THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING
IN
THERMO AND FLUID DYNAMICS

Development of Shape Optimization for Internal Flows

EYSTEINN HELGASON

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2013

Development of Shape Optimization for
Internal Flows
EYSTEINN HELGASON

Thesis for the degree of Licentiate of Engineering 2013:06
ISSN 1652-8565
Department of Applied Mechanics
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Chalmers Reproservice
Gothenburg, Sweden 2013

Development of Shape Optimization for Internal Flows
Thesis for the degree of Licentiate of Engineering in Thermo and Fluid Dynamics
EYSTEINN HELGASON
Department of Applied Mechanics
Chalmers University of Technology

Abstract

This thesis describes the development of an adjoint based optimization method. The goal is to develop a robust method capable of handling multiple design variables, whereas traditional methods have shown to be too costly for many design parameters. The main application is for internal flow geometries within the automotive industry. The advantage of using the adjoint method is that the simulation time becomes independent of the number of design variables.

The continuous adjoint Navier-Stokes equations are presented and simplified for internal flow applications. Contribution from the goal function enter only boundary conditions of the simplified adjoint Navier-Stokes equations. A goal function to minimize total pressure drop is implemented and used in the cases presented.

Two different optimization approaches using the adjoint method were applied. The first one is based on surface sensitivities. The surface sensitivities give information about how the objective function is affected by normal motion of the surface. The sensitivities were coupled to a mesh morphing library in OpenFOAM which diffuses the motion of the boundary nodes to the internal points of the mesh. This method was applied to an inlet pipe with a Reynolds number of $1.9 \cdot 10^5$ based on the diameter at the inlet. The resulting geometry gave a 6.5% decrease in the total pressure drop through the pipe.

In the second approach the sensitivities with respect to motion of the cell center were derived from the Arbitrary-Lagrangian Eulerian formulation of the Navier-Stokes equations. It was shown that the cell sensitivities can be calculated as a post processing step using the results from the adjoint and the primal flow fields. The cell sensitivities were compared to the surface sensitivities and the results show similar behavior for the cells closest to the surface. A method to connect the cell sensitivities to the shape of the geometry with different levels of smoothing is presented. Optimization was performed on a laminar internal flow geometry using the cell sensitivities and applying two different smoothing criteria.

Keywords: computational fluid dynamics, internal flow, optimization, sensitivities, continuous adjoint method.

Acknowledgements

First of all I would like to express my gratitude to my supervisor Professor Siniša Krajnović for giving me the opportunity to work in this exciting field. His motivation and provision of a stimulating research atmosphere is greatly acknowledged.

I would like to thank Per Jonsson, Dennis Konstanzer, Mattias Chevalier, Magnus Berglund, Raymond Reinmann and Stephen Conway from Scania for the interesting and helpful discussions at the steering group meetings. Computational time at Swedish National Infrastructure for Computing (SNIC) at the Center for Scientific Computing at Chalmers (C3SE) is acknowledged.

Special thanks go to Carsten Othmer and Andreas Gitt-Gehrke from Volkswagen for inviting me to stay at VW Salzgitter while working on the cell sensitivities. I would also like thank them for the many helpful discussions and for always being ready to answer my questions.

Many thanks go to my friends and colleagues at the division of Fluid Dynamics for the great working environment. The administrative support by Ulla Lindberg-Thieme and Monica Vargman is gratefully acknowledged. I would like to thank my family and my loving girlfriend Halla for the support and encouragement during this work.

This research is funded by the Swedish Energy Agency (sw. Energimyndigheten). The support of which is gratefully acknowledged.

Eysteinn Helgason
March 31, 2013
Gothenburg

Nomenclature

Latin symbols

A	cell face area
I	smoothing criteria for mesh motion
J	goal function
L	Lagrange function
$\vec{n}$	outwards pointing unit normal vector to the surface
$\vec{R}$	constraints in residual form
$\vec{s}$	mesh motion
$\vec{u}$	adjoint velocity
V	cell volume
$\vec{v}$	primal velocity
p	primal pressure
q	adjoint pressure

Greek symbols

$\vec{\alpha}$	design variable, mesh velocity in the ALE description
β	surface normal displacement
δ	variation
Ω	domain
Γ	boundary of a domain
γ	adjoint mesh motion
λ	Lagrange multiplier

Subscripts

n	outwards pointing normal vector to the surface
t	component of a vector tangential to n

Thesis

This thesis consists of previously unpublished material in Part I and the following appended paper:

Paper A E. Helgason and S. Krajnović. Aerodynamic Shape Optimization of a Pipe using the Adjoint Method. *ASME 2012 International Mechanical Engineering Congress & Exposition, 9-15 November* (2012)

Contents

Part I

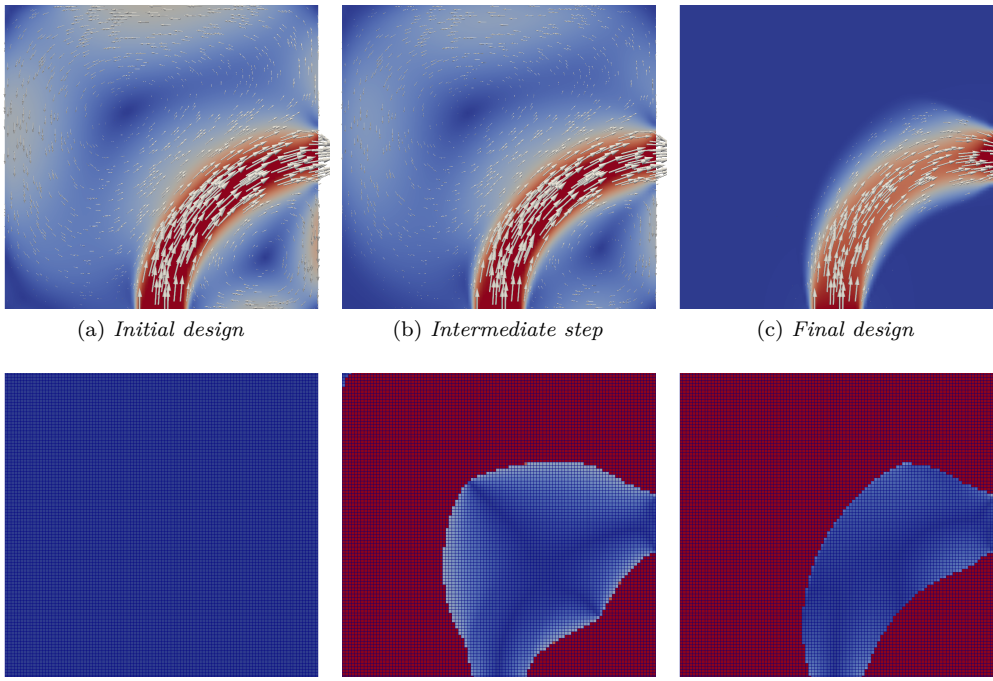
1 Introduction

Internal flow is commonly encountered in our every day life. The hot and cold water in our houses is distributed through pipes and most of us have seen the enormous piping systems in oil refineries. In modern vehicles internal flow plays also an important role. Whether it is the air intake to the engine or the ventilation system of the car the design is important. It would e.g. not be satisfactory if 80% of the ventilation air in a car entered the cabin on the passenger side. In the limited space of the under hood of today's vehicles getting all the components in place can be a puzzle. In the design process for a vehicle the location of the major components is usually decided on before the ducts that connect those components are designed. This can lead to limited design space for placing the internal ducts. A robust automatic design tool can help in creating efficient designs within the often limited design space and has the potential to shorten the design process. With the increasing use of computational fluid dynamics in the vehicle design phase this calls for flexible and robust optimization method. In traditional optimization methods, e.g. evolutionary algorithms [2], the computational cost is highly dependent on the number of design parameters. Restrictions set on the number of design variables can influence the resulting optimal geometry. A possible solution to this problem is the use of adjoint methods, where the computational cost is independent of the number of design parameters.

This thesis is organized in the following way. The introduction includes a description of a typical gradient based optimization process as well as the motivation behind using the adjoint method for optimization. This is followed by general description of the adjoint method and derivation of the continuous adjoint Navier-Stokes equations. Specialization of the continuous adjoint Navier-Stokes equations with respect to internal flow and a connection to the variation on the surface of the geometry is also presented. Ongoing work is presented in the cell sensitivities chapter. The motivation for calculating the sensitivities in the cell center is given and motion of the cell centers is derived from the Arbitrary Lagrangian-Eulerian description. This chapter includes preliminary results from optimization of a laminar internal flow geometry.

1.1 Choice of Optimization Method

Two approaches to optimization in computational fluid dynamics are shape optimization and topological optimization. Topological optimization gives probably the most freedom to the optimization algorithm by not only modifying the boundary shape but also allowing creation of new boundaries [3]. In topological optimization the starting point is to define the design space, that is the maximum size of the geometry. The design space is then meshed and the locations of inlets and outlets defined. In the optimization process each cell acts as a design variable and can be transformed from a fluid cell to a "wall" cell and vice versa. It can be used both when there is no initial design to start from and only the design space is known or when modifying a readily available design. The resulting geometry often doesn't have a smooth surface and needs to be smoothed in a post processing step. When combined with traditional gradient based optimization



(a) *Initial design* (b) *Intermediate step* (c) *Final design*

(d) *Initial design* (e) *Intermediate step* (f) *Final design*

Figure 1.1: *Example of topological optimization where each cell is a design parameter. This is a 2 dimensional case where the inlet is at the bottom and the outlet on the right. The top row shows velocity magnitude and velocity vectors and the lower row shows how the cells are being blocked with porosity. In the beginning the flow is free to flow in the whole domain, (d). The velocity vectors are compared to the results from a potential flow solver. Too much deviation from the alignment of the potential flow solvers indicates a cell that needs to be blocked.*

algorithms topological optimization quickly becomes very computationally expensive. Figure 1.1 shows an example of topological optimization of a 2 dimensional design space with inlet in the middle at the bottom and outlet at the right. Here the angle of a primal flow velocity vector is compared to the angle obtained from the velocity field from a potential flow solver. Each cell is independently validated and either kept as a fluid cell or blocked with a Darcy term in the Navier-Stokes equations, for further explanation on the implementation see [21].

In shape optimization the modifications are done directly on the shape of the geometry and the goal is to minimize a certain goal function while satisfying given constraints. Shape optimization can be very effective when doing fine-tuning on an existing design.

