Master's Thesis

Topology and Shape optimization for CFD-Computational Fluid Dynamics

Student
Esteban Betancur Valencia

Responsible professor
Manuel Julio García Ruiz

Department of Mechanical Engineering
Universidad EAFIT
Medellín

June, 2014

Abstract

On this work a CFD optimization problem is treated from two different points of view. In one hand, topology optimization using a homogenization method based on the Brinkmann penalization is presented, implemented using the finite elements method and optimized with a mesh adaptation step. Secondly, a shape optimization method based on Hadamard boundary variation using differentiation with respect to the domain is developed, implemented and tested. Finally, a coupling of both methods taking advantage of their own attributes is proposed and tested. The expected results are obtained.

Keywords: CFD optimization, topology optimization, shape derivative, two step optimization, Finite elements.

Contents

Introduction

The constant development of Computer-Aided Engineering (CAE) has made increasingly important the use of computational mechanics for design, re-design and general engineering studies. Concerning fluids, the design of aeronautical applications, wind energy mills, biomedical appliances for the circulatory system or microfluidic devices are examples of Computational Fluid Dynamics (CFD) applications.

The optimization is, by definition, the procedure to make a system or design as effective or functional as possible. In other words, is the selection of the best solution from all the feasible ones to perform a function. The optimization using CFD has been studied from different points of view and multiple approaches have been developed. The first applications can be seen on minimizing the drag on wing profiles (See for example:Pironneau (1974)). More recently, general optimization procedures are created. Mohammadi and Pironneau (2002) show developments on shape optimization for aeronautical applications based on the adjoint method. A gathering of different methods and specific applications is presented on Thvenin and Janiga (2008) with an engineering emphasis. And a wide variety of studies are continuosly reported according to the problem features.

The topology optimization is very useful to obtain an initial design. The use of two wings per side for an airplane in case of one or the number of pipes to transport a fluid from multiple inputs to multiple outputs are questions that topology optimization can answer. On the other hand, if the topology is already defined, small boundary variations can be handled with shape optimization. For instance, small modifications on the shape of high speed vehicles can lead to significant improvements on the drag.

Therefore, a complete optimum design should involve two steps, first find an initial shape with a defined topology and then mildly deform this shape to obtain a better result. This thesis deals with the complete optimization process, a two step optimization procedure is proposed and tested. On chapter 1, the development and implementation of

a CFD topology optimization method is presented. Chapter 2 deals with the shape optimization method based on shape derivatives. Finally, chapter 3 describes the procedure to combine this methods in two steps and some application to numerical examples.

Chapter 1

CFD topology optimization using SIMP method

The shape optimization with numerical methods can be tackled with a wide variety of tools. A CFD optimization process using a Solid Isotropic Material with Penalization-SIMP under a Finite Element method is developed. The SIMP method is capable of doing topology changes having an implicit representation of the fluid boundary with a density function.

This chapter explains the process of solving the topology optimization of a Stokes flow system using a SIMP formulation with a material distribution model proposed by Khadra et al. (2000) and used first by Borrvall and Petersson (2003) on fluid optimization methods. An optimization program with Finite Elements is implemented using FreeFem++ (Hecht (2012)). And a mesh adaptation step is added to improve the computational efficency of the program.

1.1 Solid domain representation

A fictitious solid domain is aproached using a Brinkmann penalization on the Stokes equation based on the theory of porous media. A term αv defined as the Darcy-friction force is added to the stokes equation. The α coefficient can be defined as an inverse permeability of the medium where the fluid moves.

$$-\mu\Delta\boldsymbol{u} + \alpha\boldsymbol{u} + \nabla p = \boldsymbol{f} \tag{1.1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{1.2}$$

The α coefficient is defined to vary according to a density function to have no significance on the fluid region but to take a great value on the solid domain, imposing a very low permeability and making the velocity tend to cero.

This material distribution and multimaterial domains have been highly implemented for topology optimization of structural mechanics, see for example Borrvall and Petersson (2001) or Garcia and Steven (1999). As it is for solids, a density function ρ is defined over the entire domain, where $0 \leq \rho \leq 1$. By definition, this function takes the value of $\rho = 1$ on the fluid domain and $\rho = 0$ on the solid one.

Now, the inverse permeability α can be defined as a function of the density. The use of penalization on this function makes α tend to be a discrete value in order to obtain "defined" boundaries.

$$\alpha(\rho) = \alpha_{max} + (\alpha_{min} - \alpha_{max})\rho\frac{1+q}{\rho+q} \tag{1.3}$$

Where α_{max} and α_{min} are the values of this coefficient for the solid and fluid domain respectively. Borrvall and Petersson (2003) propose $\alpha_{min} = 2.5\mu/100^2 \approx 0$ and $\alpha_{max} = 2.5\mu/0.01^2 \approx \infty$. The coefficient q is the penalization parameter ($0 < q$), the smaller it is, the α function is more penalized (See figure 1.1).

With this solid representation, the complete domain is discretized and the solid is implicitly defined by the density function ρ. On Lee (2012) Chapter 2, a validation of this method for the Navier-Stokes flow simulation is done, and the best results (concerning non-slip walls and zero-velocity in solid regions) is obtained with the coefficient α_{max} greater than 10^6. Although the higher α_{max} and the lower α_{min}, the better physical results are obtanied, the solution of the finite elements system of equation can be affected with the ill-conditioned matrices (scaled by these extreme coefficients).

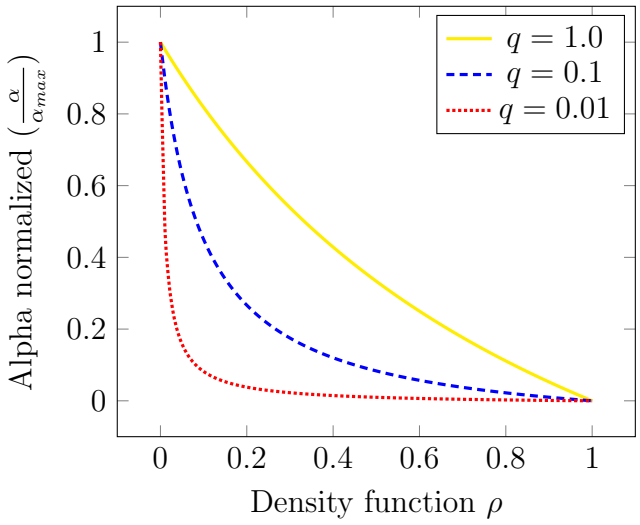


Figure 1.1: Inverse permeability function for different penalization values

1.2 Optimization Process

The optimization process begins with the definition of an objective function $\Phi(\boldsymbol{u}, \rho)$. This function is usually composed by a pair of integrals, one over the domain and the other over the boundaries (see Eq 1.4).

