress field,

$$\delta w(\boldsymbol{s}, \boldsymbol{r}) = \delta w_{int} = t \int_a \boldsymbol{\sigma_0} : \delta \boldsymbol{\epsilon} da = 0, \tag{3.45}$$

with $\boldsymbol{\sigma_0}$ being the prescribed Cauchy stress tensor components acting on the resulting geometry with area a and $\boldsymbol{\epsilon}$ the Euler-Almansi strain tensor, [12]. The membrane stresses are considered constant through the thickness t. Therefore, the integration is performed on the mid-surface of the structure. When a constant isotropic stress field is applied, the resulting surface is a minimal surface, which is the surface of minimal area content connecting given boundaries.

Since the desired internal stress field is given, there is no material description needed and the problem reduces to a geometrical one even though the formulation is initiated by a mechanical equilibrium. Consequently, the input parameters required for the solution of the form finding problem without external loading is a set of Dirichlet boundary conditions on a boundary Γ and a stress field $\boldsymbol{\sigma_0}$ defined on the deformed geometry.

Again, optimal positions of the shape variables $\boldsymbol{s}$ have to be found. Typically, the tangential mesh related variables $\boldsymbol{r}$ cannot be eliminated to allow for large changes of shape. They remain as unknowns in the formulation. As a consequence, the resulting problem is singular with respect to the mesh related variables $\boldsymbol{r}$ because of the non-uniqueness of the discretization, [10]. In particular, the reason is that the base vectors of $\boldsymbol{\sigma_0}$ are related to the unknown geometry and are functions of $\boldsymbol{r}$ as well. Also, the governing equations are non-linear in $\boldsymbol{s}$ and $\boldsymbol{r}$.

For the regularization of the problem, the Updated Reference Strategy is suggested, [10]. In this method the problem is solved on the "updated reference geometry" of the form finding step (i):

$$\delta w(\boldsymbol{s}, \boldsymbol{r}) = t \int_{A^i} \boldsymbol{S}^i : \delta \boldsymbol{E} dA^i = 0, \tag{3.46}$$

where $\boldsymbol{S}^i$ is the 2nd Piola-Kirchhoff stress tensor acting on the updated reference geometry and $\boldsymbol{E}$ the Green-Lagrange strain tensor. The updated reference geometry of step (i) with surface area A^i is defined as the intermediate equilibrium shape of the previous step $(i-1)$. If the stress tensor $\boldsymbol{S}^i$ was defined from a "pull back" operation of the Cauchy stress tensor $\boldsymbol{\sigma_0}$, equation (3.45) would remain unaltered as well as the singularity with respect to $\boldsymbol{r}$. Instead, $\boldsymbol{S}^i$ is defined by the equivalent components of $\boldsymbol{\sigma_0}$ with respect to the covariant base vectors $\boldsymbol{G}^i_\alpha$ of the updated reference geometry, defined later in the text, which remain fixed in the iteration step (i),

$$\boldsymbol{S}^i = \sigma_0^{\alpha\beta^i} \boldsymbol{G}^i_\alpha \otimes \boldsymbol{G}^i_\beta. \tag{3.47}$$

As a consequence, equation (3.46) is well defined and even linear with respect to $\boldsymbol{s}$ and $\boldsymbol{r}$ and both, the shape and the mesh are uniquely and smoothly modified during the form finding procedure [8]. As the updated reference reaches the final shape, the Cauchy stress defined in the actual configuration at iteration step (i), σ_0^i converges to σ_0.

Applying the method, the free-form shape of figure 3.6(b) is obtained for a predefined stress field $\boldsymbol{\sigma_0}$ and a set of Dirichlet boundary conditions. In this example, the four corner points are fixed and prestressed cables are supporting the edges of the membrane. Furthermore, since equation (3.46) is solved numerically, an initial discretization is required to initiate the procedure (fig. 3.6(a)).

Neither the initial discretization nor the initial geometry influence the final shape obtained by form finding. However, the discretization describing the final shape is highly influenced by the initial one because of the update rule for the reference geometry. The initial mesh is the seed of the iteration sequence. For instance, by changing the initial discretization of the problem of figure 3.6 by adding a refinement (fig. 3.7(a)), while keeping the same set of boundary conditions and prestress, solution of form finding will lead to the same shape, with the difference that the refinement of the initial mesh is maintained in the final one (fig. 3.7). Consequently, the final mesh is influenced by, and

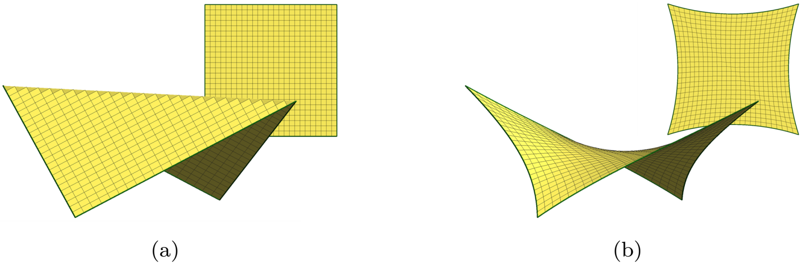


Figure 3.6: Initial discretization (a) and final shape (b) after form finding of a 4-point membrane structure.

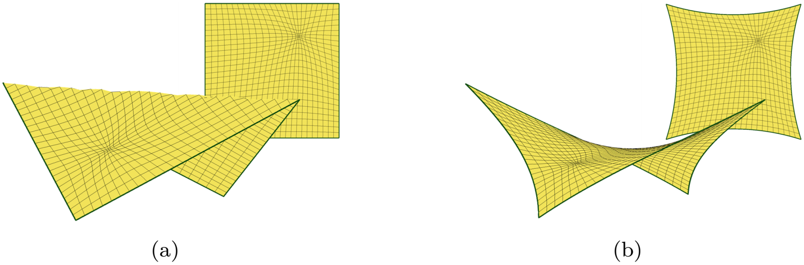


Figure 3.7: Form finding of a 4-point membrane structure with local refinement on the initial mesh.

closely related to the initial one, while the final shape obtained with the method is unique and independent of the initial geometry.

The same principle applies to the mesh regularization method, with the difference that this time the shape has to be retained and the discretization has to be improved. The point of departure is again equation (3.45). Now, a “template” or “ideal” element is defined as the reference configuration and the movement is restricted to the surface directions $\boldsymbol{r}$ since the surface geometry should remain unchanged. The template element describes the desired shape for each individual element and is described in detail in the next section. The shape is retained by

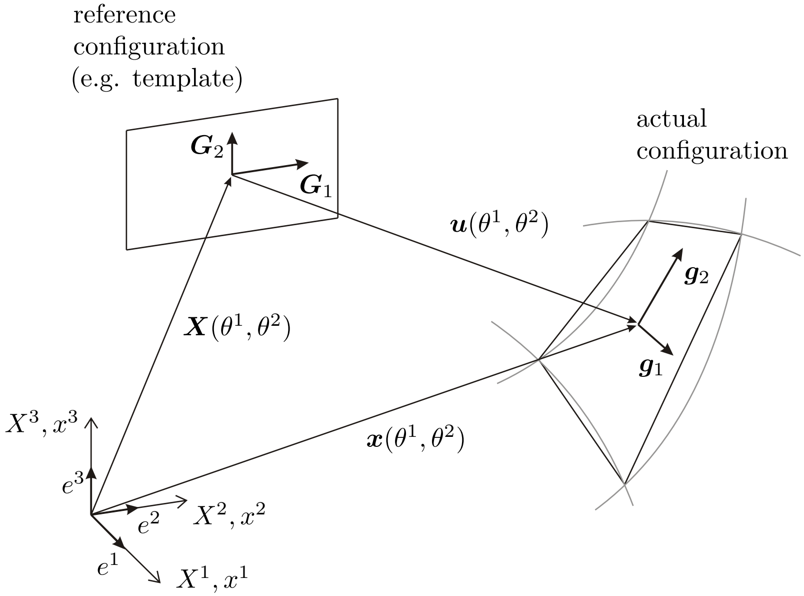


Figure 3.8: Deformation in the context of geometrical nonlinear analysis.

applying additional Dirichlet boundary conditions normal to the surface for each surface node. In this work the multi-freedom constraints were implemented based on the master-slave method.

Applying these constraints and numerically solving the equilibrium, leads to a non-singular system of equations which is linear in the surface tangent space. As a consequence, the proposed method will generate proper meshes even for large distortions of the surface (or volumetric) mesh after solving one linear system of equations. Here, equation (3.46) is solved by the finite element method. For this, the discretization is done in the reference configuration as defined by the template using the isoparametric element concept, [12], i.e. the surface geometry and the displacement field are piecewise approximated by the interpolation of nodal coordinates and displacements, respectively

$$\boldsymbol{X} = \sum_{i=1}^{n} N^i(\theta^1, \theta^2)\boldsymbol{X}_i; \quad \boldsymbol{u} = \sum_{i=1}^{n} N^i(\theta^1, \theta^2)\boldsymbol{u}_i, \tag{3.48}$$

where N^i are the standard shape functions and n is the number of nodes (fig. 3.8). The nodal displacements $\boldsymbol{u_i}, i = 1, .., n$ are the unknowns of equation (3.46) and represent in this case the change of the mesh relevant tangential variable $\boldsymbol{r}$ and the amount that each node should displace in the tangential space in order to regularize the mesh. Consequently, the position of a surface point in the actual configuration is defined as

$$\boldsymbol{x} = \sum_{i=1}^{n} N^i(\theta^1, \theta^2)(\boldsymbol{X}_i + \boldsymbol{u}_i). \tag{3.49}$$

In this context, the base vectors on the reference and on the actual configuration, respectively are

$$\begin{aligned} \boldsymbol{G}_\alpha &= \frac{\partial \boldsymbol{X}}{\partial \theta^\alpha} = \sum_{i=1}^{n} \frac{\partial N^i(\theta^1, \theta^2)}{\partial \theta^\alpha} \boldsymbol{X}_i; \\ \boldsymbol{g}_\alpha &= \frac{\partial \boldsymbol{x}}{\partial \theta^\alpha} = \sum_{i=1}^{n} \frac{\partial N^i(\theta^1, \theta^2)}{\partial \theta^\alpha} \boldsymbol{x}_i \quad \alpha = 1, 2. \end{aligned} \tag{3.50}$$

On this basis, the 2nd Piola Kirchhoff stress and the Green-Lagrange strain are defined in the reference configuration accordingly

$$\boldsymbol{S} = S^{\alpha\beta} \boldsymbol{G}_\alpha \otimes \boldsymbol{G}_\beta, \tag{3.51}$$

$$\boldsymbol{E} = E_{\alpha\beta} \boldsymbol{G}^\alpha \otimes \boldsymbol{G}^\beta = \frac{1}{2}(\boldsymbol{g}_\alpha \cdot \boldsymbol{g}_\beta - \boldsymbol{G}_\alpha \cdot \boldsymbol{G}_\beta) \boldsymbol{G}^\alpha \otimes \boldsymbol{G}^\beta \quad \alpha, \beta = 1, 2. \tag{3.52}$$

Hence, taking the variation of equation (3.46) with respect to the unknown displacements $u_m, m = 0, .., dofs$ and substituting equations (3.51) and (3.52) to (3.46), it can be rewritten for each degree of freedom m of the system as

$$t \int_A \boldsymbol{S} : \frac{\partial \boldsymbol{E}}{\partial u_m} dA = t \int_A S^{\alpha\beta} \frac{1}{2}(\boldsymbol{g}_\alpha \cdot \boldsymbol{g}_{\beta,m} + \boldsymbol{g}_\beta \cdot \boldsymbol{g}_{\alpha,m}) dA = 0, \quad m = 0, .., ndofs, \tag{3.53}$$

This equation system with the displacements $\boldsymbol{u}_i$ as the unknowns is linear. The resulting stiffness matrix is

$$K_{mn} = \int_A \boldsymbol{S} : \frac{\partial^2 \boldsymbol{E}}{\partial u_m \partial u_n} dA \ , \quad m, n = 0..ndofs, \tag{3.54}$$

whereas the right hand side is

$$R_m = \int_A \boldsymbol{S} : \frac{\partial \boldsymbol{E}}{\partial u_m} dA \ , \quad m = 0..ndofs. \tag{3.55}$$

The unknown of this system is the update of the displacement $\Delta \boldsymbol{u}_i$ and since the initial system is linear, the problem is solved in one step. In the case that the prescribed stress tensor is symmetric, the stiffness matrix and the right hand side reduce to

$$\begin{aligned} K_{mn} &= \int_A S^{\alpha\beta} \cdot (\boldsymbol{g}_{\alpha,m} \cdot \boldsymbol{g}_{\beta,n}) dA \\ R_m &= \int_A S^{\alpha\beta} \cdot (\boldsymbol{g}_{\alpha,m} \cdot \boldsymbol{g}_{\beta}) dA \quad m,n = 0..ndofs. \end{aligned} \tag{3.56}$$

In the above equations $S^{\alpha\beta}$ are the components of the predefined 2nd Piola-Kirchhoff stress tensor and $\boldsymbol{g}_{\alpha,m}, \alpha = 1,2$ are the derivatives of the base vectors of the actual configuration with respect to the displacement of the m-th degree of freedom. However, the integration is done in the reference configuration. In the reference configuration each element is assumed to have an ideal shape, i.e. its template. The individual templates are acting independently and there is no need that they match geometrically. For instance, as described also in the next section, the template can be a unit square for each element. Consequently, the calculation of the base vectors $\boldsymbol{G}_\alpha, \alpha = 1,2$, the 2nd Piola-Kirchhoff stress tensor, the Green-Lagrange strain as well as the area A are evaluated based on this template element.

As it can be seen from equation (3.56) there is no dependency of the stiffness matrix to the unknown displacement field since the stiffness matrix depends only on the constant predefined 2nd Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and on the derivatives of the base vectors with respect to the displacements which are constant values as well. Thus, the problem of in-plane regularization is linear.

As it was discussed earlier, when the method is applied to three-dimensional surfaces, additional Dirichlet boundary conditions have to be enforced in order to retain the shape. In practice, each node is restricted to move in the tangential plane to the surface at that point. The tangential plane is calculated by a weighted averaging of the normals of all the elements sharing the node. Here, each normal is weighted by the inverse of the element area. This averaging results in a more accurate normal direction compared to equally weighted averaging. For instance in a 2D case, the average of the weighted normals of each element around a node points exactly to the center of a circle passing through the discretization points. Of course, this is only an approximation and since the movement of each node during the process is finite, the nodes will deviate from the original surface. However, the true surface is usually not analytically given. These deviations are automatically corrected during shape optimization when the shape in normal direction is simultaneously modified. Moreover, if needed, higher order local approximations of the surface can also be used which of course will damage the linearity of the problem. In case of corners and sharp edges the nodes are fixed since these areas should be regarded as features of the initial design which should be retained during regularization.

Furthermore, in the special case of a square initial template for all the surface elements and an isotropic stress state, the problem reduces to the linear Amsden-Hirt method described earlier. One can easily conclude this statement by comparing the variational form of the Amsden-Hirt generator given in equation (3.57) and the variational form of the in-plane regularization problem

$$U = t \int_A \boldsymbol{S} : \boldsymbol{E} dA. \tag{3.57}$$

Hence, the method can be regarded as a generalization of the Amsden-Hirt method which offers two additional mesh control parameters: the ideal (target) element shape, which can be individually assigned for each element, and the prestress controlling the size or concentration of the elements. These control parameters are discussed in detail in the following section.

3.4.5 The template element

The kernel of the in-plane regularization method is the proper choice of adequate element reference geometries or “element templates” and the choice of the applied prestress. These are the two handles used during optimization to control the mesh. The template is the ideal shape of each element defined in the reference configuration. Obviously, for a given surface discretization there will not be any modification if the actual element geometries are taken as templates assuming a unit stress tensor. In any other case, each element will change its shape towards its template shape in a global sense.