Optimization algorithms used in computational fluid dynamics (CFD) are generally

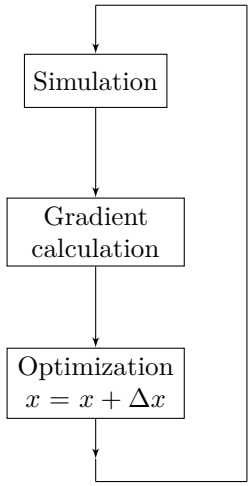


Figure 1.2: *Flow chart for a gradient based optimization method. The most time consuming part is usually the gradient calculations which limits the number of design variables used to describe the geometry.*

highly dependent on the number of design variables. Even with modern computers accurate CFD simulations are computationally expensive and this does therefore often limit the number of design parameters used to describe a geometry.

An example of a typical gradient based optimization loop can be seen in Fig. 1.2. The optimization process begins by evaluating the goal function. This is done by performing a flow simulation and calculating the goal function. The next step is to calculate the gradients with respect to the design parameters. This is normally the most time consuming part. For a numerical differential method like the forward difference,

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}, \tag{1.1}$$

this requires n number of flow simulations, where n is the number of design variables. The geometry is then modified using the information obtained from the gradient calculations and the loop starts again. The loop continues until a certain criteria is met. The total cost of each loop is therefore $n + 1$ number of simulations. The adjoint method allows for these calculations using only two solver calls, one for the primal flow solver and one for the adjoint solver. This makes the adjoint method a feasible option in optimization when dealing with multiple design parameters in computationally expensive CFD simulations.

The adjoint method is a constraint optimization method related to the method of Lagrange multipliers. In a constraint optimization the variables of the function to be maximized or minimized are not completely independent of each other but must fulfill a constraint equation. An example can be to find the shortest distance to the origin from the curve R, see Fig. 1.3. The goal is to minimize J while still fulfilling the constraints, R.

This problem can be solved using various optimization methods and one solution to this problem is to apply method of Lagrange multipliers. This is done by creating a Lagrange function, $L = J + \lambda R$, where λ is called a Lagrange multiplier. The solution is then found by solving $\nabla L = \vec{0}$.

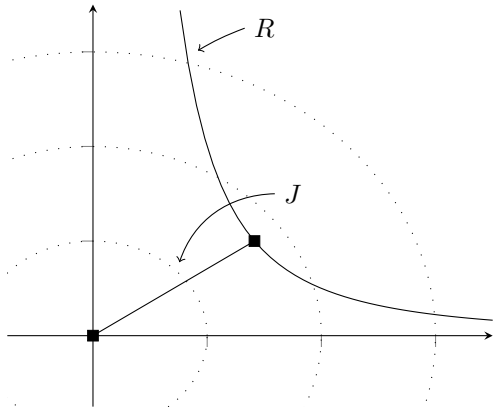


Figure 1.3: *Constraint optimization where the goal is to minimize J while fulfilling the constraints, R.*

Next chapter will focus on the adjoint method and the derivation of the continuous adjoint Navier-Stokes equations.

1.2 Project Outline

This thesis describes a project that focuses on the development of a robust and efficient optimization method for internal flow geometries. The focus is on investigating and utilizing the potentials of the adjoint method to optimize designs with respect to fluid dynamic properties. The application is within internal flow geometries in trucks which can include everything from inlet pipes to engines, exhaust pipes or ventilation systems. Possibilities of open source utilities and commercial solutions will be investigated as well as developing new methods with the aim of making the optimization process as robust and automatic as possible. The simulations presented in this thesis are done using the open source CFD toolbox OpenFOAM[19]. OpenFOAM is a non-staggered Finite Volume Method code written in C++ using object oriented approach.

The project is done in collaboration with Scania AB and is financially supported by the Swedish Energy Agency (sw. Energimyndigheten).

2 The Adjoint Method

The adjoint method recently gained popularity in the field of computational fluid dynamics for its ability to drastically decrease the simulation time in gradient based optimization processes when dealing with large number of design variables.

The adjoint method has a history in optimal control theory dating back to the 1950's and since then it has been used in various different applications ranging from optimization in computer graphics [15] to pricing options in financial applications [1]. One of the first appearances of the adjoint method in fluid dynamics is from 1973 when the continuous adjoint Stokes equations where presented by Pironneau [24]. The adjoint method for optimization gained popularity within the field of fluid dynamics following the publication of Jameson [10] in 1988, where he applied the adjoint Euler equations to transonic 2D airfoils and a decade later optimized a 3D wing using the adjoint Navier-Stokes equations [11].

The adjoint optimization process is generally implemented by using either the continuous or the discrete approach, see Fig. 2.1. In both of the approaches we start from the analytical form the the primal equations. Where in the continuous approach the primal equations are linearized, then the adjoint equations derived from the linearized primal equations and in the end the adjoint equations discretized. The discrete method starts with the discretization of the primal equations which are linearized and then finally transposed.

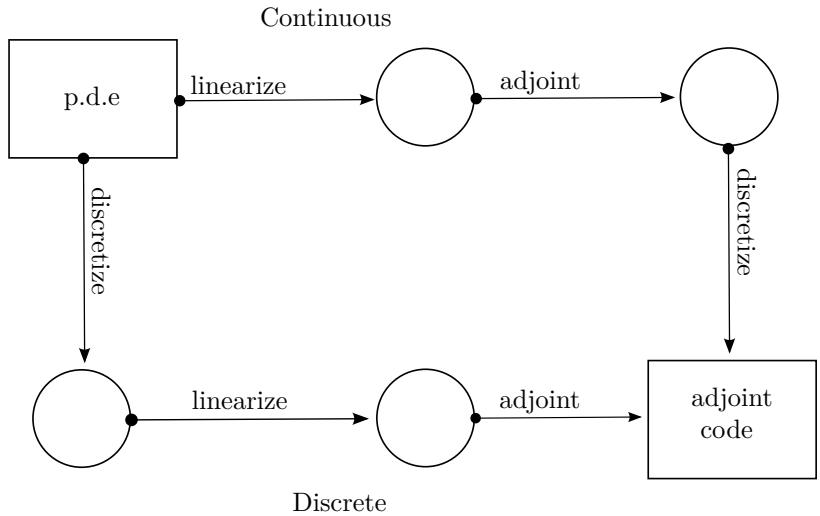


Figure 2.1: *The continuous vs the discrete adjoint approach. The continuous approach starts by linearizing the equations, deriving the adjoint equations in their analytical form and then discretizing the equations. The discrete adjoint approach on the other hand starts with the discretization of the equations, then linearizing and finally transposing the equations in order to end at the adjoint code.*

Both implementations converge to the same result in the limit of infinite grid resolution while each of the two approaches has their advantages and disadvantages. One of the advantage of the discrete approach is that it gives the exact gradient of the discretized goal function and allows for the optimization process to converge fully. Automatic Differentiation (AD) tools can also be used in automation of the derivation of the source terms in the adjoint equations. Some of the advantages of the continuous adjoint equations are generally lower memory consumption and that the adjoint equations can be implemented in a straight forward manner in an advanced C++ framework such as OpenFOAM®[19, 22]. For a more detailed discussion on the differences between the continuous and discrete implementation of the adjoint method the reader is referred to [7, 16, 23].

The next section gives a short explanation on the derivation of the continuous adjoint method followed by the derivation of the steady state, incompressible adjoint Navier-Stokes equations and simplifications made to the equations and boundary conditions for internal flow application.

2.1 The Continuous Adjoint Method

Given a goal function J, the total variation of J can be expressed as the sum of the variation with respect to the flow variables, $\vec{w}$, and the design variables, $\vec{\alpha}$:

$$\delta J = \underbrace{\frac{\partial J}{\partial \vec{w}} \delta \vec{w}}_{\text{flow}} + \underbrace{\frac{\partial J}{\partial \vec{\alpha}} \delta \vec{\alpha}}_{\text{design}}. \tag{2.1}$$

The flow equations in residual form can be written as

$$\vec{R}(\vec{w}, \vec{\alpha}) = \vec{0} \tag{2.2}$$

and expressing the total variation of $\vec{R}$ in the same way as the goal function, i.e. as the sum of the variation with respect to the flow variables, $\vec{w}$, and the design variables, $\vec{\alpha}$, gives:

$$\delta \vec{R} = \frac{\partial \vec{R}}{\partial \vec{w}} \delta \vec{w} + \frac{\partial \vec{R}}{\partial \vec{\alpha}} \delta \vec{\alpha} = \vec{0}. \tag{2.3}$$

Now we can multiply Eq. 2.3 with an arbitrary Lagrange multiplier, $\vec{\lambda}$,

$$\vec{\lambda}^T \delta \vec{R} = \vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{w}} \delta \vec{w} + \vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{\alpha}} \delta \vec{\alpha} = 0. \tag{2.4}$$

By adding Eq. 2.4 to Eq. 2.1 the total variation of J can be expressed as

$$\delta L \equiv \delta J = \left[\frac{\partial J}{\partial \vec{w}} + \vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{w}} \right] \delta \vec{w} + \left[\frac{\partial J}{\partial \vec{\alpha}} + \vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{\alpha}} \right] \delta \vec{\alpha}. \tag{2.5}$$

It can be noted that the variation of the Lagrangian, L, could also be defined by subtracting Eq. 2.4 from Eq. 2.1. We are now free to choose the value of the Lagrange multiplier so that the variation with respect to the flow variables vanishes, i.e.

$$\vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{w}} \delta \vec{w} = -\frac{\partial J}{\partial \vec{w}} \delta \vec{w}. \tag{2.6}$$

The adjoint operator G^* to an operator G is defined as

$$< \vec{u}, G\vec{v} > = < G^* \vec{u}, \vec{v} > \tag{2.7}$$

for all $\vec{u}$ and $\vec{v}$ where $< \vec{u}, \vec{v} >$ is the inner product

$$< \vec{u}, \vec{v} > \equiv \int_\Omega \vec{v}^T \vec{u} \, \mathrm{d}\Omega. \tag{2.8}$$

Integrating by parts the left hand side of Eq. 2.6 allows for the $\partial \vec{R}/\partial \vec{w}$ operator to be moved on to the Lagrange multipliers. That process results in a system of equations, the so called adjoint equations, and by solving that system gives the values of the adjoint variables, the Lagrange multipliers. The total variation can then be written as

$$\delta J = \frac{\partial J}{\partial \vec{\alpha}} \delta \vec{\alpha} + \vec{\lambda}^T \frac{\partial \vec{R}}{\partial \vec{\alpha}} \delta \vec{\alpha} \tag{2.9}$$

where the variation is calculated using the variation of the goal function and an inner product between variation of the flow equations with respect to the design variable $\vec{\alpha}$ and the Lagrange multipliers, the solution to the adjoint equations.

The most demanding part in the derivation of the continuous adjoint equations is the process of moving the derivatives to the adjoint variables and choosing the appropriate boundary conditions. Next section describes this process for the steady state, incompressible Navier-Stokes equations followed by simplifications that can be made for internal flow and implementation of the goal function for total pressure drop.