$$\Phi(\boldsymbol{u}, \rho) = \int_{\Omega} F_{\Omega}(\boldsymbol{u}, \rho) \mathrm{d}\Omega + \int_{\Gamma} F_{\Gamma}(\boldsymbol{u}, \rho) \mathrm{d}\Gamma \tag{1.4}$$

According to the optimization objectives, the functions F_{Ω} and F_{Γ} are defined. In order to reduce the "hydraulic losses" of a Stokes system, the power dissipation of the fluid is calculated. For other objectives, different functions can be defined with the form of Eq 1.4. In this case the functions used are:

$$F_{\Omega} = \frac{1}{2}\mu \sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)^2 + \sum_i \alpha(\rho) u_i^2$$
$$F_{\Gamma} = 0 \tag{1.5}$$

The complete optimization problem to be worked on this case can be stated as follows.

$$\min_{\rho} \ \Phi(u_i(\rho), \rho) = \int_{\Omega} \left[\frac{1}{2}\mu \sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)^2 + \sum_i \alpha(\rho) u_i^2 \right] \mathrm{d}\Omega$$

Subject to:

$$\begin{aligned}
&-\mu \Delta u_i + \alpha(\rho) u_i + \nabla p = f_i && \text{in } \Omega \\
&\nabla \cdot u_i = 0 && \text{in } \Omega \\
&u_i = u_D && \text{on } \partial\Omega \\
&\int_{\Omega} \rho \mathrm{d}\Omega \leq \beta \int_{\Omega} 1 \mathrm{d}\Omega \\
&0 \leq \rho \leq 1 && \text{in } \Omega
\end{aligned} \tag{1.6}$$

Where $0 \leq \beta \leq 1$ is the maximum portion of the domain that is permitted to be fluid. Following this, the coefficient $(1-\beta)$ determines the minimum fraction of the entire domain to set as solid. Having defined the objective function, the iterative optimization can be divided on three main steps, an inital guess ρ_0 should be given:

1. Given ρ_k, calculate $\alpha(\rho_k)$ and solve the state equation, in this case is the Stokes flow to find $(\boldsymbol{u}_k, p_k)$.
2. Evaluate the objective function $\Phi(\boldsymbol{u}_k, \rho_k)$ and calculate the sensitivity $\mathrm{d}\Phi/\mathrm{d}\rho_k$.
3. Obtain a new guess ρ_{k+1} according to ρ_k and $\mathrm{d}\Phi/\mathrm{d}\rho_k$.

The solution of these three steps is treated on the following subsections.

1.2.1 Stokes equation

The state eqution is defined as Eq 1.2. It is solved numerically using the Finite Element Method with the FreeFem++ software. The variational formulation, after integrating by

parts is: Find $u \in V = H_0^1(\Omega)$ and $p \in Q = L_0^2(\Omega)$ s.t .

$$\begin{aligned} \int_\Omega \mu \nabla u \cdot \nabla v \; dx + \int_\Omega \alpha(\rho) uv \; dx - \int_\Omega \text{div}\,(v) p \; dx = \int_\Omega fv \; dx - \int_{\partial\Omega} (\partial u / \partial n) \; v \; ds \quad \forall v \in V \\ \int_\Omega \text{div}\,(u) q \; dx = 0 \quad \forall q \in Q \\ u(x) = u_d \quad \text{for} \quad x \in \partial\Omega \end{aligned} \tag{1.7}$$

This variational formulation is solved having supplied a scalar field $\alpha(\rho)$.

1.2.2 Sensitivity analisys

Once the values of $(\boldsymbol{u}_k, p_k)$ are known (where k is the iteration number), the objective function value is evaluated by replacing these values on Eq 1.6. To compute the objective function sensitivity $(\mathrm{d}\Phi/\mathrm{d}\rho_i)$, the adjoint method is used.

Defining the state equation (Stokes) residual as R, the optimization problem (Eq. 1.6) can be written as:

$$\begin{aligned} &\min_{\rho} \;\; \Phi(u(\rho), \rho) \\ &\text{Subject to:} \\ &\qquad R(u, \rho) = 0 \\ &\qquad \int_\Omega \rho \mathrm{d}\Omega \leq \beta \int_\Omega 1 \mathrm{d}\Omega \\ &\qquad 0 \leq \rho \leq 1 \end{aligned}$$

The Lagrange functional is built using the cost function and the state equation residual accompanied by an adjoint variable (l).

$$\mathcal{L} = \Phi + lR \tag{1.8}$$

As the residual is defined as null $(R = 0)$ the Lagrange functional is equal to the cost function $(\mathcal{L} = \Phi)$.

The sensitivity analisys is done over the functional $\mathcal{L}$, obtaining:

$$\frac{d\mathcal{L}}{d\rho} = \frac{\partial \Phi}{\partial \rho} + l\frac{\partial R}{\partial \rho} + \left(\frac{\partial \Phi}{\partial u} + l\frac{\partial R}{\partial u}\right)\frac{du}{d\rho} \tag{1.9}$$

The last term of Eq 1.9 can be null if:

$$\frac{\partial R}{\partial u} l = -\frac{\partial \Phi}{\partial u} \tag{1.10}$$

Finally the cost function derivative with respect to the design variable is defined as:

$$\frac{d\Phi}{d\rho} = \frac{d\mathcal{L}}{d\rho} = \frac{\partial \Phi}{\partial \rho} + l\frac{\partial R}{\partial \rho} \tag{1.11}$$

Where l is obtained by solving equation 1.10.

The partial derivatives of Φ and R with respect to the variable u are calculated in the discrete form as done by Olesen et al. (2006).

1.2.3 Design parameters update

As proposed by Borrvall and Petersson (2003) and Olesen et al. (2006), the Method of Moving Asymptotes (MMA) developed by Svanberg (1987) is implemented in order to update the design variable ρ. Although this method involves solving a secondary optimization problem after doing the sensitivity analisys, the convergence results given make it an attractive tool to use.

The MMA guarantees that every step of the design parameter update is convergent and meaningful even if the optimization constraints are on the limits. This is achived by means of relaxing the constraints limits when the convergence is slow. An implementation of this method on FreeFem++ is done based on Svanberg (1987).