The use of different templates and the role of the predefined stress field

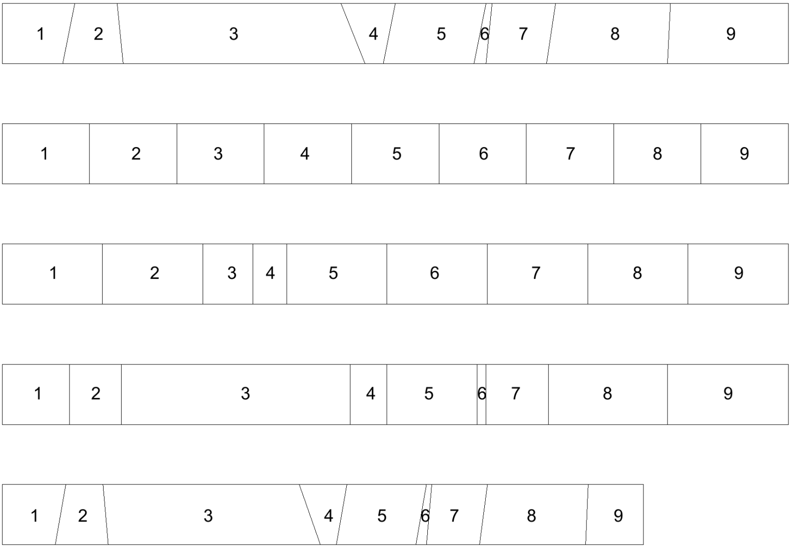


Figure 3.9: Demonstration of the in-plane regularization method for a simple 9-element example. From top to bottom: initial mesh, regularization with square templates, regularization with square templates with local refinement, regularization with shearless templates, regularization with initial templates in a moving boundary problem.

$\boldsymbol{S}$ can be demonstrated with the 9-element example of figure 3.9. In this example the nodes are placed in an irregular distance from each other. Keeping the geometry of the boundaries by applying proper boundary conditions and applying in-plane regularization using square templates and a unit stress field will lead to the second figure from the top of figure 3.9. Here, the shape of the elements changes in order to approach the square template in a global sense, conforming the Dirichlet boundary conditions. Since during this process no element is preferred, the elements cover the same distances over the span of the geometry and the skewness is automatically corrected.

Changing the applied stress field changes the relative size of elements. In the third figure from top of figure 3.9, elements 3 and 4 have three

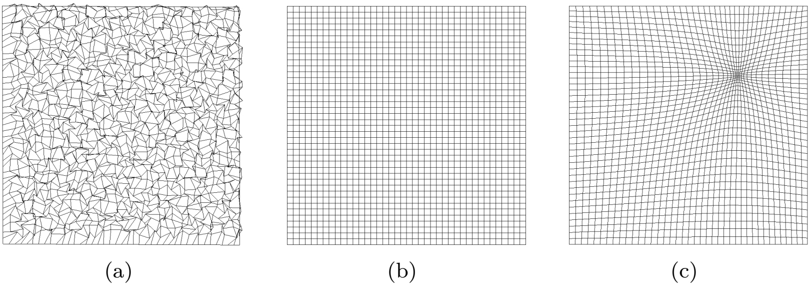


Figure 3.10: Noise removal and local refinement around a point.

and two times more stress, respectively than the rest of the elements. As a result, these elements will cover three and two times less distance in the spanwise direction of the domain respectively. In general, higher predefined stress will cause elements to shrink which is a property which can be applied to achieve local refinements on the boundary or even inside the computational domain.

A more local effect is observed by using a rectangular template with edges as large as the midsegment of each element. This individual template for each element corrects the skewness locally (fig. 3.9). Obviously, stress adaptation in desired areas can also be achieved.

In evolutionary problems, like in shape optimization or problems with moving boundaries, the goal is to retain the quality of the initial mesh which usually possesses certain properties like local refinements, boundary confined element layers and certain growth ratios between the elements. In this type of problems the initial shape of each element is regarded as the template and throughout the evolution of the process the elements retain their shape and relative size independent of the number of steps required to arrive at the specific deformed state. For instance, by moving the right boundary of the initial example of figure 3.9 together with this type of template results to the bottom mesh of figure 3.9. It is obvious that after the deformation the elements keep their initial properties.

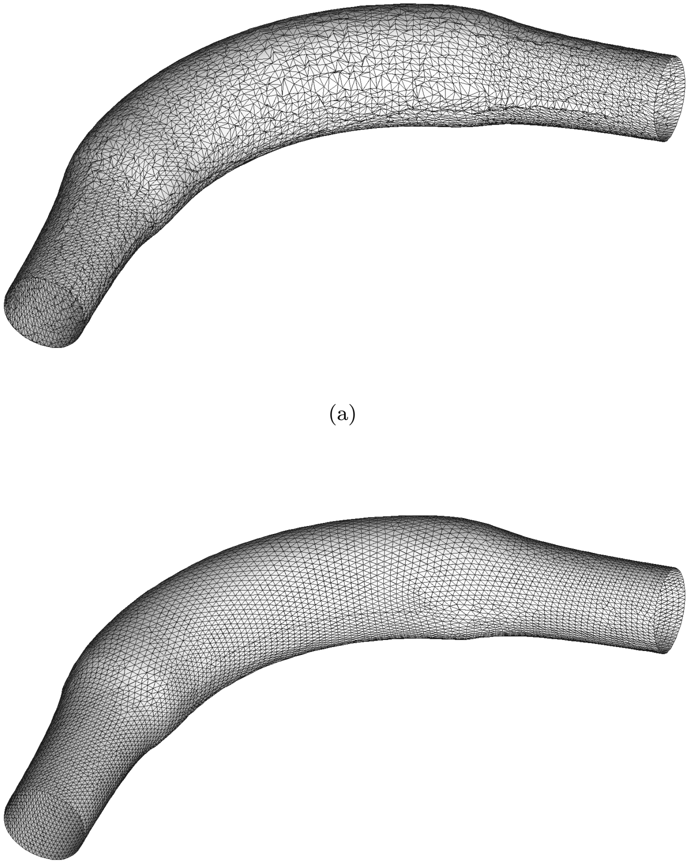
The method applies to more dimensions. For instance, in figure 3.10(a) a square template and a unit stress tensor is applied as reference geometry to remove the distortion (noise) from a two-dimensional plane mesh consisting of quad elements. Figure 3.10(b) shows the mesh after regularization, i.e. the solution of a linear system of equations. The same square template was used for every element and the resulting mesh consists of elements which match perfectly the predefined template. In this particular case the topology of the mesh allows the elements to reach their template shape. In general, this is not the case and thus the method results in a global compromise after this linear step. As it has been already shown for the one-dimensional case, changing the applied local prestress, local refinements can be achieved. In figure 3.10(c), the isotropic stress decrease when the distance to the refinement point increases. In the same way, a proper refinement for boundary layer resolution in fluid problems can be achieved with one linear step starting from any initial mesh.

A three-dimensional surface noisy mesh of a pipe is depicted in figure 3.11(a). Here, an equilateral triangle, the same for all elements was used to regularize the mesh (fig.3.11(b)). In general for a 3D case, additional boundary conditions have to be applied in order to retain the shape. However, in most of the cases the shape is not explicitly given and only an approximation of it can be estimated through the discretization. In this work, the nodes are allowed to move only on the tangential direction to the surface and they are restricted in the normal direction. The node normal $\boldsymbol{n}$ is computed as the weighted average of the normals $\boldsymbol{n}_i, i = 1, \dots, n_{ngh}$ of the elements sharing the node as follows

$$\boldsymbol{n} = \frac{\sum\limits_{i=1}^{n_{ngh}} \dfrac{\boldsymbol{n}_i}{A_i}}{\sum\limits_{i=1}^{n_{ngh}} \dfrac{1}{A_i}}, \tag{3.58}$$

where A_i is the area of the element. In practice, this approximation has shown good results for shape optimization applications.

Moreover, as in the one-dimensional case, by assuming for each element a rectangular template with edges as large as the midsegment of each element, the skewness of elements can be reduced. In the next example, the mesh of figure 3.12(a) created by a mesh generator will be improved using this template. The elements of this mesh are not severely distorted as in the previous example but still there is a group of elements around



(a)

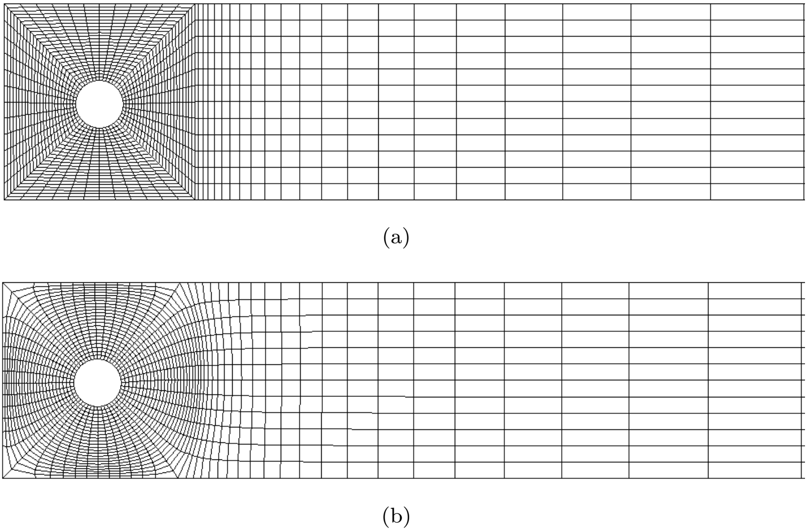
(b)

Figure 3.11: Noise removal from a 3D surface triangular mesh.

the boundary circle which are not well aligned to the boundary. This type of non-orthogonality is a remarkable source of error in numerical simulations, especially in the finite volume method. In the resulting mesh shown in figure 3.12(b), the first layers of elements around the circle are following the curved lines of the boundary. In figure 3.13 the distribution of the angle of distortion is plotted. Here, the angle of distortion is defined as $|90 - \phi|$, where ϕ is the angle between the midsegments of the quadrilateral. It can be observed that the mesh quality is significantly improved after regularization while more elements have small distortion angles and the elements with bigger distortion angles are reduced. The method can not guarantee non-folded grids when used for mesh smoothing like in this example but in many cases it can be used to improve the quality of the mesh with a proper choice of individual templates. For problems with sharp reentrant corners like the ones presented in the Rouge's Gallery [70], nonlinear procedures are required and the reader is refered to [52].

These limitations do not apply in the case of evolutionary processes which begin with an initial high quality mesh which gets distorted during the computation. For this type of problems, the initial shape of each element can be used as the template as described for the one-dimensional case. Examples of such problems are CFD simulations with moving boundaries or fluid-structure interaction problems.

As commonly observed in engineering practice, the increase in the curvature of the surface mesh can have severe effects on the mesh quality. In figure 3.14(a) the resulting mesh of an initially flat surface, after applying a displacement field normal to the surface in 100 iterations is shown. This can be the case, for example, in a metal forming process or the inflation of a rubber membrane as anticipated in figure 3.14. The resulting elements are elongated, even though, the initial mesh had a good quality consisting of square elements of the same size. Applying the regularization method together with the initial shape of each element as template, the elements of the deformed shape still have very good aspect ratios (fig. 3.14(b)). Moreover, since this method regularizes the mesh globally, there are no more elements with high aspect ratios, for the price that some of the elements with perfect initial aspect ratios were slightly deformed (fig. 3.15). This is exactly what is desired in shape optimization, because the low quality elements are the ones which limit the computation. The transmission from a flat or



(a)

(b)

Figure 3.12: Regularization of a structured mesh with a hole.

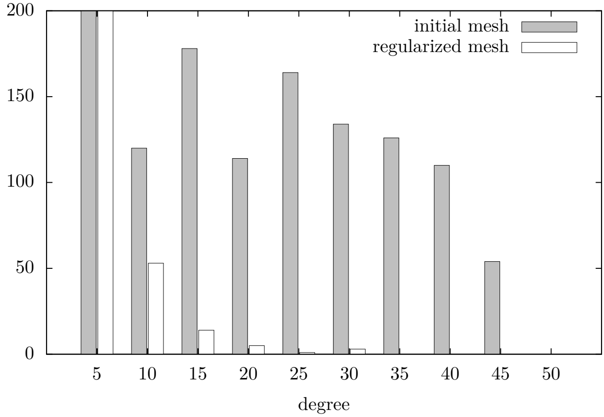
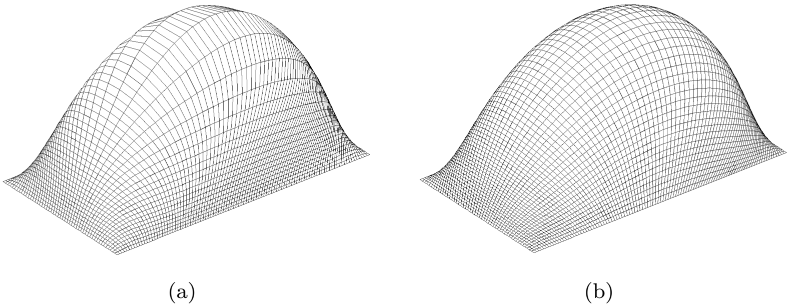


Figure 3.13: Distribution of angle of distortion in initial and regularized mesh.



(a) (b)

Figure 3.14: The resulting mesh after a curving process without and with regularization, respectively.

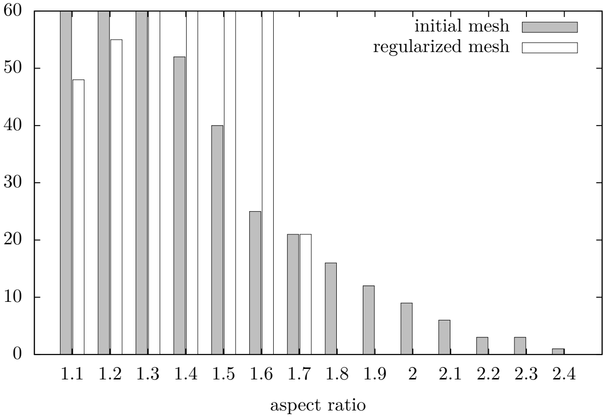


Figure 3.15: Distribution of element aspect ratio (ratio of midsegments) on final design with and without regularization.

slightly curved surface to highly curved one is a common incident in node-based optimization as well.

A more critical process appearing also very often in shape optimization is the reduction of curvature of the boundaries and is more often encountered in the shape optimization of internal flows. In this kind of applications, the update of the shape takes place usually only on the normal direction and depending on the size of the update the elements get closer to each other and often overlap as described in figure 3.16. In this figure, the curved wall is updated normal to the surface and two cases were studied without and with In-plane regularization depicted in figures 3.16(b) and 3.16(c), respectively. In this point the use of the method is decisive for the optimization because the elements overlap and an evaluation of the objective, if possible, would not be accurate.

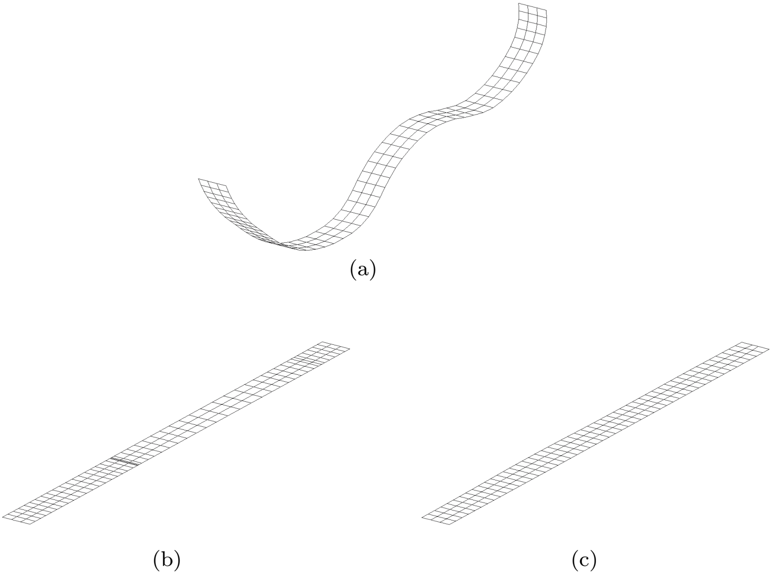


Figure 3.16: Reduction of the curvature of a 2D mesh (a) to a plane mesh without (b) and with (c) regularization.

From all the above, it follows that the In-plane regularization method is necessary for evolutionary processes with large design updates. Furthermore, the choice of the template and the prestress is decisive for each application. For optimization problems the most suitable template is the one coinciding with the initial shape of each element.