2.1.1 Derivation of the Adjoint Navier-Stokes Equations

The Navier-Stokes for an incompressible, single-phase, steady state flow consist of three momentum equations and a continuity equation:

$$\begin{aligned} (R_1, R_2, R_3)^T &= \rho v_j v_{i,j} + p_{,i} - (\mu(v_{i,j} + v_{j,i}))_{,j}, \\ R_4 &= -v_{i,i}, \end{aligned} \tag{2.10}$$

presented here in residual form where $v_{i=x,y,z}$ is the primal flow velocity vector, p is the primal pressure and ρ is the density. The viscosity term, μ, denotes the effective viscosity, i.e. sum of the molecular and turbulent viscosity modeled by the eddy viscosity turbulence models. Einstein summation notation has been used, where repeated indices imply summation, and comma in the indical notation indicates a partial derivative with respect to each coordinate x_i, e.g. $\nabla \vec{u} = \partial_i u_j = u_{j,i}$.

Now we can set up an augmented goal function with the Navier-Stokes equations as the constraints:

$$L = J + \vec{\lambda}^T \vec{R}, \tag{2.11}$$

where J is the goal function we want to minimize and $\vec{\lambda}$ is vector of the adjoint variables, $\vec{\lambda} = (u_i, q)$, where u_i is the adjoint velocity and q is the adjoint pressure.

The total variation of L is:

$$\delta L = \underbrace{\delta_{\vec{w}} L}_{\text{flow}} + \underbrace{\delta_{\vec{\alpha}} L}_{\text{design}} . \tag{2.12}$$

The total variation of the Navier-Stokes equations is now split up into contributions from the flow field, $\vec{w}$, and the design variables, $\vec{\alpha}$:

$$\delta \vec{R} = \underbrace{\delta_{\vec{w}} \vec{R}}_{\text{flow}} + \underbrace{\delta_{\vec{\alpha}} \vec{R}}_{\text{design}} . \tag{2.13}$$

Focusing now on the variation with respect to the flow field, as it is the part that will be solved in the adjoint equation system. Setting the variation of the augmented goal function with respect to the flow variables, $\vec{w}$, to zero, as shown in Eq. 2.6 gives the basis for the adjoint equations,

$$\delta_{\vec{w}} L = \delta_{\vec{w}} J + \int_\Omega \lambda_i \delta_{\vec{w}} R_i \, \mathrm{d}\Omega = 0. \tag{2.14}$$

The variation with respect to the flow field can further be split up into variation with respect to the primal flow field, $\vec{v}$, and the primal pressure, p,

$$\delta_{\vec{w}} R = \delta_{\vec{v}} R + \delta_p R. \tag{2.15}$$

The variation with respect to the flow field can now be written as:

$$\delta_{\vec{w}} L = \delta_{\vec{v}} J + \delta_p J + \int_\Omega (u_i, q) \delta_{\vec{v}} \vec{R} \, \mathrm{d}\Omega + \int_\Omega (u_i, q) \delta_p \vec{R} \, \mathrm{d}\Omega. \tag{2.16}$$

Writing out the variation with respect to the primal velocity flow field, $\delta_{\vec{v}} \vec{R}$, gives

$$\begin{aligned} \delta_{\vec{v}} (R_1, R_2, R_3)^T &= \rho \delta v_j v_{i,j} + \rho v_j \delta v_{i,j} - (\mu(\delta v_{i,j} + \delta v_{j,i}))_{,j}, \\ \delta_{\vec{v}} R_4 &= -\delta v_{i,i} \end{aligned} \tag{2.17}$$

and the variation with respect to the primal pressure, $\delta_p \vec{R}$, is

$$\begin{aligned} \delta_p (R_1, R_2, R_3)^T &= \delta p_{,i}, \\ \delta_p R_4 &= 0. \end{aligned} \tag{2.18}$$

The variation of the eddy viscosity, μ, has been neglected here and this assumption is exactly correct for laminar flow regions. This approximation is common in the derivation of the continuous Adjoint Navier-Stokes equations, generally referred to as "frozen turbulence"

[26, 11]. For recent development in the field of continuous adjoint turbulence models and adjoint wall functions, see for example [30, 31]. There are varying views on the effect of this approximation from being almost negligible in engineering applications and leading to an error of the order of a few percent compared to the exact full adjoint based gradient [14] to in some cases errors up to 50% [18]. The memory requirements and run time is though undoubtedly considerably lower with the "frozen turbulence" assumption as no adjoint turbulence equations need to be solved. For a study on the effect of common approximations in the implementation of the adjoint method see e.g. [6].

Inserting the linearized constraints, Eq. 2.17 and Eq. 2.18, into Eq. 2.16 gives the starting point for the derivation of the adjoint Navier-Stokes equations:

$$0 = \delta_{\vec{v}} J + \delta_p J + \int_\Omega (u_i \rho \delta v_j v_{i,j} + u_i \rho v_j \delta v_{i,j} - u_i(\mu(\delta v_{i,j} + \delta v_{j,i}))_{,j} + u_i \delta p_{,i} - q \delta v_{i,i}) \, \mathrm{d}\Omega. \tag{2.19}$$

Now we proceed with moving the derivatives from the primal flow variables, δv_i and δp, to the adjoint variables, u_i and q, using integration by parts.

Starting from Gauss's theorem which states that the sum of all sources minus the sum of all sinks equals the net flow out of a region, i.e.

$$\int_\Omega u_{i,i} \, \mathrm{d}\Omega = \int_\Gamma u_i n_i \, \mathrm{d}\Gamma, \tag{2.20}$$

where Γ is the closed boundary surface of Ω and n_i is the outward normal unit vector to Γ.

The product rule reads:

$$(f, g)_{,i} = f_{,i} \, g + f \, g_{,i}. \tag{2.21}$$

The integration by parts is done by reversing the product rule, $f_{,i} \, g = (f, g)_{,i} - f \, g_{,i}$, and applying the Gauss theorem, Eq. 2.20, on each of the terms in the integral in Eq. 2.19.

Assuming incompressibility we obtain the following identities:

$$\begin{aligned}\int_\Omega u_i\rho\delta v_j v_{i,j}\,\mathrm{d}\Omega &= \int_\Gamma \rho n_j u_i \delta v_j v_i\,\mathrm{d}\Gamma - \int_\Omega \rho(u_i\delta v_j)_{,j} v_i \mathrm{d}\Omega,\\ &= \int_\Gamma \rho n_j u_i \delta v_j v_i\,\mathrm{d}\Gamma - \int_\Omega \rho u_{i,j}\delta v_j v_i\,\mathrm{d}\Omega - \int_\Omega \rho u_i \cancelto{0}{\delta v_{j,j}} v_i \mathrm{d}\Omega,\\ &= \int_\Gamma \rho n_i u_j v_j \delta v_i\,\mathrm{d}\Gamma - \int_\Omega \rho u_{j,i} v_j \delta v_i\,\mathrm{d}\Omega. \end{aligned} \tag{2.22a}$$

$$\begin{aligned}\int_\Omega u_i\rho v_j\delta v_{i,j}\,\mathrm{d}\Omega &= \int_\Gamma \rho u_i v_j \delta v_i n_j\,\mathrm{d}\Gamma - \int_\Omega \rho(u_i v_j)_{,j}\delta v_i\,\mathrm{d}\Omega,\\ &= \int_\Gamma \rho u_i v_j \delta v_i n_j\,\mathrm{d}\Gamma - \int_\Omega \rho u_{i,j} v_j \delta v_i\,\mathrm{d}\Omega - \int_\Omega \rho u_i \cancelto{0}{v_{j,j}}\delta v_i\,\mathrm{d}\Omega,\\ &= \int_\Gamma \rho u_i v_j n_j \delta v_i\,\mathrm{d}\Gamma - \int_\Omega \rho u_{i,j} v_j \delta v_i\,\mathrm{d}\Omega. \end{aligned} \tag{2.22b}$$

$$\begin{aligned}\int_\Omega -u_i(\mu(\delta v_{i,j}+\delta v_{j,i}))_{,j}\,\mathrm{d}\Omega &= -\int_\Gamma \mu u_i(\delta v_{i,j}+\delta v_{j,i})n_j\,\mathrm{d}\Gamma + \int_\Omega \mu(u_{i,j}\delta v_{i,j} + u_{i,j}\delta v_{j,i}),\\ &= -\int_\Gamma \mu u_i(\delta v_{i,j}+\delta v_{j,i})n_j\,\mathrm{d}\Gamma + \int_\Gamma \mu(u_{i,j}+u_{j,i})n_j\delta v_i \mathrm{d}\Gamma\\ &\quad - \int_\Omega (\mu(u_{i,j}+u_{j,i}))_{,j}\delta v_i\,\mathrm{d}\Omega. \end{aligned} \tag{2.22c}$$

$$\int_\Omega u_i \delta p_{,i}\,\mathrm{d}\Omega = \int_\Gamma u_i n_i \delta p\,\mathrm{d}\Gamma - \int_\Omega u_{i,i}\delta p\,\mathrm{d}\Omega. \tag{2.22d}$$

$$\int_\Omega -q\delta v_{i,i}\,\mathrm{d}\Omega = -\int_\Gamma q n_i \delta v_i\,\mathrm{d}\Gamma + \int_\Omega q_{,i}\delta v_i\,\mathrm{d}\Omega. \tag{2.22e}$$

The cost function can be decomposed into contributions from the interior, Ω, and the boundary, Γ, i.e.

$$J = \int_\Gamma J_\Gamma\,\mathrm{d}\Gamma + \int_\Omega J_\Omega\,\mathrm{d}\Omega. \tag{2.23}$$

Replacing the terms in Eq. 2.19 with the terms from Eq. 2.22 and the decomposed cost function, Eq. 2.23, results in:

$$\begin{aligned} 0 = &\int_\Omega \left(\frac{\partial J_\Omega}{\partial v_i} - \rho u_{j,i}v_j - \rho u_{i,j}v_j - (\mu(u_{i,j}+u_{j,i}))_{,j} + q_{,i}\right)\delta v_i\,\mathrm{d}\Omega\\ &+ \int_\Gamma \left(\frac{\partial J_\Gamma}{\partial v_i} + \rho n_i u_j v_j + \rho u_i v_j n_j + \mu(u_{i,j}+u_{j,i})n_j - q n_i\right)\delta v_i\,\mathrm{d}\Gamma\\ &- \int_\Gamma \mu u_i(\delta v_{i,j}+\delta v_{j,i})n_j\,\mathrm{d}\Gamma\\ &+ \int_\Omega \left(\frac{\partial J_\Omega}{\partial p} - u_{i,i}\right)\delta p\,\mathrm{d}\Omega + \int_\Gamma \left(\frac{\partial J_\Gamma}{\partial p} + u_i n_i\right)\delta p\,\mathrm{d}\Gamma \end{aligned} \tag{2.24}$$