1.3 Implementation

The algorithm implementation was done completely in FreeFem++ and the PARAVIEW visualization software (Henderson (2007)) was used for postprocessing the results. 2D examples proposed on earlier publications of this method such as Borrvall and Petersson (2003) and Olesen et al. (2006) were used for the validation.

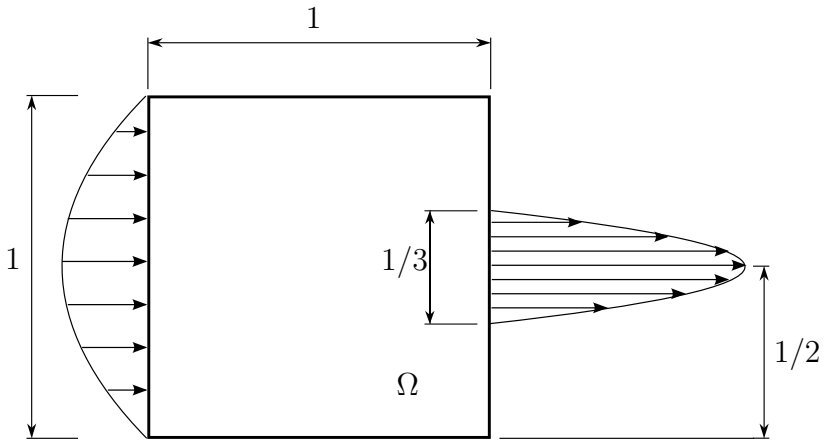


Figure 1.2: Diffuser conditions

1.3.1 Diffuser

The first implementation is the diffuser optimization. This example consists of a square domain with different inlet and outlet areas. Using the objective function defined on Eq.1.5, an optimized transition from inlet to outlet is obtained in order to reduce the fluid losses. The boundary conditions are shown on figure 1.2, the fluid volume constraint is set as $\beta = 0.5$ and the initial density is set as $\rho = 0.5$ constant on the entire domain. Both parabolic profiles are defined by the equation 1.12, where u_{max} is the maximum velocity, y is the vertical coordinate, c is the vertical coordinate of the boundary center (where the velocity is maximum) and l is the boundary length. In order to preserve mass, the maximum velocity in the parabolic inlet profile is set as $u_{max} = 1$ and $u_{max} = 3$ in the outlet.

$$u = u_{max}\left(1 - \left(\frac{2(y-c)}{l}\right)^2\right) \tag{1.12}$$

Figure 1.3 shows the obtained results after converging on 15 iterations using a 180x180 mesh and a constant penalization for the inverse permeability of $q = 0.1$. This result is consistent with the one obtained in Borrvall and Petersson (2003).

Figure 1.3: Diffuser density (ρ) solution

1.3.2 Double pipe

In this case two inlets and two outlets are defined on a rectangular domain with variable length as shown on figure 1.4. The maximum fluid portion is set as $\beta = \frac{1}{3}$, and the maximum velocity on the 4 parabolic profiles is set as $u_{max} = 1$ using equation 1.12 as the above example. The initial density is set as $\rho = \frac{1}{3}$ constant on the whole domain.

Two penalization parameters q for $\alpha(\rho)$ are used following Borrvall and Petersson (Borrvall and Petersson (2003)), in order to reach a global minimum. A first optimization process is done using $q = 0.01$ and this solution is used as the initial guess of a new process with $q = 0.1$. The domain length values simulated are $\delta = 1$ and $\delta = 1.5$ with 100x100 and 150x100 discretizations respectively. Figure 1.5 shows the obtained results.

1.4 Mesh adaptation

The current optimization process requires a refined mesh in order to obtain a good definition of the "boundaries" between the solid and fluid domains (change of density value). It is seen on Borrvall and Petersson (2003) that the solution does not depends on the mesh size, a coarse mesh gives a similar solution but with a diffuse boundary. Therefore, it is proposed a two step optimization in order to obtain better computing times.

The proposed steps are:

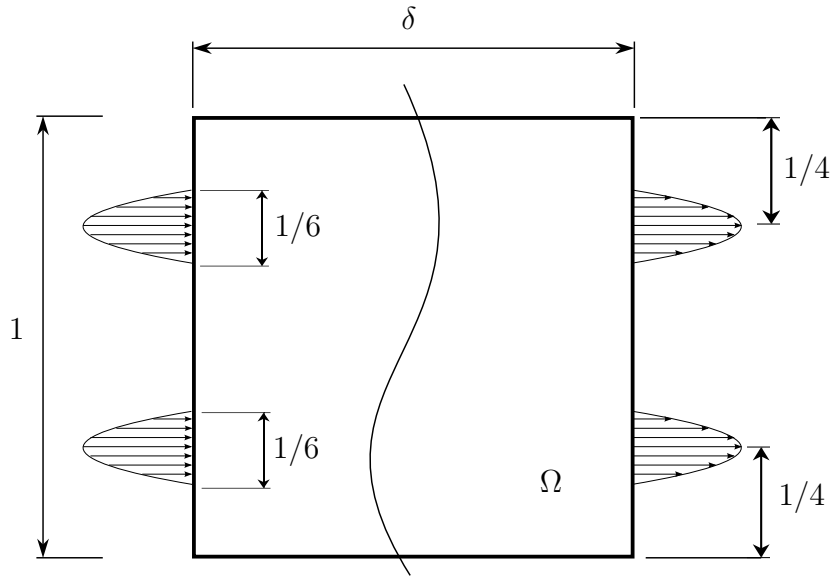


Figure 1.4: Double pipe conditions

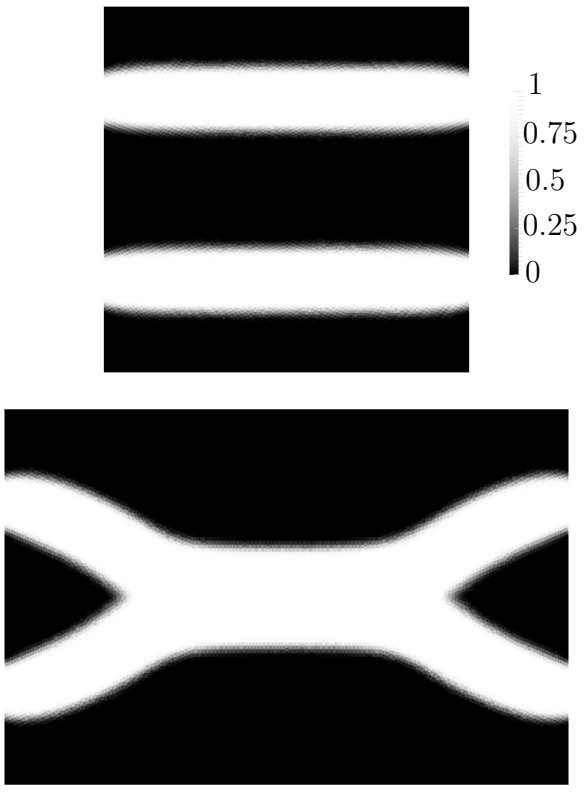


Figure 1.5: Double pipe density (ρ) solution for $\delta = 1$ and $\delta = 1.5$

1. Solve the optimization process with a rough mesh and obtain a first density approximation.

2. Based on the obtained result refine the mesh in order to have a higher element density on the interfase between the fluid ($\rho = 1$) and solid ($\rho = 0$).

3. Solve new optimization case using the refined mesh and the density function obtained on step 1 as an initial guess.

The mesh adapatation is based on the definition of the element size. The objective is to have a refined mesh on the boundary between the solid and fluid domains, and larger elements into both regions. In addition, the element size ratio is restricted in order to have a smooth size variation on the mesh. That is,

$$s(\rho) = \begin{cases} s_{max}, & \text{if } \rho = 1 \\ s_{max}, & \text{if } \rho = 0 \\ s_{min}, & \text{elsewhere} \end{cases} \tag{1.13}$$

The method is implemented and tested on the diffuser benchmark. Figure 1.6 shows the process results and 1.7 the objective function evolution over the iterations. The objective function over the computation time is shown of figure 1.8.

Although the two step process requires more iterations to converge, the computational efficiency is remarkable. The mesh optimization by refination on the desired regions is a great advantage to benefit form every degree of freedom. A reduction of the 95% of the computational time is achieved.