Chapter 4

CFD shape optimization

In the previous chapters the theoretical framework and the basic elements of node-based shape optimization were discussed. More precisely, in Chapter 2 the adjoint sensitivity analysis for a structural, fluid or coupled problem is presented while Chapter 3 is concentrated on the treatment of the design surface and the maintenance of the quality of the shape and mesh during shape optimization. As it will be shown in the sequence, these are the key elements for performing efficiently gradient-based, parameter-free shape optimization. Having the mentioned elements of the shape optimization chain, the next challenging task is to combine the individual modules into a framework with suitable interfaces and data structures.

Thus, the current chapter is organized as follows: In Sections 4.1 and 4.2 the computational framework of a node-based shape optimization as well as the structure of the actual codes are discussed. The optimization chain is presented and each of its features is explored in detail. In the next sections industrial applications supporting the efficiency and feasibility of the chosen methodology are shown. More precisely, in Section 4.3 the optimization with respect to power dissipation of a rigid three-dimensional industrial duct is described. The focus in this application is on the necessity of the smoothing methods presented in Chapter 3. In Section 4.4 the parameters of the vertex-morphing method and their influence on the optimization problem are studied on a car engine intake geometry. The fluid mesh of this example at the wall layer is a low-Reynolds mesh with prism layers of y^+ less than 5. In these cases additional effort has to be given on the update of the volume mesh. The procedure which is followed is also presented.

4.1 Computational framework

The formulation of the optimization problem is already discussed in Section 3.1. Generally, there are two approaches for the solution of such a PDE-constrained optimization problem: The "simultaneous analysis and design", SAND and the "Nested Analysis and Design" approach, NAND. In SAND, the PDE describing the equilibrium of the physical system is considered as an equality constraint to the optimization problem and the state variables of the problem are treated also as optimization variables. As a result, in each optimization iteration the equilibrium equation is not necessarily satisfied. In contrary, in NAND approach the equilibrium equation is satisfied in each optimization iteration and only the design variables are regarded as the optimization variables of the problem. An overview of the two approaches can be found in [3].

In this work, the NAND approach is applied. In this case, the general optimization workflow for parameter-free, gradient-based shape optimization is presented in figure 4.1. The procedure is generally iterative and the optimizer is the driving force of the overall loop. As shown in the figure, initially, the optimizer suggests the updated values of the design variables. Then, according to these values the geometry is modified. On the new geometry, the primal equations are not in equilibrium. Hence, as a next step, the primal equations are solved. Having a converged primal state, the sensitivities can be computed. Finally, according to the new values of the sensitivities the optimizer suggests a design update and this loop repeats until the convergence criterion is met. In the sequence, the overall process is described in detail by concentrating on each of the four important elements of the overall optimization workflow (figure 4.1). The actual codes as well as the overall implemented framework used in this work are described in

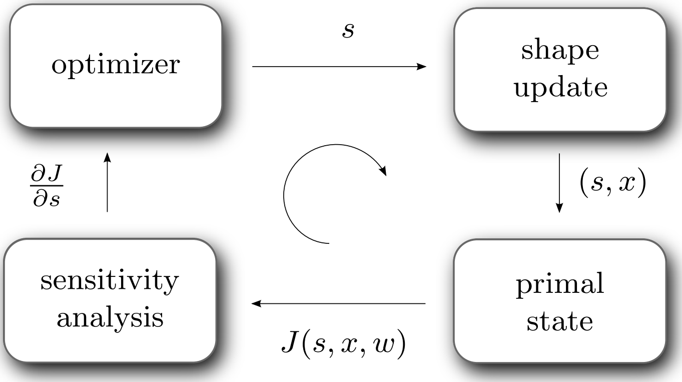


Figure 4.1: Optimization workflow

the next Section.

4.1.1 Optimizer

In gradient-based shape optimization the optimizer decides for the update of the design variables based on the gradient information provided by the sensitivity analysis. According to the order of derivative information needed by the optimization algorithm one can distinguish between zero-order, 1st-order and 2nd-order methods. Most of the methods require a starting value for the design variable s_0 and in each iteration the solution is updated as follows:

$$\boldsymbol{s}_{n+1} = \boldsymbol{s}_n + \alpha_n \boldsymbol{S}_n, \tag{4.1}$$

where $\boldsymbol{S}_n = \boldsymbol{S}_n(\nabla J(s_n); H(s_n)) \in \mathbb{R}^n$ is the search direction in the design space and α_n is the step size which is usually calculated with a line-search algorithm. For the various algorithms available one can refer to [2, 125].

In this work for the unconstrained optimization problems presented later in this chapter the Steepest Descent algorithm is used mainly due to the large problem size and the type of design variables as explained later in the section. In 1847 Cauchy observed and reported the property of the "steepest descent" of the negative gradient direction. It can be shown that the negative gradient direction

$$\boldsymbol{S} = -\nabla J(\boldsymbol{s}), \tag{4.2}$$

of a multivariant, continuous and differentiable function $J : \mathbb{R}^n \rightarrow \mathbb{R}$ is exactly the direction where the values of $J(\boldsymbol{s} + a\boldsymbol{S})$ decrease the fastest and this exactly is the search direction that the algorithm is using.

Generally, Steepest Descent is a simple minimization algorithm which is efficient when the initial guess is far from the minimum but by getting closer to the minimum it converges rather slowly. This algorithm was chosen for various reasons. Firstly, the Vertex-Morphing method, used for the linking between the design variables and the geometry, is a generalized type of a kernel filter and thus an analog to the implicit Sobolev-gradient methods. These methods are already enhanced by some approximation of the Hessian matrix. Moreover, due to the large number of design variables, the explicit evaluation of the Hessian would be very costly [56].

Furthermore, from a geometrical view point, the design variables are defined on a local space normal to the surface at each point. Thus, the direction of the update for each design variable is approximated by the node normal in every optimization iteration. Hence, a higher order algorithm would require the storage of the deformation gradient at each surface node at each optimization step. The Steepest descent algorithm does not require any higher order information or the geometrical history of the design variables.

Moreover, from a computational point of view, the use of a higher order algorithm would not affect the efficiency significantly. The vast computational time at each optimization step is spend on the convergence of the primal and adjoint solutions. A smaller design update results in a faster convergence and the cost is almost proportional to the total design update and not on the size of the optimization steps itself.

4.1.2 Shape update

The update of the shape comes after the optimizer evaluates the improved set of design variables. At this stage, the whole computational mesh needs to be updated according to the design variable values and its quality needs to be retained. In the general case, the mesh contains the design surface mesh and the volume mesh attached to the surface. In the Vertex-morphing method the geometry of the design surface is linked to the design variables through equation (3.15). Thus, the update of the surface geometry given the value of the design variables is

$$\delta x(\xi_0) = \int_{-1}^{1} A(\xi_0 - \xi)\delta s(\xi) d\xi. \tag{4.3}$$

At this point experience shows that the quality of the surface mesh is already good enough because of the construction of the operator A that the method suggests. In case additional regularization is required the surface mesh is regularized with the In-plane regularization method presented in Section 3.4. As discussed earlier, the In-plane regularization method improves the quality of the surface mesh by applying an artificial stress field on the surface and a global linear system for the equilibrium is solved. The discrete system is obtained by finite elements.

After updating the design surface, the volume mesh, if any, is adapted in order to conform to the new boundary geometry. While various different methods can be explored for this purpose - from remeshing to fixed-grid approaches - in this work the mesh is deformed by solving the Laplace equation for a defined variable diffusion γ

$$\nabla \cdot (\gamma \nabla \boldsymbol{u}) = 0 \tag{4.4}$$

for the velocity $\boldsymbol{u}$ of each point, as described in Section 3.4.3 and originally in [65].

4.1.3 Primal state and sensitivity analysis

The primal state and sensitivity values should be reevaluated after a change of the geometry. In the applications following this section steady-state CFD and fluid-structure interaction primal states are examined and the adjoint method is used for the evaluation of the sensitivities required by the Steepest Descent algorithm. More precisely, for the CFD applications the steady-state incompressible Navier-Stokes equations (2.18) are solved using the SIMPLE algorithm presented in Appendix (B.1). The continuous adjoint approach for such a problem is developed by Othmer in [96, 98] and shown in Section 2.2.2. Concerning the fluid-structure interaction application of Chapter 5 a detailed explanation of the primal and adjoint equation is given in Section 2.3.2.

In the applications presented in the sequence the iterative CFD primal and adjoint calculations consume the most of the computational time. Experience has shown that the use of not fully converged primal and adjoint variables gives satisfactory results since the values are normalized and only a search direction is needed for the small increment of an optimization step. Moreover, by using a Steepest Descent algorithm with constant and relatively small step size and by initializing the computation of each optimization step with the previous converged fields accelerates the overall computation.

4.2 Implementation issues

In the present work the following two software packages are used: OpenFOAM, the C++ open source toolbox for numerical simulation of continuum mechanics problems and the research in-house code Carat++. The two codes communicate through an MPI interface.

The overall process is controlled by the optimization module of Carat++ which is developed to act as an external optimizer independent of the code which provides the primal solution and the sensitivities. In this module the optimization algorithm which provides the update of the design variables, the Vertex-Morphing method which controls the smoothness of the design surface and the In-plane regularization method designed to control the surface discretization are included.

Moreover, the structural finite element module of Carat++ is used for the evaluation of the structural part of the primal problem as well as the adjoint semi-analytic sensitivites for the coupled fluid-structure interaction computation. OpenFOAM is used for the evaluation of the primal and adjoint flow solutions as well as for the update of the volume fluid mesh. The standard SIMPLE algorithm is used for the evaluation of the primal solution while the implementation presented in [96, 98] provides the continuous adjoint sensitivities. A modification of this algorithm which includes the necessary additional source terms provides the coupled adjoint sensitivities required for the coupled FSI problem.

The interaction of the two codes through MPI as well as the computational workflow is presented in figures (4.2) and (4.3).

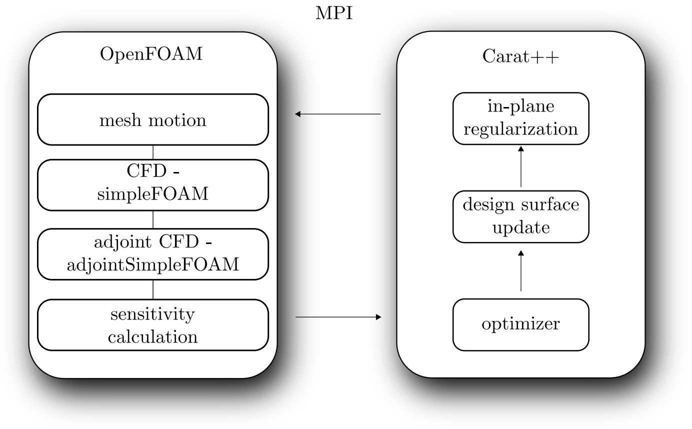


Figure 4.2: CFD optimization workflow.

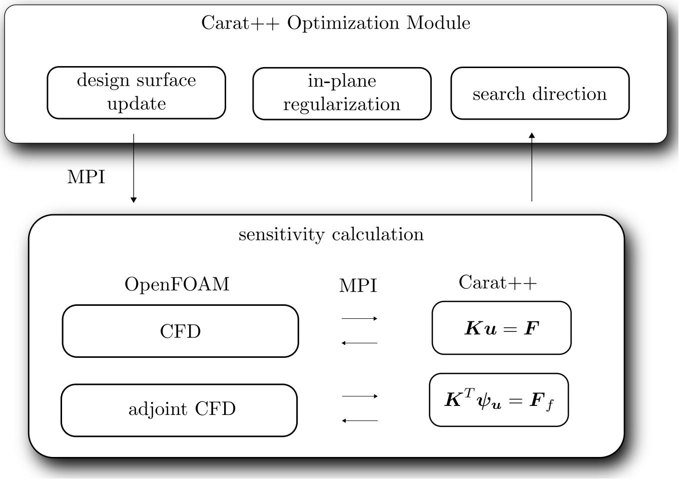

Carat++ Optimization Module
design surface update
in-plane regularization
search direction
MPI
sensitivity calculation
OpenFOAM
MPI
Carat++
CFD
$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}$
adjoint CFD
$\boldsymbol{K}^T\boldsymbol{\psi_u} = \boldsymbol{F}_f$

Figure 4.3: FSI optimization workflow.

4.3 Minimization of power dissipation in an S-bend 3D duct

In this section the in-plane regularization method presented in Section 3.4 is successfully applied to the shape optimization of the three-dimensional industrial S-bend of figure 4.4. The geometry is chosen from a series of test cases provided by Volkswagen AG within the frame of the "7th Framework Programme" EU-project, FLOWHEAD (Fluid Optimisation Workflows for Highly Effective Automotive Development Processes). Therefore, it reflects the up to date challenges and needs in CFD shape optimal design in automotive industry.

The air duct used for rear seat ventilation is studied in [97]. The design surface is depicted with dark gray in figure 4.4 and has 8324 nodes and thus design variables $\boldsymbol{s}$.

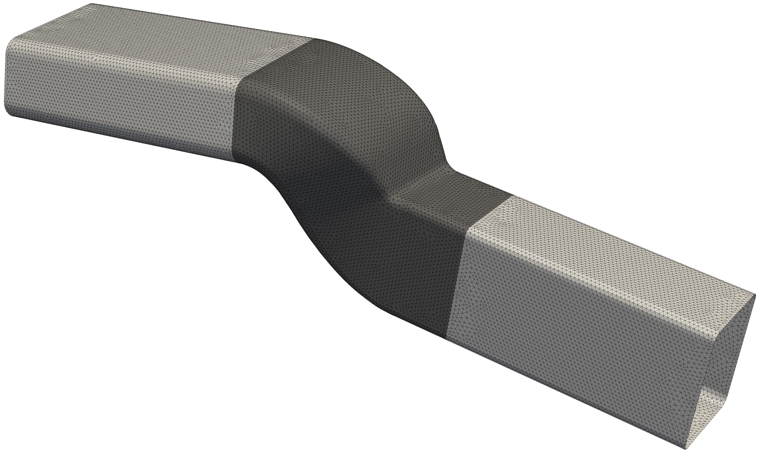


Figure 4.4: Initial mesh of S-bend with 2081346 finite volume cells. The design surface is marked with dark gray [97].

The objective function is the dissipated power defined earlier in Section

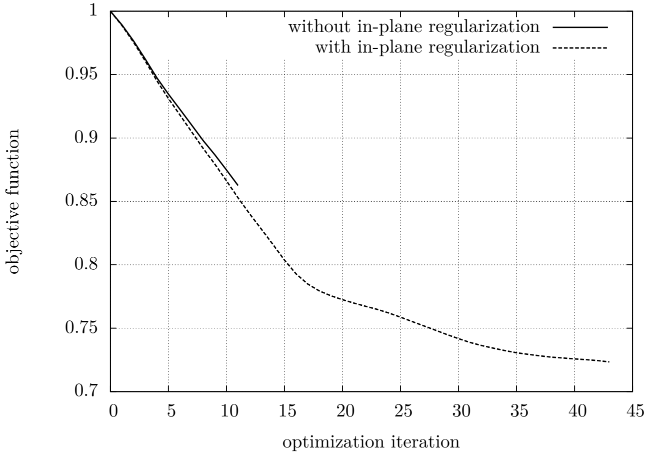


Figure 4.5: History for the normalized power dissipation with and without in-plane regularization.

2.2.2:

$$f = -\int_{\Gamma} (p + \frac{1}{2}\rho \boldsymbol{v}^2)\boldsymbol{v} \cdot \boldsymbol{n} d\Gamma, \tag{4.5}$$

where p and $\boldsymbol{v}$ are the pressure and the velocity, respectively whereas $\boldsymbol{v} \cdot \boldsymbol{n}$ is the normal component of the velocity and Γ the fluid boundary.

The procedure followed for the shape optimization of this duct is as described in Section 4.1. A detailed information about the software used is found in the previous section. In brief, for the shape optimization a kernel filtering on the normal update is applied, described in Section 3.2. The filter radius is about 10 times the element size. In addition the In-plane regularization method of Section 3.4 is used to retain the quality of the surface mesh of the design surface. Here, the initial shape of each element is taken as template and a unit stress tensor is applied to regulate the size of each element uniformly.