Combining the terms that depend on δv_i and δp and knowing that Eq. 2.24 is fulfilled for all variation of the primal flow field that satisfies the primal Navier-Stokes equations

implies that each of the integrals must become zero independently of each other. From this condition on the integrals in Eq. 2.24 the adjoint Navier-Stokes equations can be derived as:

$$-\rho(u_{j,i} + u_{i,j})v_j = -q_{,i} + (\mu(u_{i,j} + u_{j,i}))_{,j} - \frac{\partial J_\Omega}{\partial v_i}, \tag{2.25a}$$

$$u_{i,i} = \frac{\partial J_\Omega}{\partial p}. \tag{2.25b}$$

With the following boundary conditions:

$$0 = \int_\Gamma \left(\frac{\partial J_\Gamma}{\partial v_i} + \rho n_i u_j v_j + \rho u_i v_j n_j + \mu(u_{i,j} + u_{j,i})n_j - q n_i \right) \delta v_i \, \mathrm{d}\Gamma - \int_\Gamma \mu u_i (\delta v_{i,j} + \delta v_{j,i}) n_j \, \mathrm{d}\Gamma, \tag{2.26a}$$

$$0 = \int_\Gamma \left(\frac{\partial J_\Gamma}{\partial p} + u_i n_i \right) \delta p \, \mathrm{d}\Gamma \tag{2.26b}$$

The adjoint Navier-Stokes equations have some similarities with the primal Navier-Stokes equations. The main difference is though in the contribution from the goal function to the source terms and that the adjoint convection is upstream of the primal flow field, the first term on the left in the adjoint momentum equations.

The adjoint equations reverse the propagation of information which has the effect that in unsteady adjoint simulations the primal flow field needs to be calculated from the beginning to the end, as usually, and then the adjoint flow is solved backwards in time. At each point in time the primal flow field needs to be retrieved either by storing all the primal flow time steps or by recalculating the primal flow field from a solution at a previous time step. The memory requirements needed to store every time step make this very demanding for industrial applications. To limit the memory requirements so called check pointing method can be used where limited number of time steps, called check points, spread around the time line are saved. The primal simulations are then rerun and the intermediate time steps saved when the adjoint solver arrives at each of the check points [29, 17, 28].

2.1.2 The Adjoint Navier-Stokes Equations for Internal Flow

In the continuous approach the adjoint equations and boundary conditions can be simplified by considering the application that the adjoint method is to be applied on. This section focuses on simplifications that assume internal flow and follows the work done by Othmer in [20]. The adjoint Navier-Stokes equations in Eq. 2.25 accompanied by the boundary conditions in Eq. 2.26 can be simplified with respect to application in internal flow. Goal functions applied to internal flow simulations often include only integrals over the surface, i.e. inlet, outlet and walls, and not the internal domain. Assuming that the goal function is zero inside the domain eliminates the last terms in Eq. 2.25a and 2.25b. This reduces

the adjoint momentum and continuity equations to

$$-\rho(u_{j,i} + u_{i,j})v_j = -q_{,i} + (\mu(u_{i,j} + u_{j,i}))_{,j}, \tag{2.27a}$$

$$u_{i,i} = 0. \tag{2.27b}$$

This simplifies the implementation of new goal functions into the solver as the adjoint equations are now independent of the goal function. The goal function now only enters the boundary conditions of the adjoint velocity and pressure fields.

We follow the simplifications made by Othmer[20] and assume that part of the viscosity terms on the boundary can be neglected

$$\int_\Gamma \mu \left(u_{j,i} n_j \delta v_i - u_i \delta v_{j,i} n_j\right) \mathrm{d}\Gamma \approx 0. \tag{2.28}$$

The boundary conditions that need to be fulfilled are now:

$$\begin{aligned} 0 = &\int_\Gamma \left(\frac{\partial J_\Gamma}{\partial v_i} + \rho n_i u_j v_j + \rho u_i v_j n_j + \mu u_{i,j} n_j - q n_i \right) \delta v_i \, \mathrm{d}\Gamma \\ &- \int_\Gamma \mu u_i \delta v_{i,j} n_j \, \mathrm{d}\Gamma, \end{aligned} \tag{2.29a}$$

$$0 = \int_\Gamma \left(\frac{\partial J_\Gamma}{\partial p} + u_i n_i \right) \delta p \, \mathrm{d}\Gamma \tag{2.29b}$$

Some of the terms in the boundary conditions for the adjoint Navier-Stokes equations can be neglected by considering the three common boundaries in internal flow simulations, inlet, outlet and walls.

Inlet and walls: We usually have a prescribed value for the inlet velocity and no-slip boundary condition for the walls which results in $\delta v_i = 0$ and therefore the first integral in Eq. 2.29a is zero. For the second integral, by decomposing the δv_i term into contributions from the normal component to the surface , $n_i \delta v_i$ and the tangential component, $\delta \vec{v}_t = \delta v_i - n_i \delta v_i$. Then applying the continuity equation, $\delta v_{i,i} = 0$, assuming $\delta \vec{v}_t = \vec{0}$ at the inlet and walls the remaining term from the continuity equation becomes

$$n_j (n_i \delta v_i)_{,j} = 0. \tag{2.30}$$

This decomposition of the continuity equation can be seen as

$$\nabla \cdot \delta \vec{v} = \underbrace{\frac{\partial \delta \vec{v}_x}{\partial x}}_{\vec{n} \cdot \nabla (\delta \vec{v} \cdot \vec{n})} + \underbrace{\frac{\partial \delta \vec{v}_y}{\partial y} + \frac{\partial \delta \vec{v}_z}{\partial z}}_{\nabla_t \cdot \delta v_t = 0} \tag{2.31}$$

in the case where the normal component to the surface, $\vec{n}$, is aligned along the x-axis. Using the same decomposition for the term in the second integral in Eq. 2.29a we get that

$$n_j \delta v_{i,j} = \cancelto{0}{n_j (n_i \delta v_i)_{,j}} + n_j (\delta v_t)_{,j} \tag{2.32a}$$

From this we conclude that the tangential component of the adjoint velocity needs to be set to zero to fulfill the boundary conditions. No boundary condition is prescribed for the adjoint pressure, q, and Neumann boundary conditions are applied, similar as for the primal pressure.

$$\vec{u}_t = \vec{0}, \tag{2.33a}$$

$$n_i u_i = -\frac{\partial J_\Gamma}{\partial p}, \tag{2.33b}$$

$$n_i q_{,i} = 0. \tag{2.33c}$$

Outlet: For the outlet we have Neumann boundary conditions for velocity and the second integral of Eq. 2.29a is zero. The pressure is set to zero at the outlet so $\delta p = 0$ and the integral in Eq. 2.29b is therefore zero. By splitting the first integral of Eq. 2.29a into normal and tangential components the boundary conditions for the adjoint pressure, q, and the tangential component of the adjoint velocity can be determined as

$$q = \frac{\partial J_\Gamma}{\partial (v_i n_i)} + \rho u_j v_j + \rho u_i n_i v_j n_j + \mu (n_i u_i)_{,j} n_j, \tag{2.34a}$$

$$0 = \frac{\partial J_\Gamma}{\partial \vec{v}_t} + \rho \vec{u}_t v_j n_j + \mu \vec{u}_{t,j} n_j. \tag{2.34b}$$

The normal gradient of the normal component of the adjoint velocity, u_i, is determined from the continuity equation using the same decomposition as shown in Eq. 2.31

2.1.3 Total Pressure Drop

An important property of internal flow geometries is the total pressure drop through the geometry. The total pressure drop is measured from the inlet to the outlet and the goal function can written as:

$$J = \int_{Inlet} c(p/\rho + \frac{1}{2}v^2)\,\mathrm{d}\Gamma - \int_{Outlet} c(p/\rho + \frac{1}{2}v^2)\,\mathrm{d}\Gamma. \tag{2.35}$$

where a constant, $c = 1\,\rho\,\mathrm{m/s}$, is added to compensate for missing units.

The derivative of the goal function for total pressure drop in the boundary conditions at the inlet and walls is:

$$\frac{\partial J_\Gamma}{\partial p} = -c/\rho,$$

and the outlet:

$$\frac{\partial J_\Gamma}{\partial (v_i n_i)} = -c v_i n_i,$$

$$\frac{\partial J_\Gamma}{\partial \vec{v}_t} = -c \vec{v}_t.$$

This results in the adjoint Navier-Stokes equations for internal flow:

$$-\rho\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)v_j=-\frac{\partial q}{\partial x_i}+\frac{\partial}{\partial x_j}\left(\mu(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i})\right), \tag{2.37a}$$

$$\frac{\partial u_i}{\partial x_i}=0. \tag{2.37b}$$

with the following boundary conditions to minimize total pressure drop:

Inlet & walls:

$$\vec{u}_t=0, \tag{2.38a}$$

$$n_i u_i=-1\,\mathrm{m/s}, \tag{2.38b}$$

$$n_i\frac{\partial q}{\partial x_i}=0. \tag{2.38c}$$

Outlet:

$$q=-cv_i n_i+\rho u_j v_j+\rho u_i n_i v_j n_j+\mu\frac{\partial n_i u_i}{\partial x_j}n_j, \tag{2.39a}$$

$$\vec{0}=-c\vec{v_t}+\rho\vec{u_t}v_i n_i+\mu\frac{\partial\vec{u}_t}{\partial x_j}n_j, \tag{2.39b}$$

$$n_j\frac{\partial u_i n_i}{\partial x_j}=-\frac{\partial u_{t_i}}{\partial n_{t_i}} \tag{2.39c}$$

2.1.4 Shape Optimization

The adjoint equations for internal flow have been derived with the appropriate boundary conditions to minimize total pressure drop and the next step is the connection to the description of the geometry. In adjoint optimization we have to first choose between topological optimization or shape optimization. When working with fine tuning of a design direct modification of the geometry is appropriate and leads us to the surface sensitivities.

The total variation of the Lagrangian is now described as

$$\delta L=\delta_{\vec{w}}L+\delta_\beta L \tag{2.40}$$

The variation with respect to the flow variables, $\vec{w}$, has already been dealt with in sections 2.1.1 and 2.1.2. What is left is the variation with respect to normal displacement of the surface, β,

$$\delta_\beta L=\delta_\beta J+\delta_\beta\int_\Omega(u_i,q)R_i\,\mathrm{d}\Omega \tag{2.41}$$

The surface sensitivities for goal functions that do not depend on the internal domain and with fixed inlet and outlet, as is often the case for internal flow geometries, is calculated

directly from the primal and the adjoint flow field. It has been shown, see [25, 27, 20], that the variation with respect to normal motion of the surface can be approximated as:

$$\frac{\partial L}{\partial \beta} \approx -A\nu(\vec{n} \cdot \nabla)\vec{u_t} \cdot (\vec{n} \cdot \nabla)\vec{v_t} \tag{2.42}$$

where A is the face area of the surface cell.