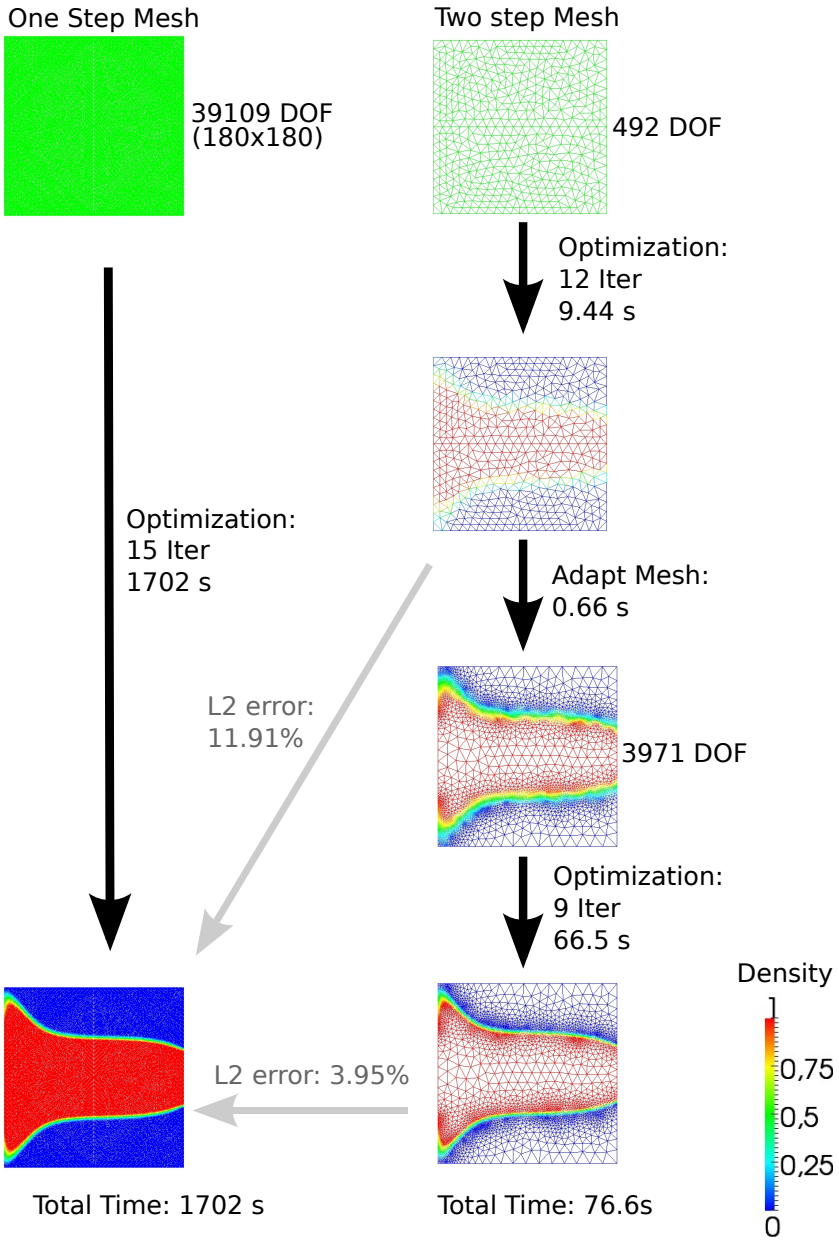


Figure 1.6: Mesh adaptation results compared with a single refined procedure

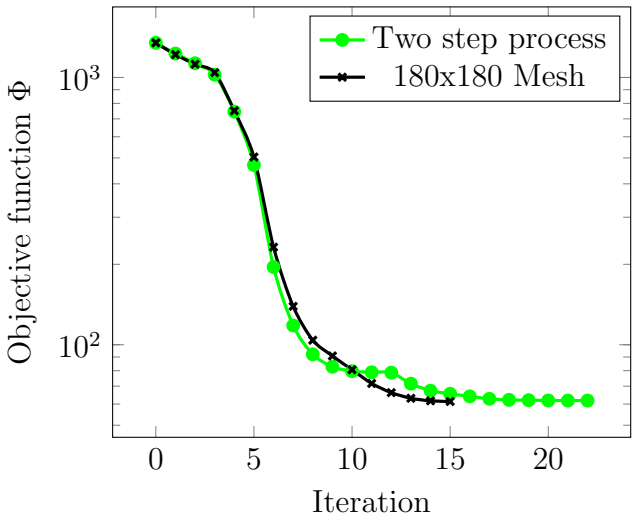


Figure 1.7: Objective function over optimization iterations

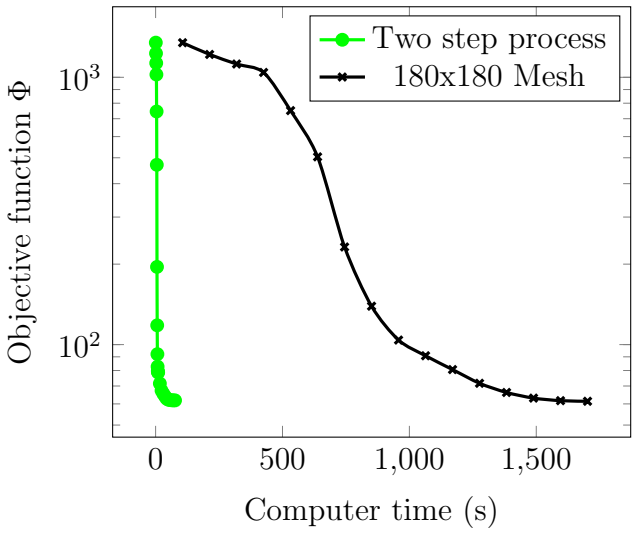


Figure 1.8: Objective function over computation time

Chapter 2

CFD shape optimization

The shape optimization using a derivative of the objective function with respect to the domain is a widely used method on solid mechanics. Applications of direct shape derivatives to fluids can be seen for instance in Mohammadi and Pironneau (2010). The main idea is to evolve the domain boundary according to a sensitivity analysis. Although the domain deformation can require remeshing and topology changes can cause problems on the boundary evolution, the explicit definiton of the domain boundary is a great advantage.

The shape derivative of the Stokes energy dissipation is obtained using a Lagrangian formulation. A finite elements optimization is implemented on FreeFem++ and validation examples are executed.

2.1 Optimization Problem

Following last chapter, the objective function used is the power dissipation of the flow. In contrast to chapter 1, no density function ρ is not needed and the classic Stokes is the state equation. The complete optimization can be defined as follows:

$$\min_{\Omega} \; J(\Omega, u_i(\Omega)) = \int_{\Omega} \mu \nabla u_i : e(u_i) \mathrm{d}\Omega$$

Subject to:

$$\begin{aligned} R_i(\Omega, u_i, p): \quad & -\mu \Delta u_i + \nabla p - f_i = 0 && \text{in } \Omega \\ C(\Omega, u_i): \quad & \operatorname{div}(u_i) = 0 && \text{in } \Omega \\ & u_i = u_i^D && \text{on } \Gamma^D \in \partial\Omega \\ & u_i = 0 && \text{on } \partial\Omega \backslash \Gamma^D \in \partial\Omega \end{aligned} \tag{2.1}$$

Where $e(u_i) = \nabla u_i + (\nabla u_i)^T$.

2.2 Shape sensitivity analysis

The optimization process is based on the variation of the domain boundaries to evolve to an optimum shape. This method is based on the structural shape optimization done by Allaire (2006) on Chapter 6 and Dapogny (2013) on chapter 2.

2.2.1 Hadamard's boundary definition

According to Hadamard (1909), any variation on the domain shape (without changing the topology) can be defined with a displacement field θ and the initial domain Ω_0 (See figure 2.1). Assuming θ sufficiently small, a deformed domain Ω is represented by $\Omega = (I + \theta)(\Omega_0)$. The shape variations $(I + \theta)$ are considered homeomorphisms close to the identity.

Mathematical details about this method can be found on Allaire (2006).

2.2.2 Shape derivative

The Cea's method, developed by Céa (1986) is proposed to obtain the shape derivative. It is based on the formulation of the Lagrange operator and the solution of the adjoint state equation to find the objective function derivative.

The first step is the Lagrangian definition of the optimization problem, here the state and continuity equations are incorporated as a constraint using lagrange multipliers (v_i, q).

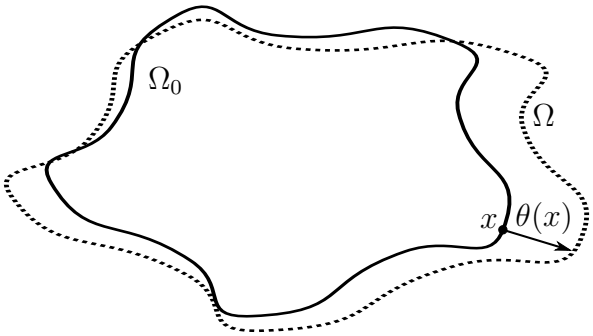


Figure 2.1: Hadamard's shape variation

The variables (u_i, p, v_i, q) must not depend on the domain. For the Dirichlet boundary conditions, a change of variable must be done according to Allaire (2006) on sections 6.4.2 and 6.4.3. A "lifting" function u_i^l is defined as Eq 2.2 and a new variable $\tilde{u_i}$ that does not depend on the Dirichlet boundary is defined as Eq 2.3.

$$u_i^l = \begin{cases} u_i^D & \text{on } \Gamma^D \\ 0 & \text{on } \partial\Omega \backslash \Gamma^D \end{cases} \tag{2.2}$$

$$\tilde{u_i} = u_i - u_i^l \tag{2.3}$$

The variables (u_i^Ω, p^Ω) are the solution of the state equations $R_i = 0$ and $C = 0$, if and only if $(\tilde{u_i}^\Omega, p^\Omega)$ are the solution of the "lifted" system defined by:

$$\begin{aligned} \tilde{R}_i(\Omega, \tilde{u_i}, p): \quad -\mu\Delta\tilde{u_i} + \nabla p - f_i - \mu\Delta u_i^l &= 0 && \text{in } \Omega \\ \tilde{C}(\Omega, \tilde{u_i}): \quad \operatorname{div}(\tilde{u_i} + u_i^l) &= 0 && \text{in } \Omega \\ \tilde{u_i} &= 0 && \text{on } \Gamma^D \in \partial\Omega \\ u_i &= 0 && \text{on } \partial\Omega \backslash \Gamma^D \in \partial\Omega \end{aligned} \tag{2.4}$$