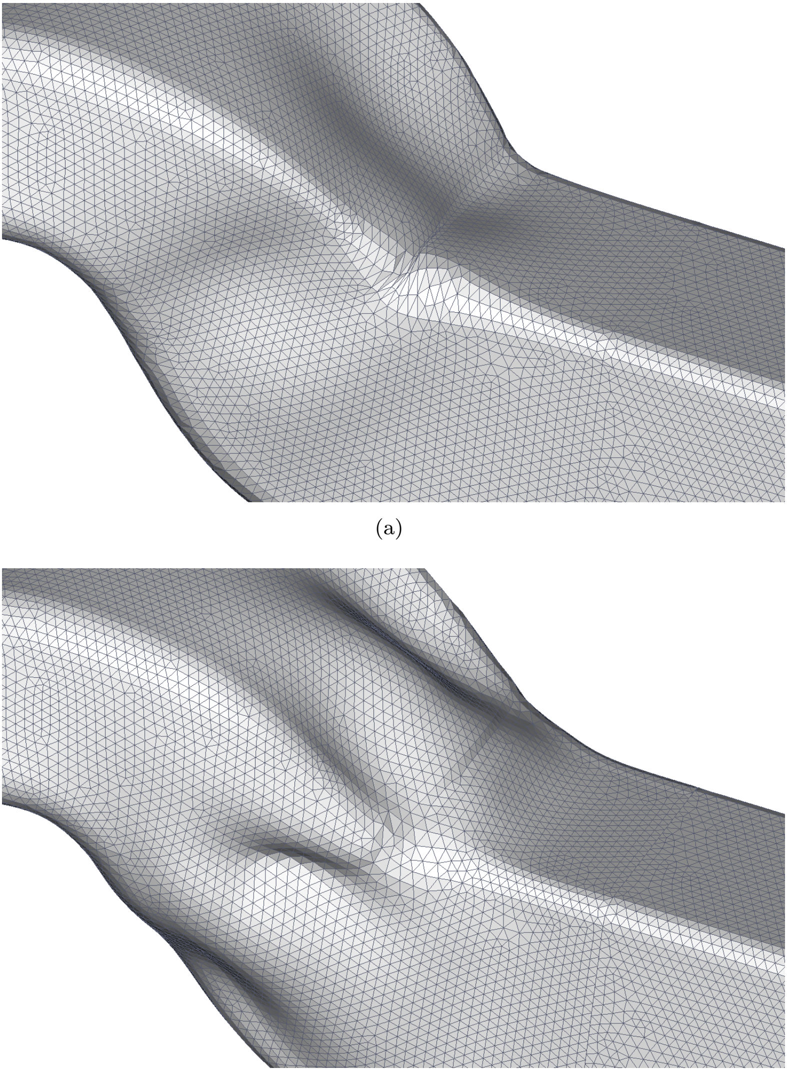
The optimization algorithm is the steepest descent algorithm with constant step. The step size is about the element size. The required sensitivities are evaluated with the adjoint method using the basic structure of OpenFOAM and a code developed in the framework of the

FLOWHEAD EU-project by ICON. The continuous adjoint formulation is based in [96]. At the end of each optimization step the volume mesh is updated by solving the Laplacian equation with variable diffusion [65]. Convergence is reached when the value of the objective function between two successive iterations remains almost unchanged.

Figure 4.5 shows the history of the optimization with and without In-plane regularization. By applying In-plane regularization around 27% of improvement in power loss is achieved. If no regularization is applied the optimization fails already at the 11th iteration because of failure of the mesh of the design surface. This failure can be clearly seen in figure 4.6 where the final designs without and with In-plane regularization are compared. The surface mesh of the S-bend at the 11th iteration suffers from degenerated elements and no further computation is allowed. In contrary, the final mesh when In-plane regularization is applied is smooth and the mesh density and quality is maintained throughout the computation. In this way significant design changes can take place while no kinks, wrinkles or any other mesh irregularities appear. This becomes obvious in figure 4.7 which shows the evolution of the shape at a cross section located at the bend of the duct.

The improvement of the design can be also seen through the sensitivity map on the bend of the duct. Figure 4.8 displays the surface sensitivity at the initial design as well as at the improved design using in-plane regularization. The high sensitivity region undergoes the largest deformation. This area is where the mesh fails when no in-plane regularization is applied. In the final design the high sensitivities are removed while some sensitivity still remains close to the boundary of the design domain since it should be kept unchanged during optimization in order to have a smooth transition from the design to the non design surface.

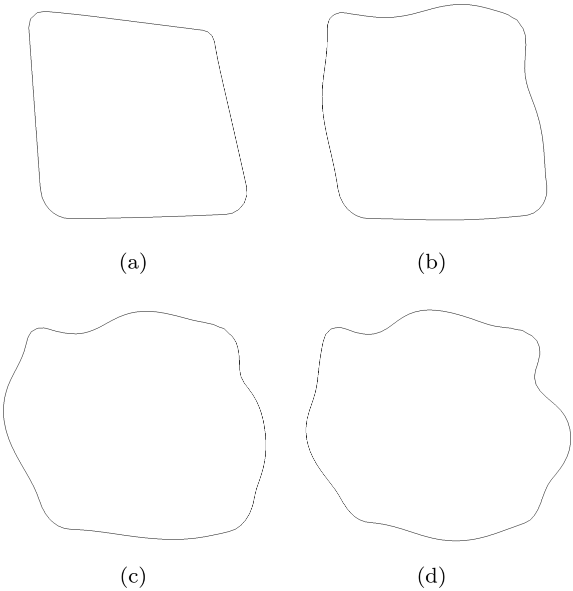
In order to study the mesh quality more quantitatively, the proportional change of area of the surface elements compared to the initial mesh is shown in figure 4.9. The value 0 means that the area of the element has not been changed and the value -1 shows an element with zero area. Note that figure 4.9(b) refers to iteration 43 and figure (4.9(a)) to iteration 11, at which, the case with no in-plane treatment collapses. Although the shape is much more deformed in (4.9(b)), the elements still have much better quality compared to (4.9(a)).



(a)

(b)

Figure 4.6: Shape optimization of S-bend. Failure of the mesh at the 11th iteration without In-plane regularization (a) and final design after the convergence criterion is met (43rd iteration). The improvement of the objective function is 27% (b).



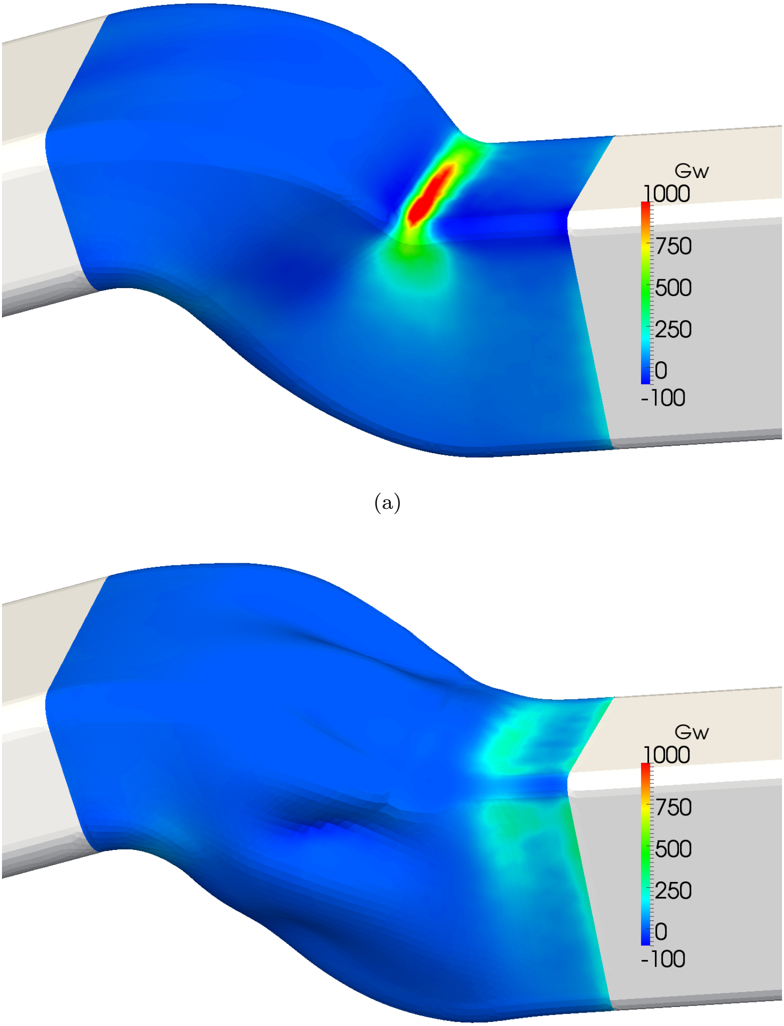
(a)

(b)

(c)

(d)

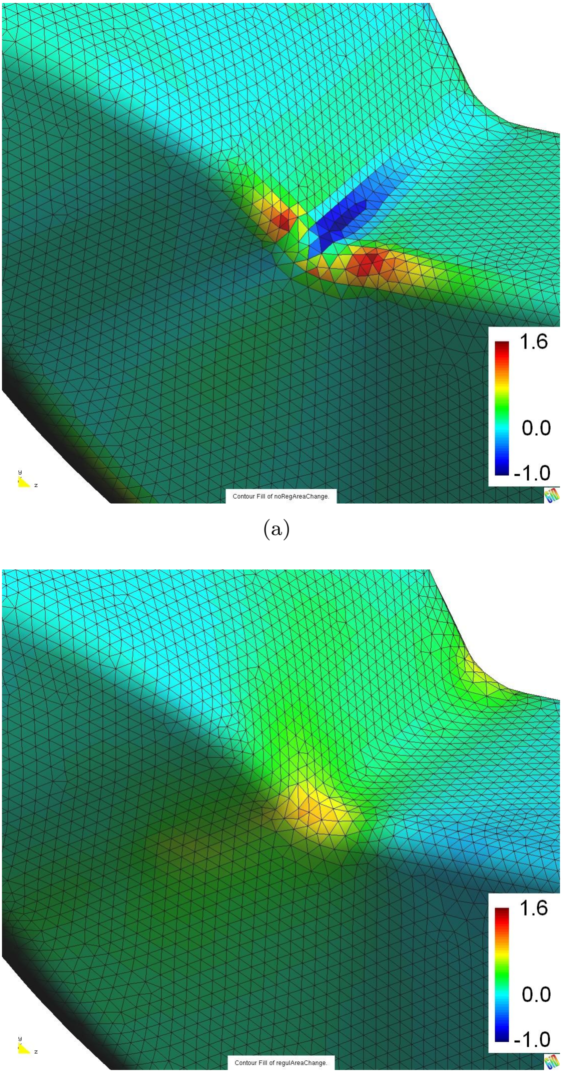
Figure 4.7: Cross sections at the bend for iterations 11, 22, 33 and 43, respectively. The filter radius is about 10 times the element size and In-plane regularization is also applied.



(a)

(b)

Figure 4.8: Surface sensitivity distribution for the initial and final design.


1.6
0.0
-1.0
y
z
Contour Fill of noRegAreaChange.
1.6
0.0
-1.0
y
z
Contour Fill of regulAreaChange.

(a)

(b)

Figure 4.9: Proportional change of element area w.r.t the initial mesh without (a) and with (b) in-plane regularization, plotted on the initial shape.

4.4 A filter and step size study

In the current section the node-based shape optimization of an intake geometry of a car engine is examined. More precisely, the purpose of this example is to discuss two important aspects in such industrial optimization problems. The first aspect is the update of the thin boundary layer in low-Re meshes, which is commonly used in turbulent flows. The second aspect is the influence of two important parameters of the optimization procedure which are the filter radius and the step size of the optimization algorithm.

The function of the part is to bring air to the combustion chamber. The design of this part has a large impact on the efficiency and the performance of the engine and therefore several optimization tasks are involved. For instance, the turbulence and the swirl have to be minimized and the mass flow rate has to be maximized. The actual geometry which is provided by VW for testing the optimization workflow is confidential but it is very similar to an older design shown in figure 4.10. The fluid mesh consists of around 2.5 million cells and the design surface has 15.000 nodes and thus design variables.

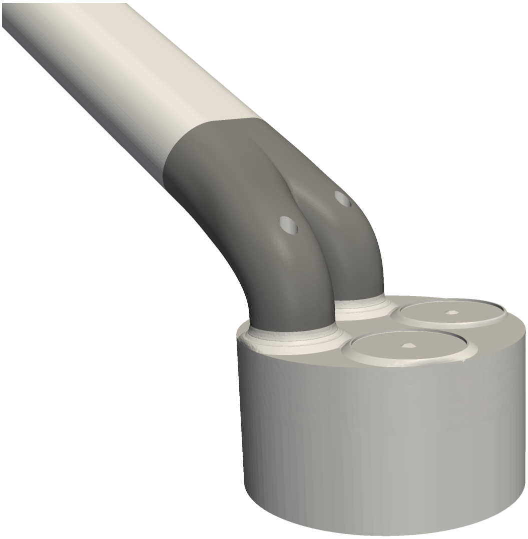


Figure 4.10: Intake Geometry

If the adjoint solver does not use adjoint wall functions, which is often the case, the only way to correctly resolve the adjoint field is to use very thin mesh (prism) layers. This is a common practice in industrial applications and the produced meshes have prism layers with extremely high aspect ratios. Albeit the fact that these meshes can have an initial good quality, small disturbances during optimization can destroy the mesh completely.

An important measure of the mesh quality is non-orthogonality which is usually the first mesh issue which leads to crashing of the numerical solution in the examples of this work. The non-orthogonality is the angle between the line connecting two cell centers and the normal of their common face. It is defined per face and the value 0.0 is the best. Mesh studies have shown that non-orthogonality more than 80-85 degrees is causing failure of the fluid solver.

As discussed before, in shape optimization, after the update of the design surface using the method presented in Chapter 3 the volume mesh is updated by solving the Laplace equation with variable diffusion [65]. Within this volume update, the cells with high aspect ratios close to the wall boundary are very sensitive and the following observations can be made:

- The initial thicknesses of the layers is not preserved: The thickness is getting smaller and in many cases the first cells on the layer collapse.
- Small disturbances of the design boundary cause big angle changes on the short face of the cells on the prism layer which results to big non-orthogonality for these faces.

To overcome this problem a correction of the boundary layer after the mesh update is suggested. The mesh is divided in two areas, the area of the prism-layer and the inner mesh. After the volume mesh deformation, the inner mesh exhibits low non-orthogonality and the problem is concentrated in the prism layers. For this reason, the inner mesh is kept as resulted from the mesh motion algorithm while the prism layers are corrected in a pure geometrical manner. The correction is such that each node is displaced the same amount and in the same

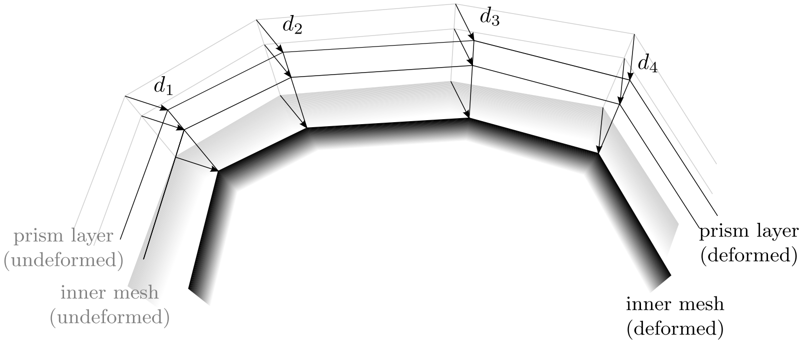


Figure 4.11: Final update of a 2D mesh after mesh motion and correction.

direction as the node on the design surface belonging to the same prism line.

This becomes more clear in figure 4.11. In this figure the two areas of the 2D mesh after the mesh motion and the correction are depicted. The update suggested from the optimizer is $d_1, \ldots, d_4$ and the nodes of each prism line are displaced by the same update.

For the various intake geometries with low-Reynolds meshes that were tested no optimization was possible without the prescribed correction on the prism layers even with a very small step size. Applying the aforementioned procedure made the following shape optimization study possible and as it is seen in the sequence relatively large design updates are achieved.