The surface sensitivities along with the boundary conditions for total pressure drop are applied to an internal flow geometry in paper A.

3 Cell Sensitivity

In previous chapters two different approaches to optimize internal flow geometries have been mentioned, topological optimization and shape optimization. The methods are very different in how they modify the geometry and each of them has its own feasible application in the optimization process. The topological optimization is ideal for an initial design of the geometry. Each cell is its own design variable and the location of the cells are fixed. There are also often no real boundary conditions present between the blocked cells and the cells which the flow is free to go through, as the cells are commonly blocked using porosity. This tends to give a rugged surface which is not very feasible as a final design. The rough surface can be avoided by using immersed boundary method as shown in [9]. The surface sensitivities allow for a direct modification on the surface and are well-suited when fine tuning a design.

The motivation behind this chapter was to develop a method that can give larger deformations than the surface sensitivities but does not work on the fixed cells as in the topological optimization. The method could then work as an intermediate step in a design chain that starts from the design space, where the draft of the design is made using topological optimization, and end with fine tuning using the surface sensitivities.

3.1 Arbitrary Lagrangian-Eulerian

The Lagrangian description of motion is based on following an associated material particle during motion. This can allow for easier tracking of free surfaces and interfaces between different materials. The Eulerian description is where the computational mesh is fixed and the continuum moves with respect to the grid which allows for large distortions in the continuum motion. In the Arbitrary Lagrangian-Eulerian (ALE) description the nodes of the computational domain can be moved with the continuum as is done in the Lagrangian description, held fixed as in the Eulerian approach or moved in some arbitrary way in between.

The ALE formulation of the Navier-Stokes equations is the following [4, 5]

$$\rho\frac{\partial v_i}{\partial t} + \rho(v_j - \alpha_j)v_{i,j} = -p_{,i} + (\mu(u_{i,j} + u_{j,i}))_{,j},$$
$$v_{i,i} = 0,$$

where $\vec{\alpha}$ is the mesh velocity.

The adjoint solvers currently used in industrial applications are generally steady state. It is possible to assume that the ALE process is steady state even though the physical meaning is not completely clear, see e.g. [4]. The ALE formulation then reduces to

$$\rho(v_j - \alpha_j)v_{i,j} = -p_{,i} + (\mu(u_{i,j} + u_{j,i}))_{,j}, \tag{3.1a}$$
$$v_{i,i} = 0. \tag{3.1b}$$

This equation is equivalent to the Navier-Stokes equations except for the second term on the left hand side in the momentum equation. By setting the mesh velocity term, $\vec{\alpha}$,

in Eq. 3.1a to zero we end up with the general Eulerian approach commonly used for fluids and by giving it a value we are giving a velocity to cells in the mesh(or assuming the mesh is in motion).

The adjoint Navier-Stokes with the boundary conditions derived from the total pressure drop have been presented, see section 2.1.3: Eq. 2.37, 2.38 and 2.39. Now we turn our focus to the additional term which shows up in the ALE description, that is the second term in Eq. 3.1a. In the optimization process we want to modify the mesh in a favorable direction with respect to a specific goal function. The proposed method is to calculate the variation with respect to the mesh motion, $\vec{\alpha}$. After the variation has been calculated the mesh is modified accordingly while still keeping the term $\vec{\alpha}$ zero between mesh updates. The constraints now become

$$\vec{R}_{1,2,3} = \vec{R}_{\text{N-S}} + \vec{R}_{\alpha} = \rho v_j v_{i,j} - \rho \alpha_j v_{i,j} + p_{,i} - (\mu(v_{i,j} + v_{j,i}))_{,j} \tag{3.2}$$

where $\vec{R}_{\alpha} = -\rho \alpha_j v_{i,j}$.

The ALE description gives an extra term in the Navier-Stokes equations and therefore we could expect there to be also an extra term in the adjoint ALE Navier-Stokes equations or contribution to the boundary conditions for the adjoint field. Calculating the variation of this additional term, $\vec{R}_{\alpha}$, with respect to the flow velocity, $\vec{v}$, gives:

$$\delta_{\vec{v}} \vec{R}_{\alpha} = -\rho \alpha_j \delta v_{i,j} \tag{3.3}$$

Now setting up the second term of the Lagrangian by multiplying with the adjoint velocity and integrating over the domain and moving the derivative over to the adjoint variable by integrating by parts results in

$$\begin{aligned}
\int_{\Omega} u_i \delta \vec{R}_{\alpha} \, \mathrm{d}\Omega &= -\int_{\Omega} u_i \rho \alpha_j \delta v_{i,j} \, \mathrm{d}\Omega, && \text{(3.4a)} \\
&= -\int_{\Gamma} n_j u_i \rho \alpha_j \delta v_i \, \mathrm{d}\Gamma + \int_{\Omega} \rho (u_i \alpha_j)_{,j} \delta v_i \, \mathrm{d}\Omega, \\
&= -\int_{\Gamma} n_j u_i \rho \alpha_j \delta v_i \, \mathrm{d}\Gamma + \int_{\Omega} \rho u_{i,j} \alpha_j \delta v_i \, \mathrm{d}\Omega + \int_{\Omega} \rho u_i \alpha_{j,j} \delta v_i \, \mathrm{d}\Omega. && \text{(3.4b)}
\end{aligned}$$

The α_i term will be kept zero between mesh updates, the last two terms in Eq. 3.4b are therefore zero and no additional contribution is in the adjoint Navier-Stokes equations. We focus now on the boundary integral and assume boundaries commonly used in internal flow, i.e. inlet, walls and outlet. For the boundary conditions for the inlet and walls we have that $\delta v_i = 0$ which makes the boundary integral go to zero. For the outlet we assume the location of the outlet to be fixed and therefore normal contribution from α_j to be zero, i.e. $n_j \alpha_j = 0$. This makes the boundary integral vanish at the outlet.

Calculating the gradient with respect to $\vec{\alpha}$ becomes now only a post-processing step after solving the primal and adjoint flow. The goal function is generally independent of $\vec{\alpha}$ and leads to $\frac{\partial J}{\partial \vec{\alpha}_i} = 0$. The gradient of the Lagrangian with respect to $\vec{\alpha}_i$, that is motion

of cell i, can be written as

$$\frac{\partial L}{\partial \vec{\alpha}_i} = \frac{\partial J}{\partial \vec{\alpha}_i} + \int_\Omega \vec{u} \cdot \frac{\partial R_{\vec{\alpha}}}{\partial \vec{\alpha}_i} \, \mathrm{d}\Omega \tag{3.5a}$$

$$= 0 - \int_\Omega \vec{u} \cdot \frac{\partial \rho \vec{\alpha} \cdot \nabla \vec{v}}{\partial \vec{\alpha}_i} \, \mathrm{d}\Omega$$

$$= -\rho \nabla \vec{v} \cdot \vec{u} V_i \tag{3.5b}$$

where V_i is the cell volume. This shows that the variation with respect to the mesh motion derived from the Arbitrary Lagrangian-Eulerian formulation of the Navier-Stokes equations can be calculated as a post processing step after solving the primal and adjoint flow field.

3.1.1 Comparison to the Surface Sensitivities

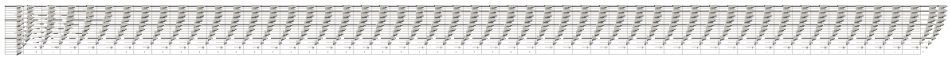
The cell sensitivities even though calculated in the cell center instead of at the surface should give similar behavior as the surface sensitivities for the cells closest to the wall. This section shows a comparison between the cell sensitivities and the surface sensitivities. Starting with a 2 dimensional case, see Fig. 3.1. The inlet is at the left and outlet on the right. The size of the domain is 1.0 m x 0.1 m and the Reynolds number is 5.

Figure 3.1: *Simple 2 dimensional test case, the domain size is 1.0 m x 0.1 m.*

Lower part of the domain is shown in Fig. 3.2 with results from the primal and adjoint velocity vectors along with the sensitivity to the mesh motion in the ALE formulation, equation 3.5b and the surfaces sensitivities introduced in section 2.1.4, Eq. 2.42.

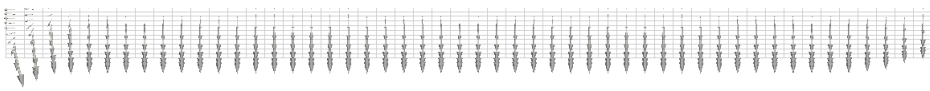
Figure 3.2 shows that the cell sensitivities are all pointing outwards of the domain as expected. The cell sensitivity vectors are not forced to be normal to the surface, as is the case with the surface sensitivities. The magnitude of the cell sensitivity vectors closest to the wall is considerably smaller than for the surface sensitivities.

The cell sensitivities in the cells closest to the surface have been compared to the surface sensitivities from paper A, see Fig. 3.3 and Fig. 3.4. The sensitivities are not in the same scale, the cell sensitivities in the first cell are approximately 2-3 times smaller than the surface sensitivities. The first case is a laminar flow case with the inlet at the left and the outlet at the right. The computational grid contains 225 000 hexahedral cells. The inlet boundary conditions of the primal velocity is $\vec{v} = 1\,\mathrm{m/s}$ with a Reynolds number of 340 based on the hydraulic diameter of the pipe at the inlet.