Now, the lagrangian is defined:

$$\mathcal{L}(\Omega, \tilde{u_i}, p, v_i, q) = J(\Omega, \tilde{u_i}) + \int_\Omega v_i \tilde{R}_i(\Omega, \tilde{u_i}, p) \mathrm{d}\Omega + \int_\Omega q\tilde{C}(\Omega, \tilde{u_i}) \mathrm{d}\Omega \tag{2.5}$$

Using the first Green identity, the last equation is transformed into:

$$\mathcal{L}(\Omega, \tilde{u_i}, p, v_i, q) = \int_\Omega \mu \nabla(\tilde{u_i}) : e(\tilde{u_i}) \mathrm{d}\Omega + \int_\Omega \mu \nabla \tilde{u_i} : \nabla v_i \ \mathrm{d}\Omega - \int_\Omega \mathrm{div}\,(v_i) p \ \mathrm{d}\Omega - \int_\Omega f_i v_i \ \mathrm{d}\Omega + \int_\Omega \mu \nabla u_i^l : \nabla v_i \ \mathrm{d}\Omega + \int_{\partial\Omega} (\partial \tilde{u_i}/\partial n) \ v_i \ ds + \int_\Omega \mathrm{div}\,(\tilde{u_i}) q \ \mathrm{d}\Omega + \int_\Omega \mathrm{div}\,(u_i^l) q \ \mathrm{d}\Omega \tag{2.6}$$

The variables $\tilde{v}_i$ and q appear as lagrangian multipliers and act as a test function to obtain the variational formulation of the state equations $\tilde{R}_i$ and $\tilde{C}$ into $\mathcal{L}$ as shown on equation 2.6. Then, the last two terms of Eq 2.5 define the weak form of the state equation. By solving the Lagrangian, and finding $(\tilde{u_i}^\Omega, p^\Omega)$ as the solution of the state eqution (Eq 2.4), one can define the objective function as:

$$J(\Omega) = \mathcal{L}(\Omega, \tilde{u_i}^\Omega, p^\Omega, v_i, q) \qquad \forall v_i, q \tag{2.7}$$

Then, the shape derivative of the function $\mathcal{L}$ is obtained. This derivative is the sensitivity of the function $\mathcal{L}$ with respect to a domain variation θ as defined in subsection 2.2.1, it reads $\mathcal{L}'(\Omega, ...)(\theta)$.

$$\begin{aligned}\mathcal{L}'(\Omega, \tilde{u_i}, p, v_i, q)(\theta) = &\frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, \tilde{u_i}, p, v_i, q)(\theta) + \\ &\frac{\partial \mathcal{L}}{\partial \tilde{u_i}}(\Omega, \tilde{u_i}, p, v_i, q)(\dot{u_i}(\theta)) + \\ &\frac{\partial \mathcal{L}}{\partial p}(\Omega, \tilde{u_i}, p, v_i, q)(\dot{p}(\theta)) + \\ &\frac{\partial \mathcal{L}}{\partial v_i}(\Omega, \tilde{u_i}, p, v_i, q)(\dot{v_i}(\theta)) + \\ &\frac{\partial \mathcal{L}}{\partial q}(\Omega, \tilde{u_i}, p, v_i, q)(\dot{q}(\theta))\end{aligned} \tag{2.8}$$

This result (Eq 2.8) is obtained taking advantage of the non dependancy of the variables $(\tilde{u_i}, p, v_i, q)$ on the domain. The partial derivatives with respect to these variables that are different from the domain are calculated:

$$\frac{\partial \mathcal{L}}{\partial \tilde{u_i}}(\Omega, \tilde{u_i}, p, v_i, q)(\hat{u_i}) = J'(\tilde{u_i}, \hat{u_i}) + \int_\Omega \mu \nabla \hat{u_i} \cdot \nabla v_i \, \mathrm{d}\Omega + \int_{\partial\Omega} (\partial \hat{u_i}/\partial n) \; v \; ds - \int_\Omega \mathrm{div}\,(\hat{u_i}) q \, \mathrm{d}\Omega \tag{2.9}$$

$$\frac{\partial \mathcal{L}}{\partial p}(\Omega, \tilde{u_i}, p, v_i, q)(\hat{p}) = -\int_\Omega \mathrm{div}\,(v_i)\hat{p} \, \mathrm{d}\Omega \tag{2.10}$$

$$\frac{\partial \mathcal{L}}{\partial v_i}(\Omega, \tilde{u_i}, p, v_i, q)(\hat{v_i}) = \int_\Omega \mu \nabla \tilde{u_i} \cdot \nabla \hat{v_i} \, \mathrm{d}\Omega - \int_\Omega \mathrm{div}\,(\hat{v_i}) p \, \mathrm{d}\Omega - \int_\Omega f \hat{v_i} \, \mathrm{d}\Omega + \int_{\partial\Omega} (\partial \tilde{u_i}/\partial n) \; \hat{v_i} \; ds \tag{2.11}$$

$$\frac{\partial \mathcal{L}}{\partial q}(\Omega, \tilde{u_i}, p, v_i, q)(\hat{q}) = \int_\Omega \mathrm{div}\,(\tilde{u_i}) \hat{q} \, \mathrm{d}\Omega \tag{2.12}$$

The objective function derivative ($J'(\tilde{u_i})$) is calculated obtaining:

$$J'(\tilde{u_i}, \hat{u_i}) = \int_\Omega \mu \nabla \hat{u_i} : e(\tilde{u_i}) \, \mathrm{d}\Omega + \int_\Omega \mu \nabla \tilde{u_i} : e(\hat{u_i}) \, \mathrm{d}\Omega \tag{2.13}$$

The last terms (Eqs 2.9 to 2.12) can be null and ignored from Eq 2.8. It is clear that the term defined by making zero Eq 2.11 is the same weak form of the state equation and Eq. 2.12 its continuity condition. In this case, one solves for finding $(\tilde{u_i}^\Omega, p^\Omega)$ s.t 2.11 and 2.12 go to cero for every $(\hat{v_i}, \hat{q})$.

Likewise, the same is done for equations 2.9 and 2.10. One can see on 2.9 the same state equation with a source term defined by the objective function derivative j'. On this case, the results of the state equation $(\tilde{u_i}^\Omega, p^\Omega)$ are used to compute the source term and the unknowns (v_i^Ω, q^Ω) are found. This is the adjoint state system.