For this geometry two parameter studies are performed: one for the filter radius and one for the optimization step size, in order to investigate their influence to the improvement of the objective function. Here, the objective function is flow-rate and the adjoint sensitivity calculation is performed by the in-house code of VW. The Vertex-morphing method presented in Section 3.3 and the steepest descent-algorithm with constant step size is applied. The focus in this case is the mesh and geometry robustness and the impact of the parameters to the optimization.

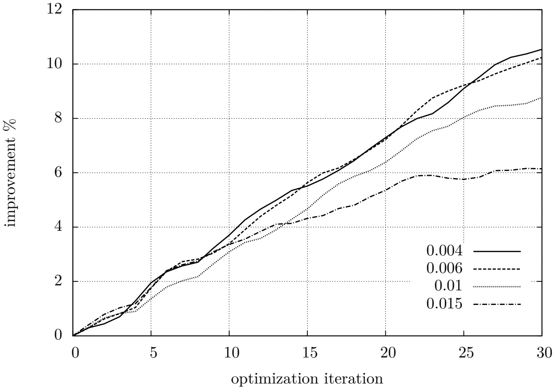


Figure 4.12: A filter radius study for the intake geometry.

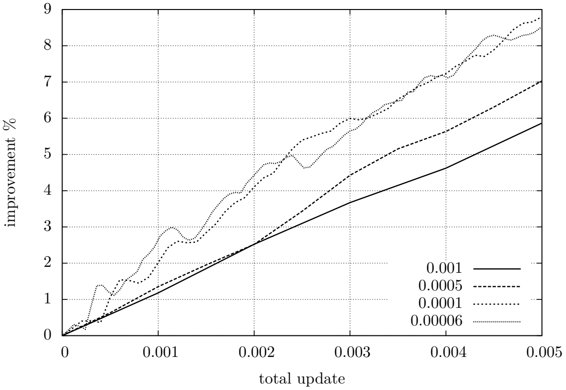


Figure 4.13: A step size study for the intake geometry. The total update is defined as the maximum update length at each iteration multiplied by the number of iterations.

The filter radius is the standard deviation of the Gaussian filter used to model the operator A of the Vertex-morphing method. As already presented in [56], the smaller the filter radius the faster the optimizer approaches the local optimum. This is also the conclusion derived from this industrial case. Of course there are certain restrictions on the size of the filter radius. First, it can not be smaller than a certain limit because then the stability criterion is not fulfilled and shape irregularities appear rabidly. In other words, the ill-posed shape optimization problem which can be solved only by a regularization operator will remain ill-posed due to the small regularization intensity of the operator. Moreover, a limit on the filter size is set from the designers since the final geometries should preserve a certain level of smoothness and a minimum curvature. Figure 4.12 compares the objective improvement for four different filter radii for the first 30 optimization iterations. The final geometry for a certain amount of improvement is similar in all four cases with small differences in the curvatures. As already expected, the rate of improvement of the objective function is the highest for the smallest filter radius.

Furthermore, concerning the influence of the step size of the steepest-descent method on the objective improvement the study of figure 4.13 is performed. It demonstrates the improvement on the objective function as a function of the total design update defined as iterations times step size, for various step sizes. The question that this study tries to answer is how influential is the linearity assumption for the update direction and the gradient vector. Theoretically, because of the linearization of the gradient information a bigger step size would cause the optimizer to deteriorate from the real direction to the local optimum. As seen in figure 4.13 this idea applies also in practical application since for smaller step sizes the improvement rate of the objective is higher. This study also verifies the fact that the computational cost is almost proportional to the total design update and not on the size of the optimization steps itself as already discussed in Section 4.1.

Chapter 5

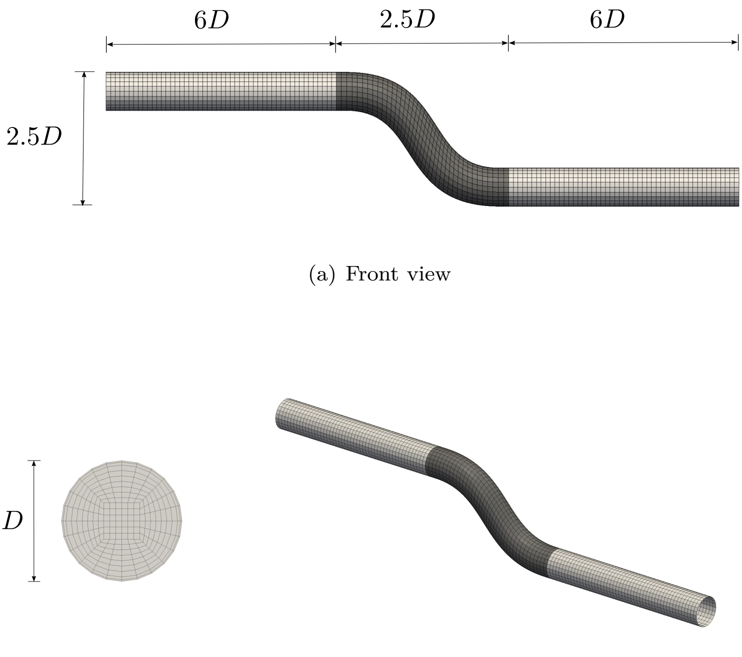
Adjoint coupled optimization

In this chapter the sensitivity analysis and the regularization methods presented in the previous chapters are applied on the node-based shape optimization of a flexible pipe. Firstly, the coupled sensitivity analysis and the verification of the results is discussed. Then, the shape optimization is performed based on the coupled adjoint sensitivity for the flexible part of the pipe using the Vertex-Morphing and the In-plane regularization methods.

5.1 Sensitivity analysis on a flexible pipe

In this section the coupled sensitivity analysis for a three-dimensional elastic pipe which interacts with an internal flow is studied. The primal and the adjoint solutions are presented as well as the computed sensitivities and the results are verified against finite differences.

More precisely, the pipe of figure 5.1 is examined. The pipe has a circular cross section of diameter D and a thickness $t = D/20$. Only the dark gray bend is elastic while the rest of the pipe is considered to be rigid.



(a) Front view

(b) Cross section

(c) 3D view

Figure 5.1: Elastic pipe.

The internal flow of Reynolds number 10, with inlet on the left of figure 5.1(a), exerts a pressure field which causes a deformation to the elastic bend. This deformation changes the internal flow considerably and hence results in an interaction between the fluid and the structure. For the investigation of such a problem a fluid-structure interaction analysis is required. In this case, the system reaches a steady state and no time evolution is considered. The response function here is the power loss between the inlet and the outlet at the converged state. The derivatives of this response function with respect to the normal at each surface point with the coupled steady primal problem as a constraint are of interest.

The coupled primal problem is solved for three different meshes with 10.416, 35.280 and 147.840 fluid cells. The number surface points on the elastic part are 448, 984 and 2600, respectively. For the evaluation of the structure and flow fields with a partitioned way matching meshes are considered.

More precisely, the elastic bend has a radius to thickness ratio of 10 and thus is considered to be a thick shell. The deformation of the bend given the pressure applied by the fluid flow is computed with finite elements using a linear corotational shell element developed in [34, 36, 86]. For the formulation of this shell element additional to the base configuration which is the origin of the displacements in the Lagrangian framework, a corotational configuration is considered. This configuration is obtained from a rigid body motion of the base configuration and the displacements are measured accordingly. Thus, the formulation allows for large element rotations although the strains are considered small.

After computing the deformation of the elastic bend, the fluid discretization is updated accordingly. For this, the mesh motion method described in Section 3.4.3 is used. On the new boundary the fluid equations are not in equilibrium and thus the fluid problem needs to be computed. In this case, the Navier-Stokes equations are solved and the SIMPLE pressure-correction scheme is used [38, 126]. The updated pressure field is applied on the elastic bend and the fluid-structure interaction loop is repeated till convergence. The convergence of the primal with a residual measured on the structural displacements is shown in figure 5.2 and the deformed structure in comparison to the initial (undeformed) bend is shown in figure 5.3.

After the primal calculation the adjoint calculation is performed as described in Section 2.3.2 and the algorithm which is followed is the one of table 1. The dual problem is linear but is solved with a staggered manner. The convergence of this staggered loop is shown in figure 5.4.

Having the converged primal and adjoint fields the sensitivities are evaluated from equation (2.60) repeated here for clarity

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{s}} + \boldsymbol{\psi}^T_{(\mathbf{v},\mathbf{p})} \frac{\partial \mathbf{F}}{\partial \mathbf{s}} + \boldsymbol{\psi}^T_{\mathbf{u}} \frac{\partial \mathbf{S}}{\partial \mathbf{s}}. \tag{5.1}$$

The first term of this equation is zero in the case of power loss response. Moreover, the third term is non-zero only on the elastic bend. For the calculation of the sensitivity field equation (5.1) is evaluated with

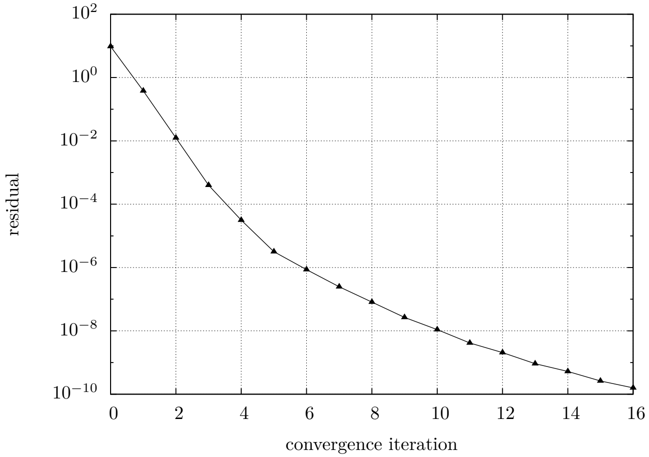


Figure 5.2: Convergence of the steady fluid-structure interaction problem as a part of every optimization iteration.

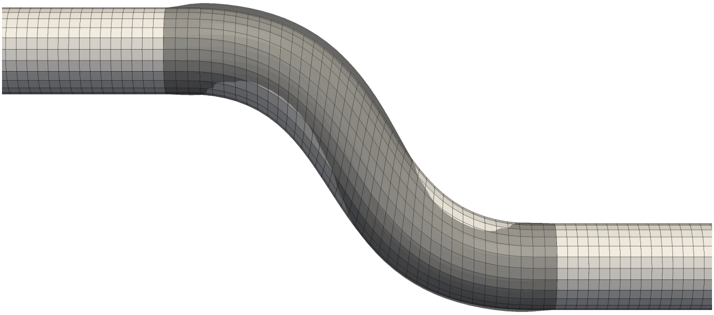


Figure 5.3: Deformation of the bend (dark gray) after FSI convergence.

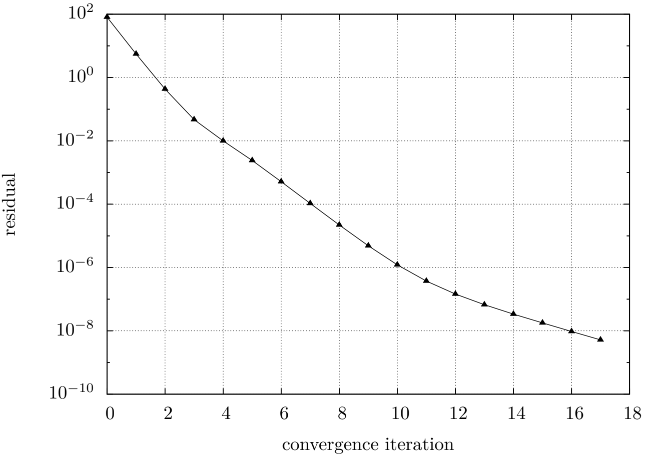


Figure 5.4: Convergence of the adjoint fluid-structure interaction problem.

the coupled adjoint fields computed according to Section 2.3.2. This sensitivity field involves the interaction between the structure and the fluid and the elasticity of the bend is included. In the sequence it will be referred to as the “coupled” or “fully-coupled” sensitivity field.

In order to show the importance and the necessity of the coupled sensitivity analysis the aforementioned sensitivities are compared to the “rigid” sensitivities obtained when the bend is rigid. In this case, the structure does not have any influence on the problem and only a CFD adjoint calculation is required.

Moreover, the term "semi-coupled" sensitivity is defined to be the CFD adjoint sensitivities calculated on the deformed pipe after a fluid-structure interaction. In other words, for the "semi-coupled" sensitivity, the primal problem is treated as a coupled system and the adjoint as a decoupled one, but on the deformed geometry. It will be shown that the “fully-coupled” sensitivities are significantly different than the "semi-coupled" ones.

In figure 5.5 the “rigid”, “semi-coupled” and “fully-coupled” sensitivities are compared. This figure clearly shows that the “rigid” and “semi-

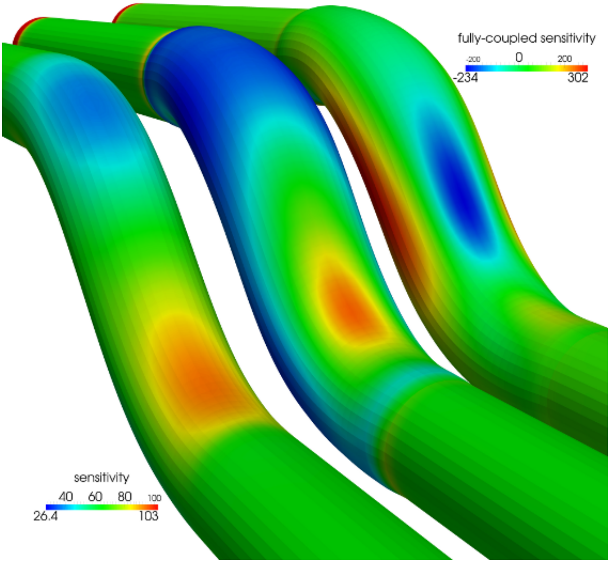
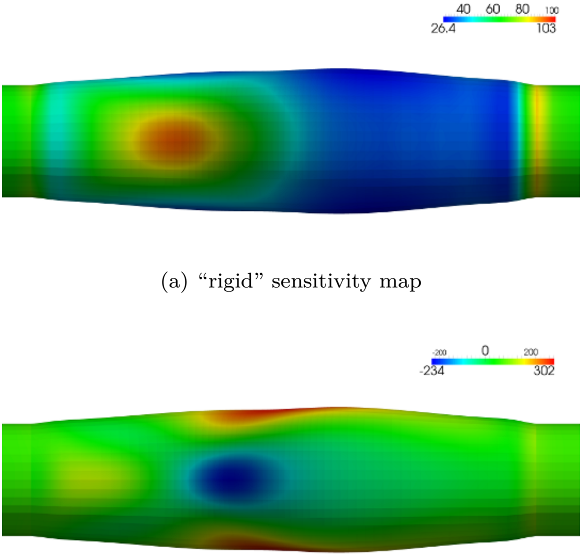


Figure 5.5: From left to right: “rigid”, “semi-coupled” and “fully-coupled” sensitivity maps (note the different color scale).

coupled” sensitivity fields are similar with only their peak values and their influence area to be different. In contrary, the map of “fully-coupled” sensitivity field is significantly different with higher peak values appearing on the top as well as on the bottom part of the pipe (fig. 5.6).

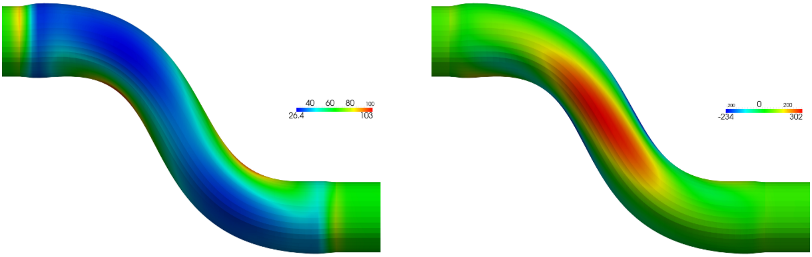
One more notable difference between the sensitivity maps is the sign of the values. The peak values on the coupled case can be both negative or positive ordering the bend the be updated outwards or inwards, respectively while in the other two cases the update would take place only on the outward direction. Interestingly, on the sides of the bend a sensitivity peak appears for the “fully-coupled” map which did not exist and could not be predicted from the calculation of the “rigid” or “semi-coupled” sensitivities as appears in figure 5.7.