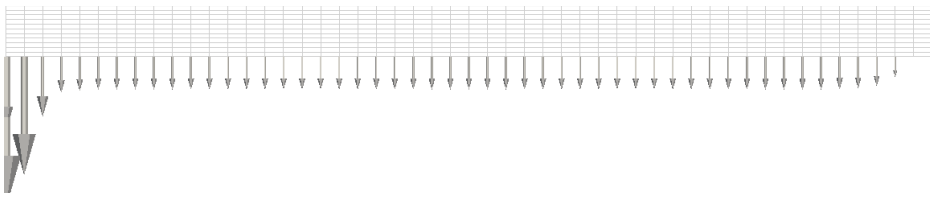


(a) *Primal flow,* $\vec{v}$

(b) *Adjoint flow,* $\vec{u}$



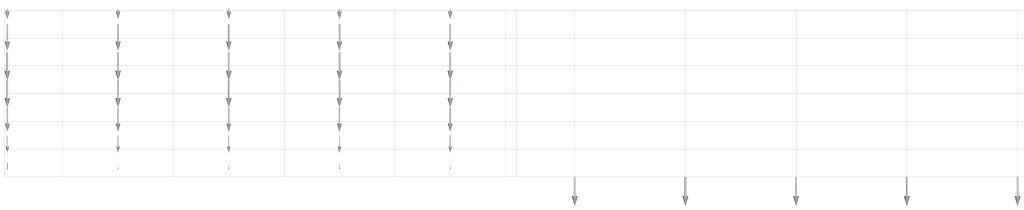
(c) *Cell sensitivities*



(d) *Surface sensitivities*

(e) *Primal flow,* $\vec{v}$

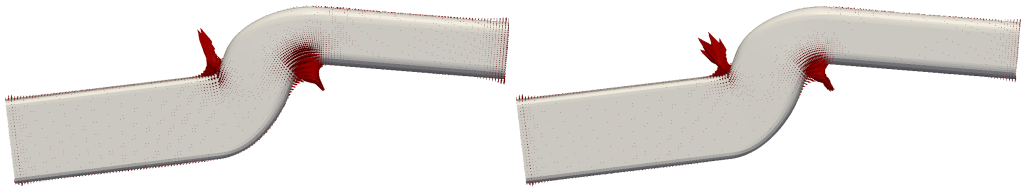
(f) *Adjoint flow,* $\vec{u}$



(g) *Cell sensitivities*

(h) *Surface sensitivities*

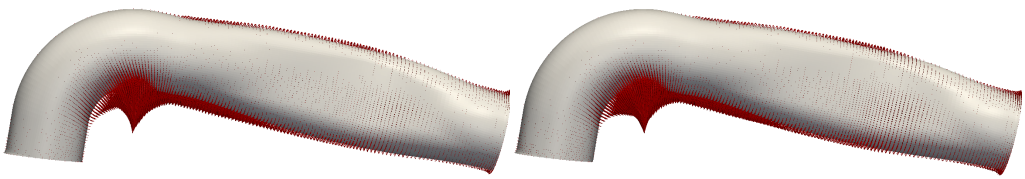
Figure 3.2: *Lower half of the domain shown in Fig. 3.1 along with a figure showing the behavior close to the wall. Velocity vectors (a) and (e), adjoint velocity (b) and (f),* $\frac{\partial L}{\partial \vec{\alpha}}$ *(c) and (g) and the surface sensitivities (d) and (h). Figures (g) and (h) show the cell sensitivity vectors and the surface sensitivities in the same scale close to the wall.*



(a) *Cell sensitivities* (b) *Surface sensitivities*

Figure 3.3: *Comparison of the shape of the cell sensitivities with the surface sensitivities for a laminar case. The shape is similar while the cell sensitivities are roughly 3 times smaller than the surface sensitivities in this case.*

Figure 3.4 shows only the part that was optimized of the second geometry in paper A. The inlet velocity is $40\,\mathrm{m/s}$ entering from the right and the Reynolds number is $1.9 \cdot 10^5$ based on the diameter of the pipe at the inlet. The standard k-ϵ turbulence model [12] was used for the primal flow field along with standard wall functions.



(a) *Cell sensitivities* (b) *Surface sensitivities*

Figure 3.4: *Comparison of the shape of the cell sensitivities with the surface sensitivities for a turbulent case. The shape is similar while the cell sensitivities are roughly 2 times smaller than the surface sensitivities in this case.*

The behavior of the cell and surface sensitivities is very similar close to the surface, the cell sensitivities are though considerably smaller (approximately 2-3 times). It should also be noted that the cell sensitivities are not necessarily orthogonal to the surface as is the case with the surfaces sensitivities.

The cell sensitivities obtained from Eq. 3.5b give a sensitivity map for the whole domain. Directly applying those sensitivities to the motion of the mesh results in a bad mesh quality after only a few iterations. Therefore a coupling is needed between the cell sensitivities and the motion of the mesh that gives a smoother motion of the nodes in the domain. In common mesh morphing methods the boundary of the mesh is modified and the motion of the internal nodes coupled to the motion of the boundary in order to preserve the quality of the mesh while deforming the geometry. One way to do this is to diffuse the velocity of the boundary nodes to the internal cells by setting the motion on the boundary as boundary conditions for the Laplace equation.

3.2 Coupling with Mesh Motion

Mesh motion is usually controlled by some algorithm in order to minimize cell distortion. For overview of mesh motion methods in CFD see for example [13]. In this approach we will couple the cell sensitivities to the mesh motion using a mesh velocity smoothing algorithm based on the Poisson equation. The mesh motion is governed by the following diffusion equation:

$$\nabla^2 \vec{s} = -\vec{f} \tag{3.6}$$

where $\vec{s}$ is the mesh motion and $\vec{f}$ is a source term. For the case when $\vec{f}$ is zero this equation becomes the Laplace's equation, $\nabla^2 \vec{s} = \vec{0}$, which governed the mesh motion in paper A.

The idea is to calculate how variation in the source term can minimize a goal function that is created from the sensitivity with respect to the mesh motion derived from the ALE description, section 3.1.
The goal function to minimize is:

$$J = \int_\Omega (\vec{s} - \vec{\beta})^2 \, \mathrm{d}\Omega \tag{3.7}$$

where $\beta = \frac{\delta L}{\delta \vec{\alpha}_i}$ is the variation of the cells obtained from our steady ALE calculation, Eq. 3.5b.
The Lagrangian, with $\vec{\gamma}$ as our Lagrange multiplier, reads

$$L = \int_\Omega (\vec{s} - \vec{\beta})^2 \, \mathrm{d}\Omega + \int_\Omega \vec{\gamma} \cdot (\nabla^2 \vec{s} + \vec{f}) \, \mathrm{d}\Omega. \tag{3.8}$$

Total variation of the Lagrangian can be split into contributions from the mesh motion, $\vec{s}$, and the source term, $\vec{f}$,

$$\delta L = \delta_{\vec{s}} L + \delta_{\vec{f}} L. \tag{3.9}$$

The variation of the Lagrangian with respect to the mesh motion, $\vec{s}$, can now be written as

$$\delta_{\vec{s}} L = \int_\Omega 2(\vec{s} - \vec{\beta}) \delta\vec{s} \, \mathrm{d}\Omega + \int_\Omega \vec{\gamma} \cdot \nabla^2 \delta\vec{s} \, \mathrm{d}\Omega. \tag{3.10}$$

Integrating by parts the last term gives the identity

$$\int_\Omega \vec{\gamma} \cdot \nabla^2 \delta\vec{s} \, \mathrm{d}\Omega = \int_\Omega \nabla^2 \vec{\gamma} \cdot \delta\vec{s} \, \mathrm{d}\Omega + \int_\Gamma \vec{\gamma} \frac{\partial \delta\vec{s}}{\partial n} \, \mathrm{d}\Gamma - \int_\Gamma \frac{\partial \vec{\gamma}}{\partial n} \delta\vec{s} \, \mathrm{d}\Gamma \tag{3.11}$$

By inserting Eq. 3.11 into Eq. 3.10 and splitting the goal function into contribution from the boundary, Γ, and the interior, Ω, we get:

$$\begin{aligned} \delta_{\vec{s}} L = & \int_\Omega 2(\vec{s} - \vec{\beta}) \delta\vec{s} \, \mathrm{d}\Omega + \int_\Gamma 2(\vec{s} - \vec{\beta}) \delta\vec{s} \, \mathrm{d}\Gamma \\ & + \int_\Omega \nabla^2 \vec{\gamma} \cdot \delta\vec{s} \, \mathrm{d}\Omega + \int_\Gamma \vec{\gamma} \frac{\partial \delta\vec{s}}{\partial n} \, \mathrm{d}\Gamma - \int_\Gamma \frac{\partial \vec{\gamma}}{\partial n} \delta\vec{s} \, \mathrm{d}\Gamma \end{aligned} \tag{3.12}$$

The goal function gives only contributions on the interior of he domain, where the values for β are calculated, and therefore the boundary contribution from the goal function is zero.

Collecting the terms from the interior and the boundary results in

$$\begin{aligned}\delta_{\vec{s}} L = &\int_\Omega \left(2(\vec{s}-\vec{\beta}) + \nabla^2\vec{\gamma}\right)\cdot\delta\vec{s}\,\mathrm{d}\Omega \\ &+ \int_\Gamma \vec{\gamma}\frac{\partial\delta\vec{s}}{\partial n}\,\mathrm{d}\Gamma - \int_\Gamma \frac{\partial\vec{\gamma}}{\partial n}\delta\vec{s}\,\mathrm{d}\Gamma\end{aligned} \tag{3.13}$$

Equation 3.13 should be valid for any arbitrary variation in the mesh motion, for this condition to hold the integrals need to vanish individually. This leads to the adjoint equations:

$$\nabla^2\vec{\gamma} = 2(\vec{\beta}-\vec{s}) \tag{3.14}$$

With the following boundary conditions:

$$\vec{\gamma}\frac{\partial\delta\vec{s}}{\partial n} = \delta\vec{s}\frac{\partial\vec{\gamma}}{\partial n} \tag{3.15}$$

To fulfill the boundary conditions we consider two cases, one where the boundary is moving and second one where the boundary is fixed. This gives the following boundary conditions:

Moving boundary:

$$\begin{aligned}\frac{\partial\vec{s}}{\partial n} &= \vec{0}, \\ \frac{\partial\vec{\gamma}}{\partial n} &= \vec{0}.\end{aligned} \tag{3.16}$$

Fixed boundary:

$$\begin{aligned}\vec{s} &= \vec{0} \\ \vec{\gamma} &= \vec{0}\end{aligned} \tag{3.17}$$

The variation with respect to the source term, $\vec{f}$, can now be calculated as:

$$\begin{aligned}\frac{\partial L}{\partial\vec{f}_i} &= \int_\Omega \vec{\gamma}_i\,\mathrm{d}\Omega, \\ &= \vec{\gamma}_i V_i.\end{aligned} \tag{3.18}$$

where V_i denotes the volume of cell i and γ_i is obtained from the solution of the adjoint equation, Eq. 3.14, with the boundary conditions from Eqs. 3.16 and 3.17. The gradient is then added to the source term $\vec{f}$ in Eq. 3.6. This results in two coupled equations that need to be solved. If we assume the small variation of the source term leads to small variation in the solution of the primal equation 3.6, we can add the gradient directly to $\vec{s}$ and solve the adjoint equation with the updated mesh motion term. This assumption has

been compared to solving the coupled Poisson equations, Eqs. 3.6 and 3.14, for the cases shown in Fig. 3.5b-e. The difference in the resulting mesh motion term, $\vec{s}$, was negligible while approximately 20 times more iterations were needed for the coupled equations.