Finally, after making all these terms null, the shape derivative reads:

$$\mathcal{L}'(\Omega, \tilde{u_i}, p, v_i, q)(\theta) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, \tilde{u_i}^\Omega, p^\Omega, v_i^\Omega, q^\Omega)(\theta) \tag{2.14}$$

Where the partial derivative $\frac{\partial \mathcal{L}}{\partial \Omega}$ is calculated according to the structure theorem (See Delfour and Zolésio (2011) theorem 9.3.6 or Dapogny (2013) theorem 2.2.2).

$$\frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, \tilde{u_i}^{\Omega}, p^{\Omega}, v_i^{\Omega}, q^{\Omega})(\theta) =$$
$$\int_{\partial\Omega_0} \theta \cdot \hat{n}_i(\mu \nabla \tilde{u_i}^{\Omega} : e(\tilde{u_i}^{\Omega}) + \mu \nabla \tilde{u_i}^{\Omega} : \nabla v_i^{\Omega} - \operatorname{div}(v_i^{\Omega}) p^{\Omega} - f_i v_i^{\Omega}$$
$$+\mu \nabla u_i^l : \nabla v_i^{\Omega} + \operatorname{div}(\tilde{u_i}^{\Omega}) q^{\Omega} + \operatorname{div}(u_i^l) q) \mathrm{d}s \tag{2.15}$$

Where $\hat{n}_i$ is the unitary vector that defines the normal direction of $\partial\Omega_0$. By continuity condition: $\operatorname{div}(\tilde{u_i}^{\Omega}) = -\operatorname{div}(u_i^l)$ and $\operatorname{div}(v_i^{\Omega}) = 0$, then if the body forces are null ($f_i = 0$), the shape derivative of the objective function defined on equation 2.1 reads:

$$J'(\Omega) = \frac{\partial \mathcal{L}}{\partial \Omega}(\Omega, \tilde{u_i}^{\Omega}, p^{\Omega}, v_i^{\Omega}, q^{\Omega})(\theta) =$$
$$\int_{\partial\Omega_0} \theta \cdot \hat{n}_i \left(\mu \nabla \tilde{u_i}^{\Omega} : e(\tilde{u_i}^{\Omega}) + \mu \nabla \tilde{u_i}^{\Omega} : \nabla v_i^{\Omega} + \mu \nabla u_i^l : \nabla v_i^{\Omega}\right) \mathrm{d}s \tag{2.16}$$

The term v_i^{Ω} is the solution to the adjoint system defined by equation 2.9 and 2.10 as continuity condition. The shape derivative can be defined as the sensitivity of the objective function $J(\Omega)$ with respect to a variation of the domain shape defined by the boundary displacement field of magnitude θ on the normal direction $\hat{n}_i$.

2.3 Domain variation

The shape sensitivity gives the direction to deform the domain in order to change the objective function in a desired way. As defined on equation 2.15, if $\theta \cdot \hat{n}_i = 1$ the variation of the objective function would have the magnitude of the term in brackets. This way a displacement field $m_i^{\partial\Omega} = \theta \hat{n}_i$ over the boundary is found in order to minimize significantly the cost function. This field is known on the entire boundary, has normal direction and variable magnitude according to the sensitivity.

Now the purpose is to move the boundary according to this field $m_i^{\partial\Omega}$. To avoid remeshing, a continuous displacement field over the entire domain m_i^{Ω} is found and the complete mesh is displaced according to it. To determine the field m_i^{Ω} by knowing its value on the boundary ($m_i^{\partial\Omega}$), a displacement based linear elasticity problem is solved:

$$
\begin{aligned}
-\operatorname{div}\left(\mu(\nabla m_i + (\nabla m_i)^T) + \lambda(\operatorname{div}(m_i))I\right) &= f_i \qquad \text{in } \Omega \\
f_i &= 0 \qquad \text{in } \Omega \\
m_i &= m_i^{\partial\Omega} \qquad \text{on } \partial\Omega
\end{aligned}
\tag{2.17}
$$

Where μ and λ are the Lamé constants. The domain is updated by moving the complete mesh according to the solution of equation 2.17 ($m_i^{\Omega} = m_i$).

2.4 Implementation

The implementation of the method is done in FreeFem++ taking advantage of functions such as `movemesh` to make the domain variation. The visualization is done in Paraview, an open source visualization system, via exporting files with VTK format from FreeFem++. The diffuser example was used to test the program.

2.4.1 Diffuser

The validation of the method is done with the diffuser example as defined in section 1.3.1 but with different inital conditions. The inital shape and boundary conditions can be seen on figure 2.2. The maximum volume is 0.8, then the optimization iterations should stop when it is reached. The maximum movement is defined as $5x10^{-4}$ and only the walls are allowed to move, the inlet and the outlet should remain the same.

The initial, intermediate and final state of the optimization process are displayed in figure 2.3. The program performed 248 iterations and stopped when the volume reached the maximum, 0.8 in this case. Figures 2.4 and 2.5 show the objective function value and the domain volume respectively over the iterations.

A comparison of this optimization method and the SIMP treated in chapter 1 is done with this example. Figure 2.6 shows the results of both methods. Similar shapes are obtained and the objective function values are $\Phi = 26,49$ for the SIMP method and $\Phi = 27,32$ for the shape derivative method.