(a) “rigid” sensitivity map

(b) “fully-coupled” sensitivity map

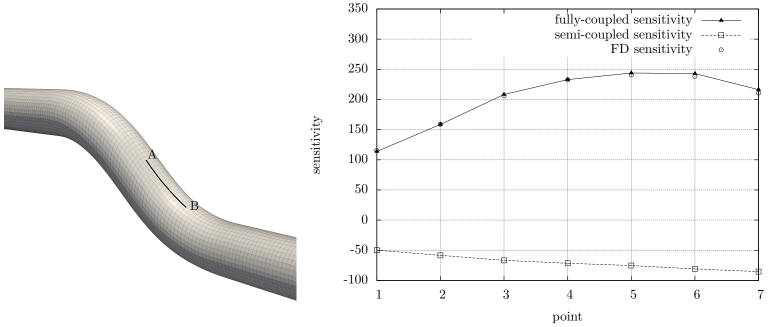
Figure 5.6: Bottom view (note the different color scale).



(a) “rigid” sensitivity map

(b) “fully-coupled” sensitivity map

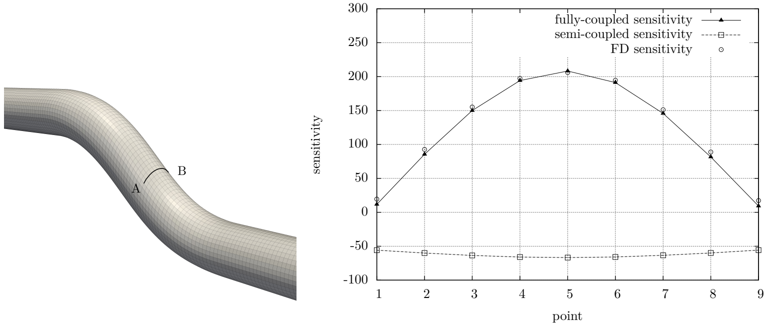
Figure 5.7: Side view (note the different color scale).



(a) line on which the sensitivities are measured

(b) sensitivity values

Figure 5.8: Comparison of the “semi-coupled” and “fully coupled ” sensitivities against finite differences along the bend.



(a) line on which the sensitivities are measured

(b) sensitivity values

Figure 5.9: Comparison of the “rigid”, “semi-coupled” and “fully coupled ” sensitivities against finite differences on a cross section of the bend.

A more qualitative result is the one of figures 5.8 and 5.9. Here, the three types of sensitivities are compared against finite differences on a line along the bend and on a cross section of the bend. The "fully-coupled" sensitivities match perfectly with the finite difference reference values with an average error bellow 2%. Moreover, the method predicts the correct sign for the sensitivities unlike the other two sensitivity types.

As already shown in Section 2.3.2, the presented method involves the following finite difference calculations:

(a) The coupling term of equation (2.52) representing the change of the fluid residuals by a change of the structural displacements.

(b) The derivative of the stiffness matrix with respect to the design variables, $\partial\mathbf{K}/\partial\mathbf{s}$ appearing in the final sensitivity equation (2.61).

(c) The derivative of the applied force on the structure with respect to the design variables $\partial\mathbf{f}/\partial\mathbf{s}$ appearing in the same equation.

For the evaluation of the term (a) the residuals of the fluid equation have to be computed for a disturbance of the design nodes of the interface in three Cartesian directions since the structural displacement field is defined in these three directions. In this step no additional fluid solution is required and therefore the computation time is limited to evaluation of the derivatives for the perturbed interface cells.

The last two terms, (b) and (c) are evaluated by the semi-analytic method presented in Section 2.2.1 which involves calculations of the element stiffness matrix and element force vector for a disturbance of the design node to the normal to the surface direction. As shown in [11] the method is efficient and it does not slow down the overall computation.

In general, for the aforementioned terms the proper finite difference step should be chosen which compromises between the truncation and cancellation error discussed in Section 2.1. For the specific problem it is shown in figure 5.10 that the range of the finite difference step for which the sensitivity values are not changing is relatively large. In this figure the deviation of the sensitivity value from a reference value at a point is shown while changing the finite difference step size for the calculation of the terms $\partial\mathbf{F}/\partial\mathbf{u}$ and $\partial\mathbf{K}/\partial\mathbf{s}$. During this numerical experiment only the step size was changing while all the other parameters remained the same. Both terms have proven to be very stable concerning the finite difference step size since the deviation of the sensitivity value was almost zero for a range of step values from $10e-3$ to $10e-13$. The calculations were performed on the coarsest mesh and a slight smaller range is expected for the finer meshes. This error analysis shows that the finite difference evaluation of the mentioned terms brings a negligible amount of error on the sensitivity calculation.

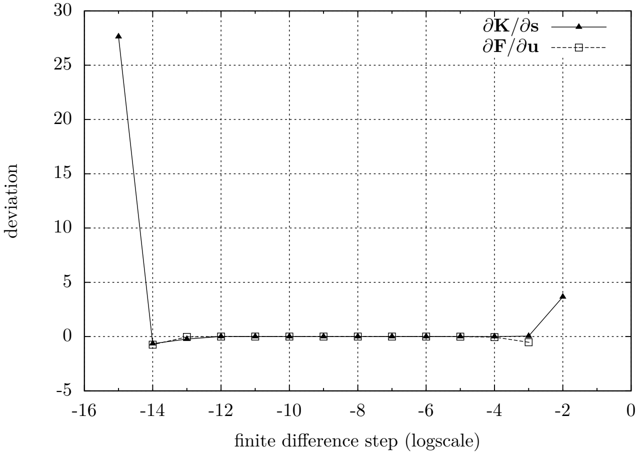
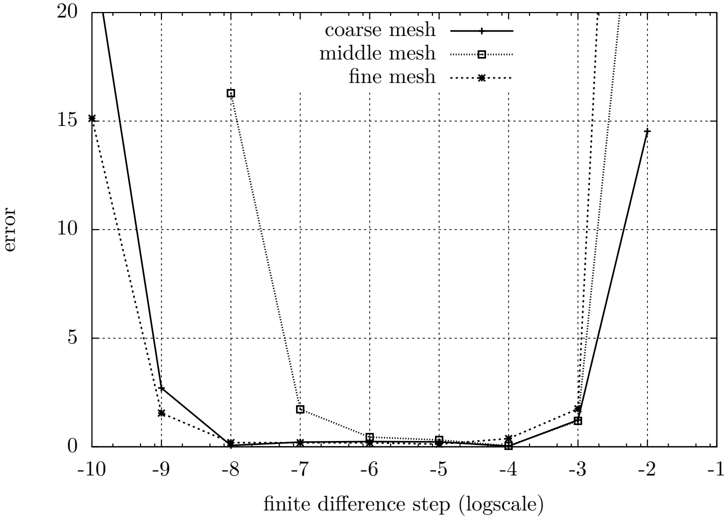


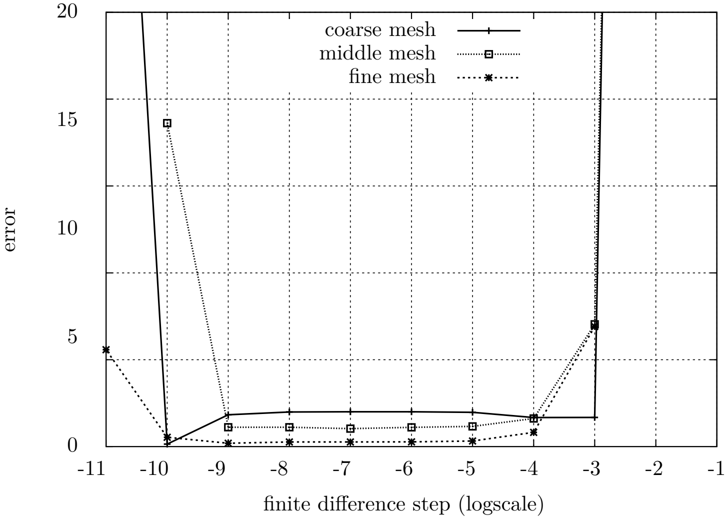
Figure 5.10: Deviation of the sensitivity values from the value of a finite difference step of $10e-8$ obtained by changing the step size for the terms $\partial\mathbf{F}/\partial\mathbf{u}$ and $\partial\mathbf{K}/\partial\mathbf{s}$.

In what follows the results obtained using the method are compared with finite differences sensitivity values by varying the finite difference step. For this, two sample points one on the flexible part of the pipe (bend) and one on the rigid part are chosen and their sensitivities obtained on three different meshes with 10.416, 35.280 and 147.840 fluid cells are compared against various finite difference step sizes. In figure 5.11 the final results are presented. Here, in each subfigure, the relative error against various finite difference step sizes for each point on three different meshes is shown.

The two points have the same behavior with respect to the order of the error for each mesh and at the finest mesh the error is less than 1%. Moreover, the range that the finite difference step gives an almost constant error is relatively large. As expected, the error is bigger on the coarse mesh and gets smaller as the mesh is refined. The errors of the coarse mesh at point (2) are relatively high. This is because the fluid and structural problem are calculated with different numerical schemes. The structural equations are solved by a node-based finite element discretization whereas the fluid equations are discretized by finite volume with pressures defined on the cells centers. Hence, the error comes from the interpolation from the faces to the nodes and vice versa within the coupling. This is more pronounced when the values from the two fields should be added to each other like in the final sensitivity equation (2.60). In this equation the second term is evaluated on the design faces while the third one on the design nodes. The addition of these terms requires an interpolation from the faces to the nodes which on coarse meshes gives a relative big error and this becomes obvious comparing the points (1) and (2) of figure 5.11. Point (2) is on the bend where both terms are non-zero and point (1) is on the straight pipe where the third term of equation (2.60) is zero. The error of point (1) is noticeably smaller on the coarse mesh compared to point (2) and this difference is getting smaller as the mesh is refined.



(a) Finite difference study for point (1)



(b) Finite difference study for point (2)

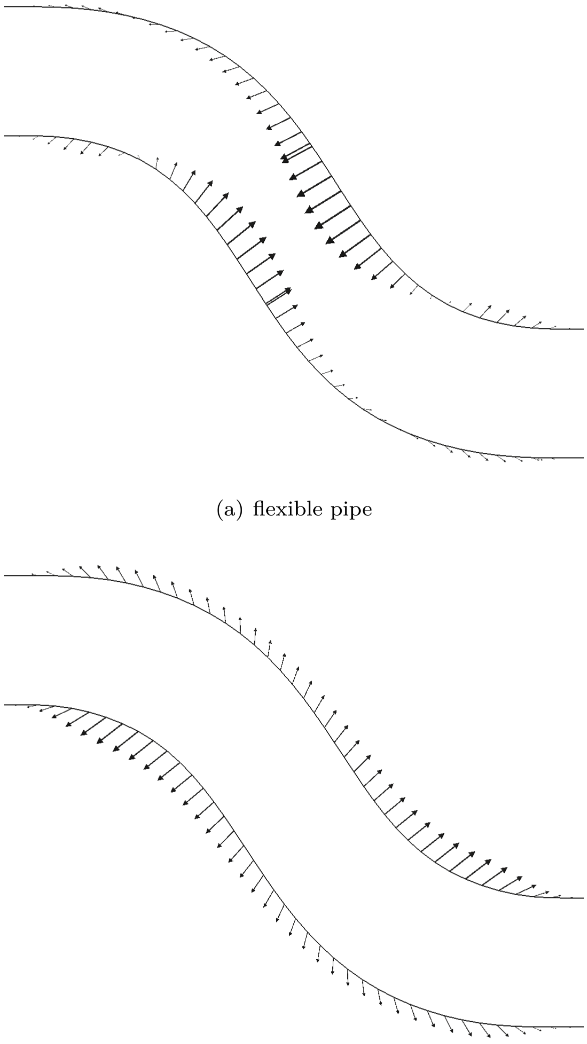
Figure 5.11: Finite difference calculations for two sample points on the pipe for three difference meshes. Point (1) is on the rigid part of the pipe while point (2) is on the flexible bend of the pipe.

5.2 Node-based shape optimiztion on a flexible pipe

In this section a node-based shape optimization of the flexible pipe of figure 5.1 is performed with 35.280 fluid cells and 984 surface points on the elastic bend. The design surface is the flexible bend of the pipe and the objective is the power loss shown in equation (2.21). The required sensitivity analysis is validated in the previous section. For this case, the Vertex-Morphing and the In-plane regularization methods presented in Chapter 3 are used to maintain the smoothness of the shape and mesh. The computer programs used for the optimization are explained in detail in the previous chapter. First, the computational loop is presented and in the sequence the improved shape is discussed and compared with the one obtained by a CFD shape optimization of the pipe considering the bend to be rigid.

For the optimization, first the coupled sensitivity analysis is performed based on algorithm (1) of Chapter 2, through the following steps: After the convergence of the fluid-structure interaction iteration loop, the adjoint coupled analysis is performed until its convergence. As the result, the adjoint velocity and pressure from the fluid side and the adjoint displacement on the structure side are obtained. Having the adjoint fields available, the coupled sensitivities are computed based on equation (2.60). As it is observed before, both fluid and structure contribute to this evaluation since the design surface is also flexible. The optimizer uses the derived sensitivity field and calculates the shape update vector based on the Vertex-Morphing technique. In the sequence, the In-plane regularization method is applied to smooth the surface mesh. The initial shape of each surface element is used as template and a unit stress tensor is defined to regulate the size of each element uniformly. Having the surface update, the mesh motion algorithm, which in this case is the one suggested by [65] and presented in Section 3.4.3, updates the volume fluid mesh. At the end, the primal and adjoint states are computed on the updated discretization. The loop is continued until a desired improvement is reached.

For the specific problem the final (optimal) shape is trivial since the optimizer tries to displace the boundaries of the bend far from the flow so the power loss decreases. However, the intermediate stages of the optimization procedure are interesting as they show great differences



(a) flexible pipe

(b) rigid pipe

Figure 5.12: Shape update on a longitudinal section during CFD and FSI shape optimization.

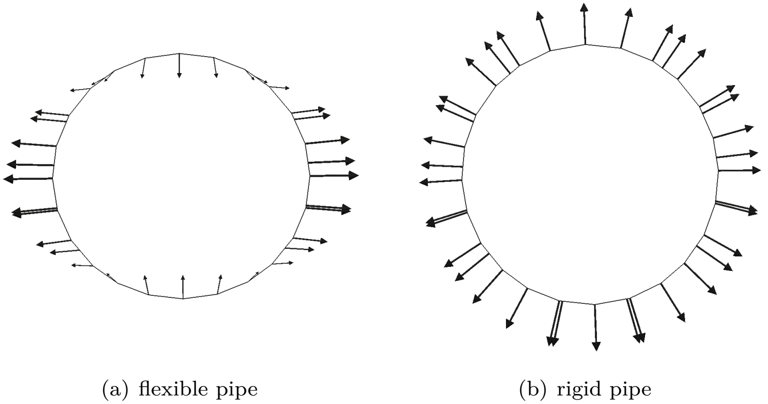

(a) flexible pipe (b) rigid pipe

Figure 5.13: Shape update on a cross section during CFD and FSI shape optimization.

from an optimization with the assumption of a rigid bend. For instance figure 5.12 compares the update pattern in an early stage of the optimization for the flexible as well as the rigid pipe. As expected from the sensitivity map of figure 5.5, the update on the flexible pipe directs the shape inwards in some areas while in the rigid pipe the update is always outwards. This does not mean that the cross section of the pipe is getting smaller since the update on the sides is outwards with a larger magnitude compared to the inward vectors as shown in figure 5.13.