The gradient with respect to the source term is now accumulated in $\vec{s}$ using method of steepest decent

$$\vec{s}_{\text{new}} = \vec{s}_{\text{old}} - c\gamma \tag{3.19}$$

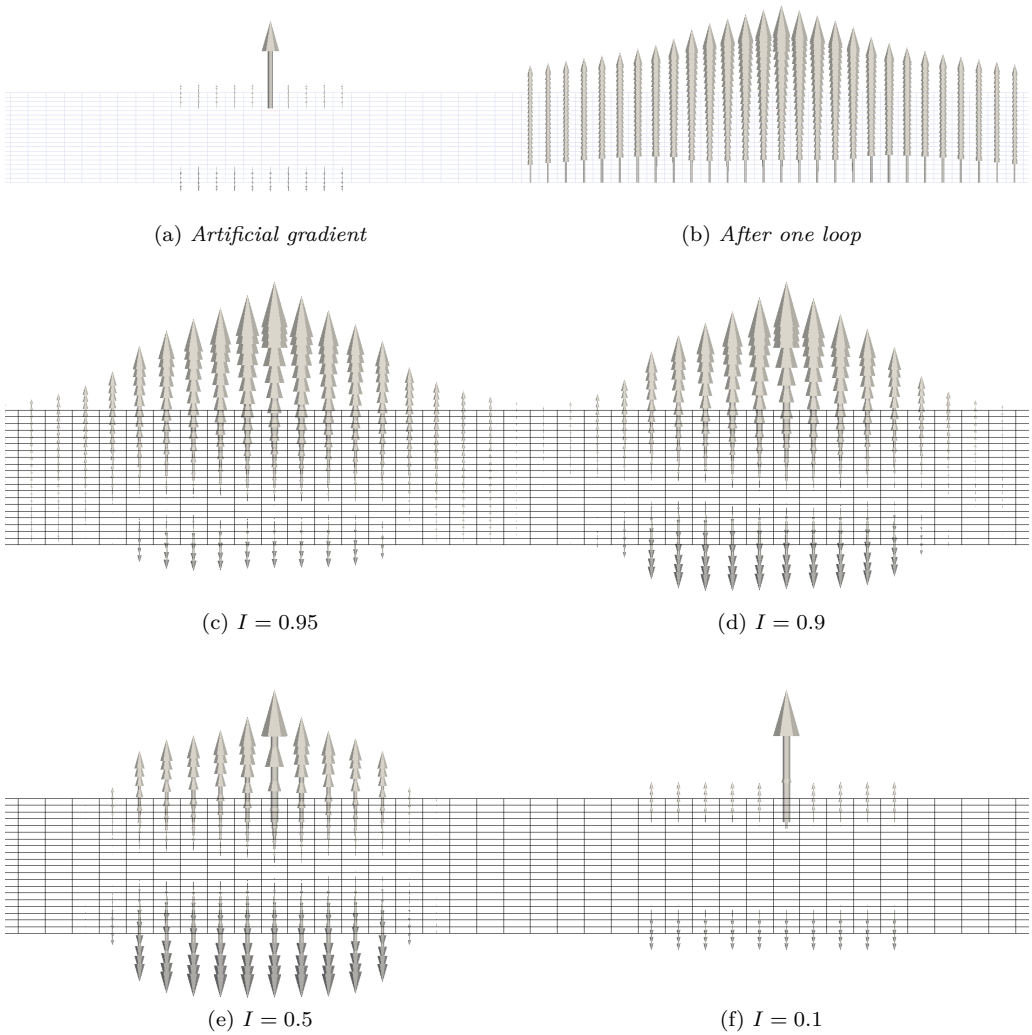
where c denotes the step size.

A condition related to the goal function is set where the optimization is considered converged when the condition I falls below a certain criteria

$$I = \frac{\int_\Omega (\vec{s} - \vec{\beta})^2 \, \mathrm{d}\Omega}{\int_\Omega \vec{\beta}^2 \, \mathrm{d}\Omega}. \tag{3.20}$$

3.2.1 Simple 2-Dimensional Case

In this simple test case the effect from different convergence criteria set on I is investigated. The domain is the same as used when comparing the cell sensitivities to the surface sensitivities in section 3.1.1. An artificial gradient is now added to the cell centers, implying favorable mesh motion. The gradient resembles the gradient obtained from the cell sensitivities, Eq. 3.5b. One gradient is 10 times larger than the others while the others are equally large. The artificial gradients are shown in Fig. 3.5a. The adjoint equation is now solved using Eq. 3.14 with fixed boundary conditions on the left and right while zero gradient boundary condition is applied on the upper and lower wall. After one loop in the optimization process the effect from the large artificial gradient on the whole domain can be seen, Fig. 3.5b. Figures 3.5 (c)-(f) show the result for different conditions on I. The vectors are not in the same scale, ranging from 0.2 for the largest one in Fig. 3.5c to 8 in 3.5f. The artificial gradients have the length equal to 1 except for the large one which has the length 10.



(a) *Artificial gradient*

(b) *After one loop*

(c) $I = 0.95$

(d) $I = 0.9$

(e) $I = 0.5$

(f) $I = 0.1$

Figure 3.5: *(a) Artificial gradients are added to the domain and the resulting mesh motion shown in (b)-(f). (b) After one loop through the mesh coupling algorithm. (c)-(f) show the mesh motion for different criteria on I. Note that the vectors are not in the same scale.*

3.3 Optimization using Cell Sensitivities

The methods introduced in sections 3.1 and 3.2 have been tested separately. The cell sensitivities show similarities with the surface sensitivities and the effect from different criteria set on I was presented. The methods are now combined and the resulting flow chart for the optimization presented in Fig. 3.6.

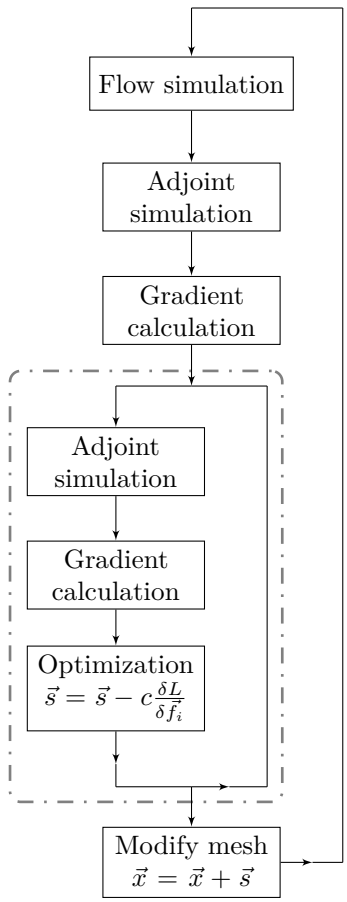


Figure 3.6: *Flow chart of the optimization process. The dotted box contains the optimization loop used for smoothing of the gradients. The cell sensitivities are used as an input to the mesh motion. When the convergence criteria is reached the mesh motion, $\vec{s}$, is directly added to the location of the points in the mesh, denoted with $\vec{x}$, and the loop starts again.*

The geometry optimized here is the same as the laminar one in paper A. It consists of three parts, inlet, outlet and walls (see Fig 3.7). The computational grid contains 225 000 hexahedral cells. The inlet boundary conditions of the primal velocity are $\vec{v} = 1\,\mathrm{m/s}$, and the kinematic viscosity, $\nu = 1.6 \cdot 10^{-4}\,\mathrm{m^2/s}$, results in a laminar flow with a Reynolds

number of 340 based on the hydraulic diameter of the pipe at the inlet. No slip boundary condition is applied to the wall. For the adjoint solver, no slip and zero gradient boundary conditions for the adjoint velocity and pressure respectively are applied at the wall. The adjoint boundary conditions for inlet and wall are set according to Eq 2.38 and the outlet boundary conditions according to Eq. 2.39 in order to minimize the total pressure drop through the pipe. The whole wall section from the inlet to the outlet is selected for optimization.

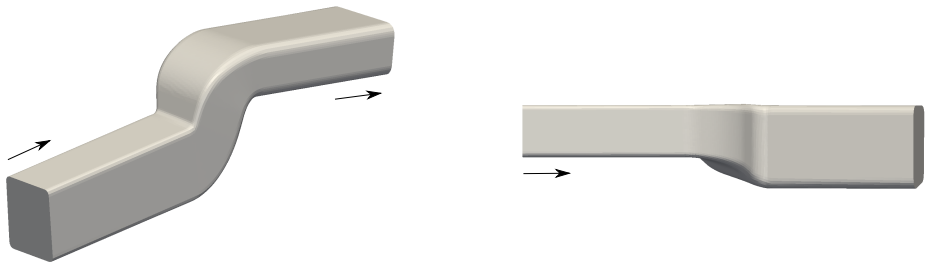
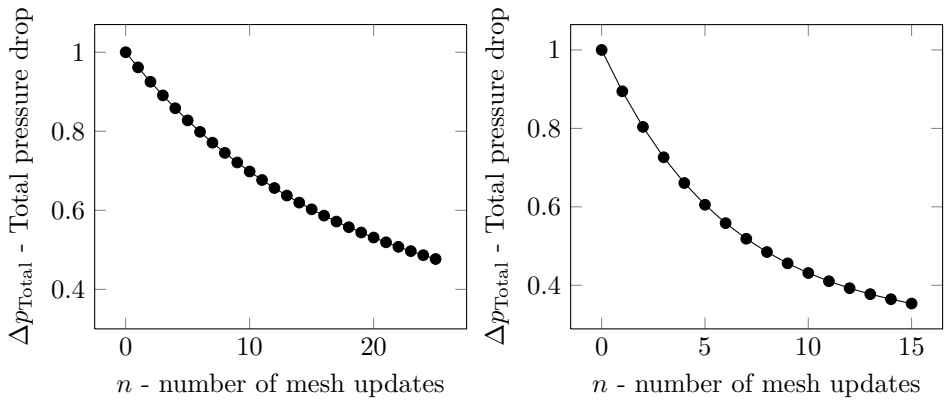


Figure 3.7: *The S-bend seen from the side and below. The flow direction is from left to right as depicted by the arrows.*

The results are presented for two different criteria on I, $I = 0.5$ and $I = 0.95$. The total pressure drop through the pipe after each mesh deformation for the two cases is shown in Fig. 3.8.