The different update pattern of the coupled sensitivity optimization causes the shape to improve in a slightly different direction which brings a 9% of improvement compared to the 5% improvement achieved by the decoupled sensitivity assumption. Figure 5.14 compares the updated shape after 5 optimization steps with the one of a CFD (rigid wall) optimization with the same step size and the same filter radius. The shape resulted from the fully-coupled sensitivity is depicted with red color while the one obtained by regarding the bend as rigid is shown with blue color. It is clear that two different improvement paths are followed. The mechanical interpretation of the inward sensitivities at the "neck“ of the bend can be that stiffening of this part of the flexible

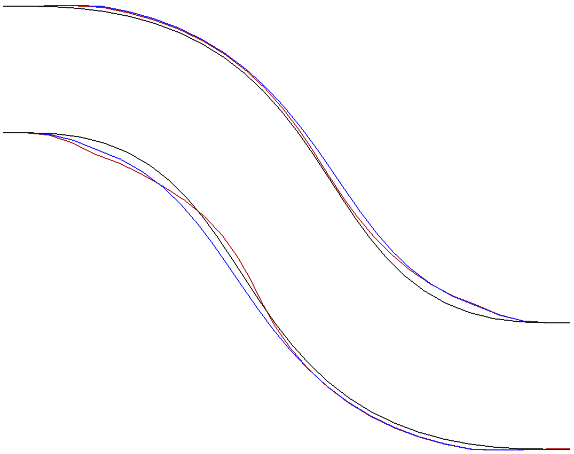


Figure 5.14: Improved shape comparison based on CFD (blue line) and FSI (red line) sensitivitivies. Black line: initial design

bend would decrease the structural deformation at that position and consequently the bend angle becomes smaller which is favorable for pressure loss.

Chapter 6

Conclusions and outlook

6.1 Concluding remarks

This thesis deals with gradient-based steady-state shape optimization of fluid and coupled problems using a node-based parametrization. The focus is on the following aspects of the optimization procedure:

- the adjoint-based sensitivity analysis with emphasis to coupled problems.
- the regularization of the design surface as well as the discretization.

The sensitivity analysis is performed in the adjoint manner because of the vast number of design variables. For the structure, the discrete semi-analytic method is applied which involves an element-level finite difference step allowing for a robust solution of the structural self-adjoint problem. On the other side, for the adjoint sensitivity analysis of the fluid problem the continuous method is used. These different adjoint formulations have to be coupled at the interface of the fluid and structure domains for the adjoint sensitivity analysis of the coupled fluid-structure interaction problem. Here, this coupling is done in a partitioned way, which is similar to the solution of the primal problem. This is achieved by decomposing the adjoint monolithic system into a structure and a fluid problem. The effect of coupling matrices is added to the respective equations as sources. These "fixed-point" iterations are performed until a desired level of residuum is achieved. In this work, emphasis is given to the construction of the different components of the monolithic system and their role during the solution of the partitioned problem. Moreover, the inclusion of the discrete coupling terms to the continuous fluid adjoint equations is also described.

Updating the design in each optimization step is performed by the Vertex-Morphing method. This method incorporates the smoothing operator into the definition of the design variables. In this way, large design updates are achieved while retaining a smooth design surface.

The maintenance of the quality of the mesh during shape evolution is also considered. For the treatment of the design surface mesh, a novel In-plane regularization method is developed. This method regularizes the surface mesh by assuming an artificial stress field on the surface and a global structural equilibrium is solved. The applied stress adapts the shape of each element towards a desired predefined template geometry and at the end a globally smooth mesh is achieved.

Finally, the various methods are incorporated in a robust and efficient optimization workflow and the performance of the method is proven by successfully optimizing the shape of industrial cases.

It can be concluded that node-based shape optimization enhanced with the presented regularization tools is a powerful design tool which facilitates the design process and offers wide range design solutions. Furthermore, taking into consideration the elasticity of the structure, even for small structural deformations, has a dramatic influence on the sensitivity field and thus should not be neglected.

6.2 Outlook

Concerning the sensitivity analysis of a coupled problem, the presented method can be extended to transient problems. Then, the memory and computation time requirements will be an obstacle and therefore efficient and robust algorithms which deal with this issue are required.

In general, a field of application for which such a method is essential is the shape optimization of wind turbine blades. These structures undergo large deformations due to wind loads and at the same time their shape has a great impact in their efficiency. Even though there has been many publications on the FSI numerical modeling of this problem [45, 58, 67], an adjoint-based shape optimization of the blade geometry is still missing. The wind blade has a complex geometry and thus the abilities of the Vertex-Morphing and In-plane regularization methods can be well used.

Moreover, the wind turbine problem by its nature is transient, but because of the rotational pattern of the motion, it can be modeled as a quasi-steady fluid-structure interaction system defined on a rotational frame. At this end, additional momentum terms should be added to the state equations of both the structure and fluid. For this primal system, the adjoint coupled sensitivity analysis can be performed in a steady manner and will result in additional terms in the adjoint coupled equations. A basis for this research can be the work of Nielsen et al. [94, 95] and Economon et al. [26] on the adjoint CFD sensitivity analysis of rotating geometries. As a starting geometry the NREL Phase VI rotor [110] can be used. The NREL Phase VI Unsteady Aerodynamic Experiment can provide an excellent validation test case for the primal state.

Appendix A

Gauss and Divergence theorem

Let Ω be a closed volume bounded by a smooth surface $\Gamma = \partial\Omega$ and consider the unit normal to the surface pointing outwards as shown in the figure A.1. Consider also $f(\mathbf{x})$ being a scalar, vector or tensor field

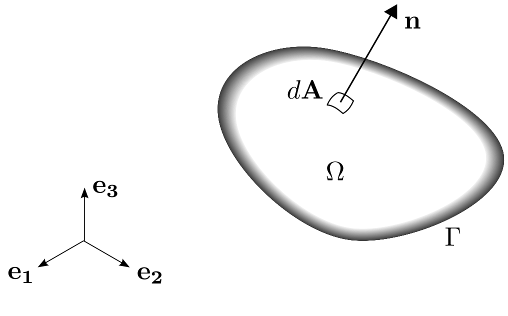


Figure A.1: Volume Ω and elementary area $d\mathbf{\Gamma}$.

of any order. The following relationship holds

$$\int_{\Omega} \nabla \cdot f d\Omega = \int_{\Gamma} f \cdot \mathbf{n} dA \tag{A.1}$$

and is called the Gauss' theorem. The Gauss' theorem relates a volume integral to an integral over each bounded surface.

The most common form of the Gauss' theorem is in vector form where f is a vector field denoted by $\mathbf{F}$ and is usually referred to as divergence theorem

$$\int_{\Omega} \nabla \cdot \mathbf{F} d\Omega = \int_{\Gamma} \mathbf{F} \cdot \mathbf{n} dA, \tag{A.2}$$

and physically, it states that the source rate in Ω is equal to the integral of the fluxes along the closed boundary Γ. The proof of these theorems can be found in any text book of multi-variable calculus and an presentation closely related to fluids can be found in [71].

Appendix B

The pressure-correction method

B.1 The SIMPLE method for the Navier-Stokes

The Navier-Stokes equations for incompressible flows (2.18) in a three-dimensional case are composed from the three momentum equations for each velocity component and one continuity equation constraining the velocity field. The gradient of the pressure field contributes to each component of the momentum equation. The way that these equations are defined implies that the momentum equation should be used to determine the velocity components. As a result, the pressure should be evaluated from the continuity equation. However, the continuity equation itself does not contain any pressure term and for this reason it is combined with the momentum equation to produce an equation for the pressure. This elliptic equation is the Poisson equation for the pressure and together with the momentum equation determines the unknown velocity and pressure fields.

The Poisson equation is derived by taking the divergence of the momentum equation (2.18)

$$\frac{\partial}{\partial x_i}\left(v_j\frac{\partial v_i}{\partial x_j}+\frac{\partial p}{\partial x_j}-\frac{\partial}{\partial x_j}(2\nu\varepsilon_{ij})+b_i\right)=0. \tag{B.1}$$

Here the momentum equation is presented in Cartesian coordinates and b_i is the source term. Assuming a constant viscosity and density field and applying the continuity in the above equation results to the following Poisson equation for the pressure

$$\frac{\partial}{\partial x_i}\left(\frac{\partial p}{\partial x_i}\right)=-\frac{\partial}{\partial x_i}\left(v_j\frac{\partial v_i}{\partial x_j}\right). \tag{B.2}$$

In the following, the iterative SIMPLE method [14, 99] used to solve the momentum and Poisson equation is presented and the notation used is the one of Ferziger and Peric [38].

In the SIMPLE method the semi-discretized momentum and Poison equations are solved iteratively. The starting point of iteration m is a divergence-free velocity and pressure field (v_i^{m-1}, p^{m-1}). Then the discrete momentum equation for the velocity is solved

$$A_p v_{i,P}^{m*}+\sum_l A_l v_{i,l}^{m*}=-\Delta\Omega\left(\frac{\partial p^{m-1}}{\partial x_i}\right)_P+S\Delta\Omega. \tag{B.3}$$

The index P refers is the current index of the velocity while l refers to its neighboring nodes for each discretized equation. S is the source term containing all the constant terms and the ones treated explicitly with respect to the velocity. The solution v_i^{m*} of the above nonlinear equation does not satisfy continuity and a “projection” or “correction” of this field such that it results to a divergence free-field is required.

In these class of methods the velocity field is corrected by a correction v_i'

$$v_i^m=v_i^{m*}+v_i', \tag{B.4}$$

which results to the divergence-free velocity v_i^m. The pressure is corrected from the previous iteration $m-1$ as follows

$$p^m=p^{m-1}+p'. \tag{B.5}$$

Inserting the above corrections into (B.3) yields to a momentum equation for the correction v_i'

$$A_p v'_{i,P} + \sum_l A_l v'_{i,l} = -\Delta\Omega \left(\frac{\partial p'}{\partial x_i} \right)_P \tag{B.6}$$

since (B.3) is satisfied also from the pair (v_i^m, p_i^m).

From this equation the correction for the velocity is obtained

$$v'_{i,P} = -\frac{\Delta\Omega}{A_p} \left(\frac{\partial p'}{\partial x_i} \right)_P , \tag{B.7}$$

by neglecting the second term of the left hand side of (B.3).

Furthermore, the term v_i^m should satisfy continuity. So, taking the divergence of equation (B.4) and substituting to it equation (B.7), the semi-discrete pressure correction equation is obtained

$$\frac{\partial}{\partial x_i} \left(\frac{\Delta\Omega}{A_p} \left(\frac{\partial p'}{\partial x_i} \right)_P \right) = \left(\frac{\partial v_i^{m*}}{\partial x_i} \right)_P . \tag{B.8}$$

After the solution of the pressure-correction equation which results in the update of p', the pressure and velocity fields are corrected to obtain v_i^m and p^m. This procedure is repeated till the corrections p' and v_i' are negligibly small.

Omitting the second term of the left hand side of (B.3) makes the SIMPLE method to converge very slow. The convergence can be significantly improved by underrelaxing the correction of velocities and pressures, as for instance

$$p^m = p^{m-1} + a_p p', \tag{B.9}$$

with $0 \leq a_p \leq 1$.

B.2 The SIMPLE method for the adjoint Navier-Stokes

The SIMPLE method is also applied for the solution of the adjoint Navier-Stokes problem and the procedure which is followed is similar.

The adjoint Poisson equation is obtained by taking the divergence of the adjoint momentum equation (2.31). This equation in Cartesian coordinates is the following

$$\frac{\partial}{\partial x_i}\left(\frac{\partial \psi_p}{\partial x_i}\right) = -\frac{\partial}{\partial x_i}\left(\frac{\partial \psi_{vj}}{\partial x_i} v_j\right) - \frac{\partial}{\partial x_i}\left(v_j \frac{\partial \psi_{vi}}{\partial x_j}\right). \tag{B.10}$$

Compared to respective equation (B.2) of the primal problem one additional term appears: the first term of the right hand side. This term is treated explicitly and by that only small modifications compared to the primal solution are required.

Following the same sequence as in section (B.1), the starting point is the divergence-free adjoint velocity and the adjoint pressure $({\psi_v}_i^{m-1}, \psi_p^{m-1})$ of the previous iteration $m-1$. In the next iteration m, the semi-discretized momentum equation for the adjoint velocity is

$$\begin{gathered} \alpha_P {\psi_v}_{i,P}^{m*} + \sum_l \alpha_l {\psi_v}_{i,l}^{m*} = \\ -\Delta\Omega\left(\frac{\partial {\psi_p}^{m-1}}{\partial x_i}\right)_P - \Delta\Omega\left(v_j \frac{\partial {\psi_v}_j^{m-1}}{\partial x_i}\right)_P + s\Delta\Omega \end{gathered} \tag{B.11}$$

The problem is solved also with a projection manner by correcting the adjoint velocities ${\psi_v}_i^{m*}$ such that they give a divergence-free adjoint velocity field ${\psi_v}_i^{m}$

$${\psi_v}_i^{m} = {\psi_v}_i^{m*} + {\psi_v}_i'. \tag{B.12}$$

The pressure field is again corrected from the previous iteration $m-1$

$$\psi_p^m = \psi_p^{m-1} + \psi_p' \tag{B.13}$$

As already mentioned before the second term of equation (B.11) is treated explicitly and thus it can be included on the last source term s of the same equation. Hence, the adjoint velocity correction is obtained as in the primal solution by

$${\psi_u}_{i,P}' = -\frac{\Delta\Omega}{A_p}\left(\frac{\partial {\psi_p}'}{\partial x_i}\right)_P \tag{B.14}$$

by neglecting the second term of the left hand side of (B.11), exactly as in the primal solution.

Following the same ideas, the term ${\psi_{v}}_i^m$ should satisfy continuity. Hence, by taking the divergence of equation (B.12) and substituting v_i' from equation (B.14), the adjoint pressure-correction equation is obtained

$$\frac{\partial}{\partial x_i}\left(\frac{\Delta\Omega}{A_p}\left(\frac{\partial {\psi_p}'}{\partial x_i}\right)_P\right)=\left(\frac{\partial {\psi_u}_i^{m*}}{\partial x_i}\right)_P . \tag{B.15}$$

After the solution of the above equation which gives the adjoint correction for the pressure ${\psi_p}'$, the adjoint velocity and adjoint pressure fields are corrected to obtain ${\psi_v}_i^m$ and ψ_p^m and the procedure is repeated till convergence.

Bibliography

[1] M. Abu-Zurayk and J. Brezillon. Development of the adjoint approach for aeroelastic wing optimization. In A. Dillmann, G. Heller, H.-P. Kreplin, W. Nitsche, and I. Peltzer, editors, *New Results in Numerical and Experimental Fluid Mechanics VIII*, volume 121 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 59–66. Springer Berlin Heidelberg, 2013.

[2] J. Arora. *Introduction to optimum design.* Academic Press, 2004.

[3] J. S. Arora and Q. Wang. Review of formulations for structural and mechanical system optimization. *Structural and Multidisciplinary Optimization*, 30(4):251–272, 2005.

[4] D. Ashlock. *Evolutionary computation for modeling and optimization*, volume 103. Springer, 2006.

[5] R. Balasubramanian and J. C. Newman. Discrete direct and adjoint sensitivity analysis for arbitrary Mach number flows. *International Journal for Numerical Methods in Engineering*, 66 (2):297–318, 2006.

[6] B. Barthelemy and R. T. Haftka. Accuracy analysis of the semi-analytical method for shape sensitivity calculation. *Mechanics of Structures and Machines*, 18(3):407–432, 1990.

[7] M. Behr and F. Abraham. Free-surface flow simulations in the presence of inclined walls. *Computer Methods in Applied Mechanics and Engineering*, 191(47):5467–5483, 2002.

[8] K.-U. Bletzinger. *Form finding and optimization of membranes and minimal surfaces.* Institut für Baustatik, 1998.

[9] K.-U. Bletzinger. A consistent frame for sensitivity filtering and the vertex assigned morphing of optimal shape. *Structural and Multidisciplinary Optimization*, 49(6):873–895, 2014.

[10] K.-U. Bletzinger and E. Ramm. A general finite element approach to the form finding of tensile structures by the updated reference strategy. *International Journal of Space Structures*, 14(2):131–145, 1999.

[11] K.-U. Bletzinger, M. Firl, and F. Daoud. Approximation of derivatives in semi-analytical structural optimization. *Computers & Structures*, 86(13):1404–1416, 2008.

[12] J. Bonet and R. D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, 1997.

[13] V. Braibant and C. Fleury. Shape optimal design using B-splines. *Computer Methods in Applied Mechanics and Engineering*, 44(3): 247–267, 1984.

[14] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding. Two calculation procedures for steady, three-dimensional flows with recirculation. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, pages 60–68. Springer, 1973.

[15] J. E. Castillo, S. Steinberg, and P. J. Roache. On the folding of numerically generated grids: use of a reference grid. *Communications in Applied Numerical Methods*, 4(4):471–481, 1988.