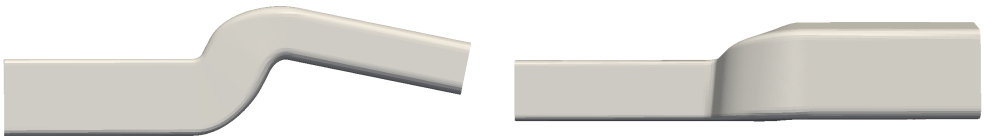


(a) $I = 0.5$ (b) $I = 0.95$

Figure 3.8: *The total pressure drop compared to the original design for two different criteria on I.*

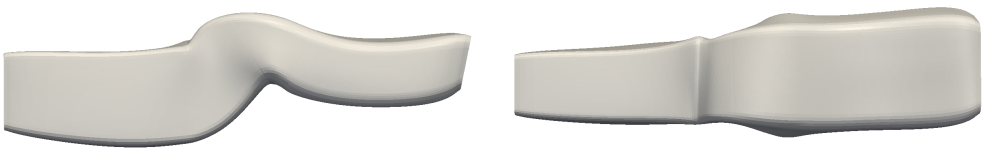
Comparison between the original geometry and the optimal design for the two cases is

shown in Fig. 3.9. The final geometry for the case with $I = 0.5$ gives 52% lower total pressure drop through the pipe. The largest modifications to the geometry are at the lower part of the bend and also the second half of the pipe has expanded as shown in Fig. 3.9b. For the case where $I = 0.95$ the total pressure drop decreased by 65%. Figure 3.9c shows that the modification is not confined to the region around the bend. In this case the mesh motion affects a larger region which resulted in the bend moving downwards and the whole geometry has expanded.



(a) *Orginal geometry*

(b) $I = 0.5$



(c) $I = 0.95$

Figure 3.9: *(a) The original geometry seen from the side and top. The inlet is at the left and the outlet ad the right. The original geometry can be compared to the final geometry obtained using two different conditions on I, (b) and (c).*

The results show the largest deformation in the region around the bend. In this optimization the mesh morphing is directly coupled to the mesh motion, $\vec{s}$. This is done by interpolating the values from the cell centers to the points in the mesh. When the convergence criteria set for I is decreased the mesh motion vector resembles more the cell sensitivities and the modifications to the mesh can result in lower quality mesh. This can explain why the lower pressure drop through the pipe was obtained in the case with higher value for I. The results for this optimization process are promising. Larger modifications to the geometry were made than in previous attempts using the surface sensitivities, see Paper A. The next step will be to apply this method to a turbulent flow pipe.

4 Conclusions and Future Work

Two strategies of using continuous adjoint method were explored. The first relies on the existing method using surface sensitivities [27, 20]. The second approach is a newly proposed method using the sensitivities in the cell center. The motivation behind this new method was to design an approach that works as an intermediate step between topological optimization and the surface sensitivities. It should allow for larger deformations than the surface sensitivities but not work on a fixed mesh as in the case of topological optimization.

The method of surface sensitivities was applied in paper A. First to a simple laminar flow pipe and then to a real industrial case. The method of cell sensitivities was applied to a laminar flow pipe. The results when using the surface sensitivities connected with the mesh motion libraries in OpenFOAM gave a fine tuning of an existing design when used with high quality mesh. The cell sensitivities coupled with the mesh motion were shown to give larger deformation for the laminar flow pipe than the surface sensitivities.

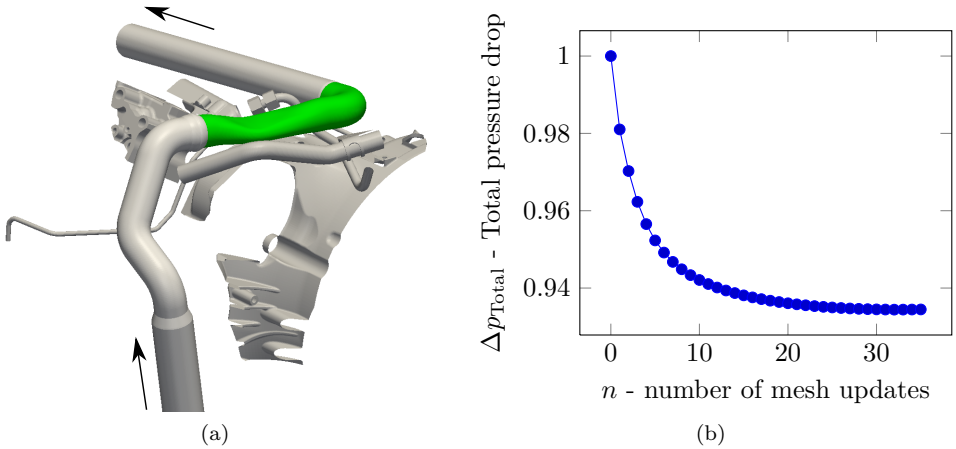
Future work will involve further developing the cell sensitivity method and apply it to the second inlet pipe geometry that was described in paper A. The code works in parallel and is quite robust, a possible bottleneck is the fixed step size used in the optimization algorithm for the mesh motion which can in some instances be slow to converge. Implementing a more robust optimization algorithm with variable step size is therefore a possible solution to that problem.

It would be interesting to investigate other uses of the sensitivities in the cell center. Having the sensitivities in the cell center gives the possibility to use the adjoint method to fine tune a mesh, e.g. in order to minimize velocity gradients in the mesh. This could possibly be achieved by fixing the boundaries and moving only the internal nodes.

5 Summary of Paper A

5.1 Aerodynamic Shape Optimization of a Pipe using the Adjoint Method

Paper A presents results from optimization of an internal flow geometry using the adjoint method. The surface sensitivities, introduced in section 2.1.4 are connected to the mesh motion solvers in OpenFOAM®. The goal function is to minimize the total pressure drop and for that purpose the appropriate boundary conditions derived in section 2.1.3 are implemented for the adjoint flow field. An inlet pipe to the exhaust gas re-circulator cooler of a truck engine is optimized, see Fig 5.1. The mesh contains 1.0 million hexahedral cells and the inlet velocity is 40 m/s, which is equivalent to boundary conditions at cruising speed, resulting in a Reynolds number of $1.9 \cdot 10^5$. The standard k-ϵ turbulence model [12] was applied and standard wall functions used for the primal flow field.



(a) (b)

Figure 5.1: *(a) The pipe showing in green the area chosen for optimization. (b) Total pressure drop for the whole inlet pipe. After 35 mesh updates the total pressure drop has decreased by 6.5%.*

Total simulation time was 7 hours on 16 cores in which a total of 35 mesh updates were performed. The resulting geometry gives 6.5% lower total pressure drop compared to the original design.

References

[1] Y. Achdou and O. Pironneau. *Computational methods for option pricing*. Vol. 30. Society for Industrial Mathematics, 2005.

[2] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press on Demand, 1996.

[3] T. Borrvall and J. Petersson. Topology optimization of fluids in Stokes flow. *Internation Journal for Numerical Methods in Fluids* **41** (2003), 77–107.

[4] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodríguez-Ferran. "Arbitrary Lagrangian–Eulerian Methods". *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd, 2004. ISBN: 9780470091357. DOI: 10.1002/0470091355.ecm009.

[5] A. Duran. "A numerical formulation to solve the ALE Navier-Stokes equations applied to the withdrawal of magma chambers". PhD thesis. 2000.

[6] R. P. Dwight and J. Brezillon. Effect of Various Approximations of the Discrete Adjoint on Gradient-Based Optimization. *AIAA* **2006-0690** (2006).

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

[8] E. Helgason and S. Krajnović. Aerodynamic Shape Optimization of a Pipe using the Adjoint Method. *ASME 2012 International Mechanical Engineering Congress & Exposition, 9-15 November* (2012).

[9] C. Hinterberger and M. Olesen. Industrial application of continuous adjoint flow solvers for the optimization of automotive exhaust systems. *CFD & OPTIMIZATION* May (2011), 1–17.

[10] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing* **3**.3 (1988), 233–260. DOI: 10.1007/BF01061285.

[11] A. Jameson and L. Martinelli. Optimum Aerodynamic Design Using the Navier – Stokes Equations 1. *Theoretical and Computational Fluid* (1998), 213–237.

[12] B. E. Launder and B. I. Spalding. The Numerical Computation of Turbulent Flows. *Computer Methods in Applied Mechanics and Engineering* **3** (1974), 269–289.

[13] R. Löhner. *Applied CFD Techniques: An Introduction based on Finite Element Methods, second edition*. John Wiley & Sons Ltd, 2008.

[14] A. C. Marta and S. Shankaran. On the handling of turbulence equations in RANS adjoint solvers. *Computers & Fluids* **74** (Mar. 2013), 102–113. DOI: 10.1016/j.compfluid.2013.01.012.

[15] A. McNamara, Z. Treuille A.and Popović, and J. Stam. "Fluid control using the adjoint method". *ACM Transactions On Graphics (TOG)*. Vol. 23. 3. ACM. 2004, pp. 449–456.

[16] S. Nadarajah and A. Jameson. A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *AIAA paper* **667** (2000), 20.

[17] S.K. Nadarajah and C. Tatossian. Multi-objective aerodynamic shape optimization for unsteady viscous flows. English. *Optimization and Engineering* **11** (1 2010), 67–106. DOI: 10.1007/s11081-008-9036-4.

[18] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier-Stokes equations. *AIAA Journal* **37** (Jan. 1999), 1411–1419. DOI: 10.2514/3.14337.

[19] *OpenFOAM® open source CFD toolbox*. 2013. URL: http://www.openfoam.com.

[20] 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** (2008), 861–877. DOI: 10.1002/fld.1770.

[21] C. Othmer, T. Kaminski, and R. Giering. Computation of topological sensitivities in fluid dynamics: cost function versatility. *ECCOMAS CFD* (2006), 1–12.

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

[23] J.E.V. Peter and R.P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids* **39**.3 (Mar. 2010), 373–391. DOI: 10.1016/j.compfluid.2009.09.013.

[24] O. Pironneau. On optimum profiles in Stokes flow. *Journal of Fluid Mechanics* **59**.01 (1973), 117–128.

[25] O. Pironneau. Optimal Shape Design for Elliptic Systems. *Springer-Verlag* (1984).

[26] B. Soemarwoto. "The Variational Method for Aerodynamic Optimization Using the Navier-Stokes Equations Operated by Universities Space Research Association". 97. 1997.

[27] O. Soto and R. Lohner. On the computation of flow sensitivities from boundary integrals. *AIAA Paper* (2004).

[28] D.N. Srinath and S. Mittal. An adjoint method for shape optimization in unsteady viscous flows. *Journal of Computational Physics* **229**.6 (2010), 1994–2008. DOI: 10.1016/j.jcp.2009.11.019.

[29] Q. Wang, P. Moin, and G. Iaccarino. Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation. *SIAM Journal on Scientific Computing* **31**.4 (2009), 2549–2567.

[30] A.S. Zymaris, D.I. Papadimitriou, K.C. Giannakoglou, and C. Othmer. Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics* **229**.13 (July 2010), 5228–5245. DOI: 10.1016/j.jcp.2010.03.037.

[31] 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 (Sept. 2009), 1528–1538. DOI: 10.1016/j.compfluid.2008.12.006.