[16] G. Cheng and N. Olhoff. New method of error analysis and detection in semi-analytical sensitivity analysis. In G.I.N. Rozvany, editor, *Optimization of Large Structural Systems*, volume 231 of *NATO ASI Series*, pages 361–383. Springer Netherlands, 1993.

[17] K. K. Choi and N.-H. Kim. *Structural sensitivity analysis and optimization 1: linear systems*, volume 1. Springer, 2006.

[18] A. De Boer, M. S. Van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers & Structures*, 85(11-14):784–795, 2007.

[19] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structures*, 80(3):305–316, 2002.

[20] J. Degroote. *Development of algorithms for the partitioned simulation of strongly coupled fluid-structure interaction problems*. PhD thesis, Ghent University, 2010.

[21] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels. Stability of a coupling technique for partitioned solvers in FSI applications. *Computers & Structures*, 86(23):2224–2234, 2008.

[22] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. *Computers & Structures*, 87(11):793–801, 2009.

[23] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid–structure interaction. *Computers & Structures*, 88(7):446–457, 2010.

[24] J. Degroote, M. Hojjat, E. Stavropoulou, R. Wüchner, and K.-U. Bletzinger. Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem. *Structural and Multidisciplinary Optimization*, 47(1):77–94, 2013.

[25] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. Arbitrary Lagrangian–Eulerian methods. *Encyclopedia of Computational Mechanics*, 2004.

[26] T. D. Economon, F. Palacios, and J. J. Alonso. A viscous continuous adjoint approach for the design of rotating engineering applications. *AIAA Paper*, 2580:24–27, 2013.

[27] L. F. R. Espath, R. V. Linn, and A. M. Awruch. Shape optimization of shell structures based on NURBS description using automatic differentiation. *International Journal for Numerical Methods in Engineering*, 88(7):613–636, 2011.

[28] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000.

[29] C. Farhat and P. Geuzaine. Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. *Computer Methods in Applied Mechanics and Engineering*, 193(39):4073–4095, 2004.

[30] C. Farhat, M. Lesoinne, and N. Maman. Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution. *International Journal for Numerical Methods in Fluids*, 21(10): 807–835, 1995.

[31] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163 (1):231–245, 1998.

[32] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17):1973–2001, 2006.

[33] A. Fazzolari, N. R. Gauger, and J. Brezillon. Efficient aerodynamic shape optimization in MDO context. *Journal of Computational and Applied Mathematics*, 203(2):548–560, 2007.

[34] C. A. Felippa. A study of optimal membrane triangles with drilling freedoms. *Computer Methods in Applied Mechanics and Engineering*, 192(16):2125–2168, 2003.

[35] C. A. Felippa. Introduction to finite element methods. University Lectures, University of Colorado, Boulder, http://www. colorado. edu/engineering/CAS/courses. d/IFEM. d, 2004.

[36] C. A. Felippa and B. Haugen. A unified formulation of small-strain corotational finite elements: I. Theory. *Computer Methods in Applied Mechanics and Engineering*, 194(21):2285–2335, 2005.

[37] C. A. Felippa, K. C. Park, and C. Farhat. Partitioned analysis of coupled mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3247–3270, 2001.

[38] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 2002.

[39] M. Firl and K.-U. Bletzinger. Shape optimization of thin walled structures governed by geometrically nonlinear mechanics. *Computer Methods in Applied Mechanics and Engineering*, 237:107–117, 2012.

[40] M. Firl, R. Wüchner, and K.-U. Bletzinger. Regularization of shape optimization problems using fe-based parametrization. *Structural and Multidisciplinary Optimization*, 47(4):507–521, 2013.

[41] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and Simulation of the Circulatory System*. MS & A. Springer, 2009.

[42] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.

[43] T. G. Gallinger. *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*. PhD thesis, Technische Universität München, München, 2010.

[44] M. W. Gee, U. Küttler, and W. A. Wall. Truly monolithic algebraic multigrid for fluid–structure interaction. *International Journal for Numerical Methods in Engineering*, 85(8):987–1016, 2011.

[45] P. Giguere and M. S. Selig. Design of a tapered and twisted blade for the NREL combined experiment rotor. *NREL/SR*, 500:26173, 1999.

[46] 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.

[47] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3):283–303, 1994.

[48] D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine Learning*, 3(2):95–99, 1988.

[49] C. Grandmont. Existence for a three-dimensional steady state fluid-structure interaction problem. *Journal of Mathematical Fluid Mechanics*, 4(1):76–94, 2002.

[50] R. T. Haftka and Z. Gürdal. *Elements of structural optimization*, volume 11. Springer, 1992.

[51] G. Hansen, A. Zardecki, D. Greening, and R. Bos. A finite element method for unstructured grid smoothing. *Journal of Computational Physics*, 194(2):611–631, 2004.

[52] G. A. Hansen, R. W. Douglass, and A. Zardecki. *Mesh enhancement: selected elliptic methods, foundations and applications.* Imperial College Press, 2005.

[53] M. Heil. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1):1–23, 2004.

[54] M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. National Bureau of Standards Washington, DC, 1952.

[55] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 135(2):203–216, 1997.

[56] M. Hojjat, E. Stavropoulou, and K.-U. Bletzinger. The Vertex Morphing method for node-based shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 268:494 – 513, 2014.

[57] J. Hron and S. Turek. A monolithic FEM/Multigrid solver for an ALE formulation of fluid-structure interaction with applications in biomechanics. In H.-J. Bungartz and M. Schäfer, editors,

Fluid-Structure Interaction, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 146–170. Springer Berlin Heidelberg, 2006.

[58] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. High-performance computing of wind turbine aerodynamics using isogeometric analysis. *Computers & Fluids*, 49(1):93–100, 2011.

[59] J. Huan and V. Modi. Optimum design of minimum drag bodies in incompressible laminar flow using a control theory approach. *Inverse Problems in Engineering*, 1(1):1–25, 1994.

[60] T. J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Dover Civil and Mechanical Engineering Series. Dover Publications, 2000.

[61] B. M. Irons and R. C. Tuck. A version of the Aitken accelerator for computer iteration. *International Journal for Numerical Methods in Engineering*, 1(3):275–277, 1969.

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

[63] A. Jameson. Aerodynamic shape optimization using the adjoint method. Lectures at the Von Karman Institute, Brussels, 2003.

[64] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1):213–237, 1998.

[65] H. Jasak and Z. Tukovic. Automatic mesh motion for the unstructured finite volume method. *Transactions of FAMENA*, 30 (2):1–20, 2006.

[66] X. Jiao. Face offsetting: A unified approach for explicit moving interfaces. *Journal of Computational Physics*, 220(2):612–625, 2007.

[67] J. Johansen, N. N. Sorensen, J. A. Michelsen, and S. Schreck. Detached-eddy simulation of flow around the NREL phase-VI blade. In *ASME 2002 Wind Energy Symposium*, pages 106–114. American Society of Mechanical Engineers, 2002.

[68] J. M. Kiendl. *Isogeometric Analysis and Shape Optimal Design of Shell Structures*. PhD thesis, Technische Universität München, Munich, Germany, 2011.

[69] P. M. Knupp. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers*, 15(3):263–268, 1999.

[70] P. M. Knupp and S. Steinberg. *The fundamentals of grid generation*. CRC Press, 1993.

[71] P. K. Kundu and I. M. Cohen. *Fluid Mechanics*. Elsevier Science, 2010.

[72] A. Kupzok. *Modeling the interaction of wind and membrane structures by numerical simulation*. PhD thesis, Technische Universität München, München, 2009.

[73] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43 (1):61–72, 2008.

[74] C. Le, T. Bruns, and D. Tortorelli. A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9):985–996, 2011.

[75] J. Linhard and K.-U. Bletzinger. "Tracing" the equilibrium - Recent advances in numerical form finding. *International Journal of Space Structures*, 25(2):107–116, 2010.

[76] R. Löhner, J. Cebral, C. Yang, J. D. Baum, E. Mestreau, C. Charman, and D. Pelessone. Large-scale fluid-structure interaction simulations. *Computing in Science & Engineering*, 6(3):27–37, 2004.

[77] E. Lund and N. Olhoff. Shape design sensitivity analysis of eigenvalues using "exact" numerical differentiation of finite element matrices. *Structural Optimization*, 8(1):52–59, 1994.

[78] E. Lund, H. Møller, and L. A. Jakobsen. Shape design optimization of stationary fluid-structure interaction problems with large displacements and turbulence. *Structural and Multidisciplinary Optimization*, 25(5-6):383–392, 2003.

[79] M. Marius and G. L. David. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[80] M. Marius and G. L. David. Fast matching of binary features. In *Computer and Robot Vision (CRV)*, pages 404–410, 2012.

[81] M. Marius and G. L. David. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36, 2014.

[82] J. Martins, J. Alonso, and J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004.

[83] J. Martins, J. Alonso, and J. Reuther. A coupled-adjoint sensitivity analysis method for high-fidelity aero-structural design. *Optimization and Engineering*, 6(1):33–62, 2005.

[84] K. Maute, M. Nikbay, and C. Farhat. Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. *AIAA journal*, 39(11):2051–2061, 2001.

[85] K. Maute, M. Nikbay, and C. Farhat. Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. *International Journal for Numerical Methods in Engineering*, 56(6):911–933, 2003.

[86] C. Militello and C. A. Felippa. The first ANDES elements: 9-dof plate bending triangles. *Computer Methods in Applied Mechanics and Engineering*, 93(2):217–246, 1991.

[87] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[88] B. Mohammadi. Shape optimization for 3D turbulent flows using automatic differentiation. *International Journal of Computational Fluid Dynamics*, 11(1-2):27–50, 1998.

[89] B. Mohammadi and O. Pironneau. *Applied shape optimization for fluids*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2009.

[90] D. P. Mok, W. A. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. *Computational Fluid and Solid Mechanics*, pages 1325–1328, 2001.

[91] S. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA paper*, 667, 2000.

[92] A. Nemili, E. Özkaya, and N. Gauger. Discrete adjoint based sensitivity analysis for optimal active flow control of high-lift configurations. *PAMM*, 13(1):347–348, 2013.

[93] E. J. Nielsen, J. Lu, M. A. Park, and D. L. Darmofal. An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids. *Computers & Fluids*, 33(9):1131–1155, 2004.

[94] E. J. Nielsen, B. Diskin, and N. K. Yamaleev. Discrete adjoint-based design optimization of unsteady turbulent flows on dynamic unstructured grids. *AIAA Journal*, 48(6):1195–1206, 2010.

[95] E. J. Nielsen, E. M. Lee-Rausch, and W. T. Jones. Adjoint-based design of rotors in a noninertial reference frame. *Journal of Aircraft*, 47(2):638–646, 2010.

[96] C. Othmer. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *International Journal for Numerical Methods in Fluids*, 58(8): 861–877, 2008.

[97] C. Othmer, T. Kaminski, and R. Giering. Computation of topological sensitivities in fluid dynamics, cost function versatility. In *Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CFD*, Egmond aan Zee, Netherlands, 2006, 2006.

[98] C. Othmer, E. de Villiers, and H. G. Weller. Implementation of a continuous adjoint for topology optimization of ducted flows. In *18th AIAA Computational Fluid Dynamics Conference*, 2007.

[99] S. Patankar. *Numerical heat transfer and fluid flow.* CRC Press, 1980.

[100] C. S. Peskin. The fluid dynamics of heart valves: experimental, theoretical, and computational methods. *Annual Review of Fluid Mechanics*, 14(1):235–259, 1982.

[101] S. Piperno and C. Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems, Part II: Energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24):3147–3170, 2001.

[102] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, 124 (1):79–112, 1995.

[103] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(01):97–110, 1974.

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

[105] M. Scherer, R. Denzer, and P. Steinmann. A fictitious energy approach for shape optimization. *International Journal for Numerical Methods in Engineering*, 82(3):269–302, 2010.

[106] S. Schmidt. *Efficient large scale aerodynamic design based on shape calculus.* PhD thesis, Universität Trier, Trier, 2010.

[107] S. Schmidt, C. Ilic, N. Gauger, and V. Schulz. Shape gradients and their smoothness for practical aerodynamic design optimization. *Optimization and Engineering, submitted (Preprint No. SPP1253-10-03, 2008)*, 2008.

[108] S. Shankaran and A. Jameson. Adjoint formulations for topology, shape and discrete optimization. In *45th AIAA Aerospace Science Meeting and Exhibit, Reno, NV*, 2007.

[109] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, and K.-U. Bletzinger. Interface Jacobian-based Co-Simulation. *International Journal for Numerical Methods in Engineering*, 98(6):418–444, 2014.

[110] D. A. Simms, S. Schreck, M. Hand, and L. J. Fingersh. *NREL unsteady aerodynamics experiment in the NASA-Ames wind tunnel: a comparison of predictions to measurements.* National Renewable Energy Laboratory Colorado, USA, 2001.

[111] J. Sobieszczanski-Sobieski. Sensitivity of complex, internally coupled systems. *AIAA journal*, 28(1):153–160, 1990.

[112] O. Soto and R. Löhner. CFD shape optimization using an incomplete-gradient adjoint formulation. *International Journal for Numerical Methods in Engineering*, 51(6):735–753, 2001.

[113] O. Soto and R. Löhner. On the computation of flow sensitivities from boundary integrals. *AIAA Paper*, 112, 2004.

[114] E. Stavropoulou, M. Hojjat, and K.-U. Bletzinger. In-plane mesh regularization for node-based shape optimization problems. *Computer Methods in Applied Mechanics and Engineering*, 275: 39 – 54, 2014.

[115] A. Stück. *Adjoint Navier–Stokes methods for hydrodynamic shape optimisation.* PhD thesis, Technischen Universität Hamburg-Harburg, Hamburg, Germany, 2011.

[116] A. Stück and T. Rung. Adjoint RANS with filtered shape derivatives for hydrodynamic optimisation. *Computers & Fluids*, 47 (1):22–32, 2011.

[117] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space-time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195(17):2002–2027, 2006.

[118] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5743–5753, 2006.

[119] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of grid generation.* CRC Press, 1999.

[120] S. Timoshenko. *Theory of elasticity.* Engineering societies monographs. McGraw-Hill, 1951.

[121] D. A. Tortorelli and P. Michaleris. Design sensitivity analysis: overview and review. *Inverse problems in Engineering*, 1(1): 71–105, 1994.

[122] F. Van Keulen, R.T. Haftka, and N.H. Kim. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194 (30):3213–3243, 2005.

[123] R. Van Loon, P. D. Anderson, F. N. Van de Vosse, and S. J. Sherwin. Comparison of various fluid–structure interaction methods for deformable bodies. *Computers & Structures*, 85(11):833–843, 2007.

[124] A. H. van Zuijlen, S. Bosscher, and H. Bijl. Two level algorithms for partitioned fluid–structure interaction computations. *Computer Methods in Applied Mechanics and Engineering*, 196(8): 1458–1470, 2007.

[125] G. N. Vanderplaats. *Numerical optimization techniques for engineering design: with applications*, volume 1. McGraw-Hill New York, 1984.

[126] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method.* Pearson Education, 2007.

[127] W. A. Wall1, A. Gerstenberger, P. Gamnitzer, C. Förster, and E. Ramm. Large deformation fluid-structure interaction - advances in ALE methods and new fixed grid approaches. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 195–232. Springer Berlin Heidelberg, 2006.

[128] M. Widhalm, J. Brezillon, C. Ilic, and T. Leicht. Investigation on adjoint based gradient computations for realistic 3d aero-optimization. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, Texas*, 2010.

[129] A. M. Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics*, 1(2):149–172, 1966.

[130] R. Wüchner and K.-U. Bletzinger. Stress-adapted numerical form finding of pre-stressed surfaces by the updated reference strategy. *International Journal For Numerical Methods in Engineering*, 64 (2):143–166, 2005.

[131] R. Wüchner, A. Kupzok, and K.-U. Bletzinger. A framework for stabilized partitioned analysis of thin membrane–wind interaction. *International Journal for Numerical Methods in Fluids*, 54(6-8): 945–963, 2007.

[132] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer. Continuous adjoint approach to the Spalart–Allmaras turbulence model for incompressible flows. *Computers & Fluids*, 38(8):1528–1538, 2009.