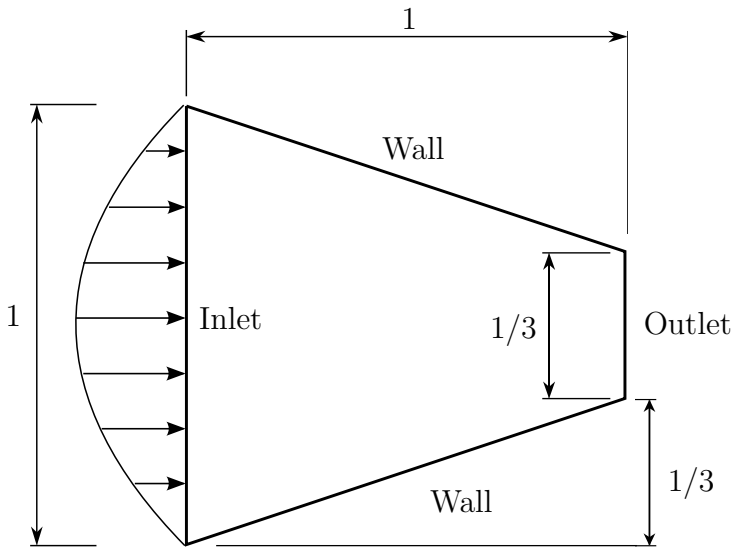
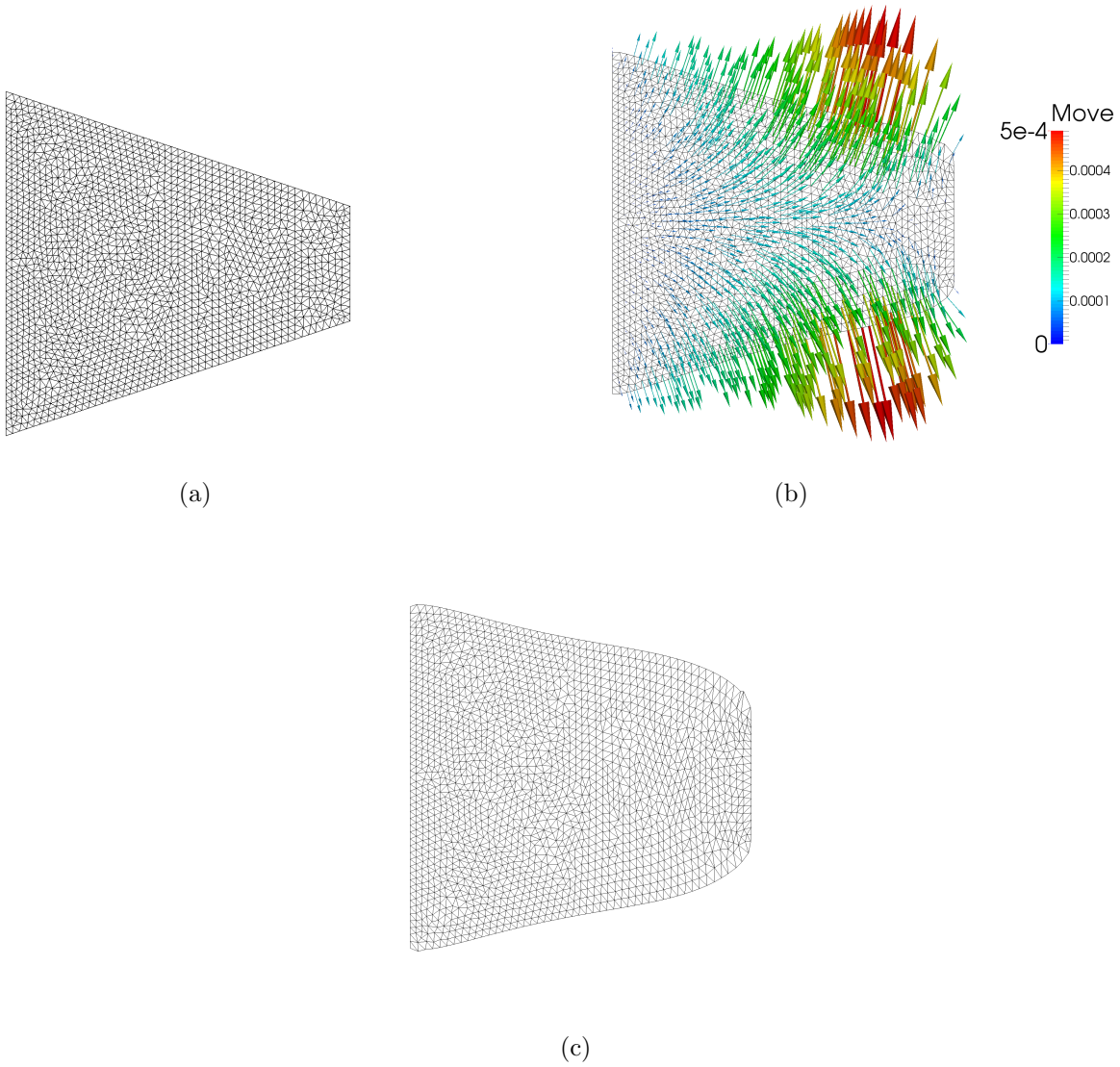


Figure 2.2: Diffuser conditions



(a) (b)

(c)

Figure 2.3: Diffuser initial shape (a), intermediate position with displacement field (b) and optimized final shape (c).

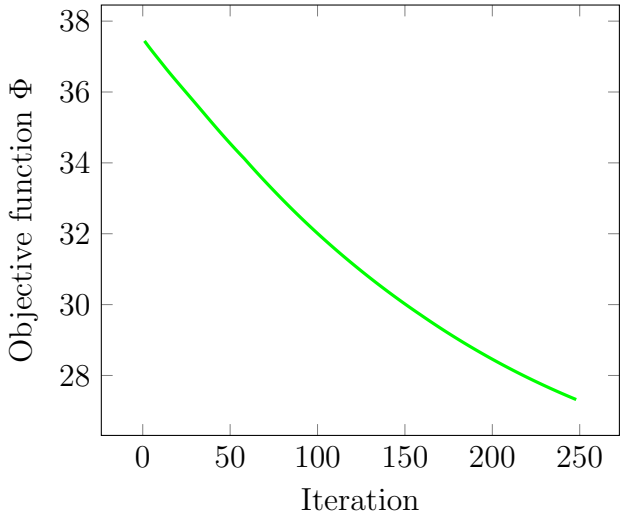


Figure 2.4: Objective function over iterations

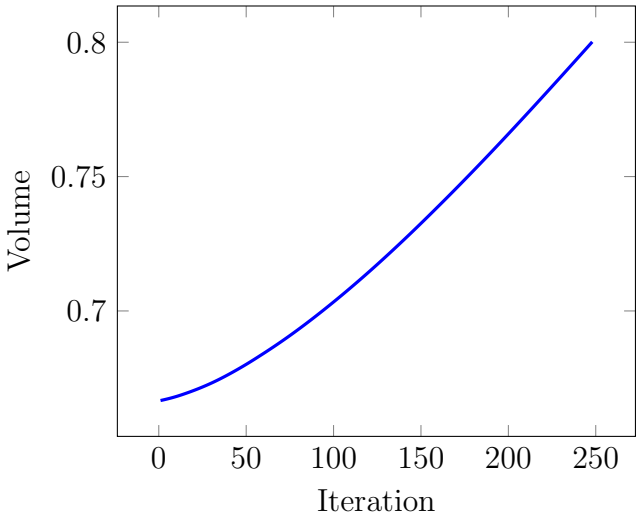


Figure 2.5: Domain volume over iterations

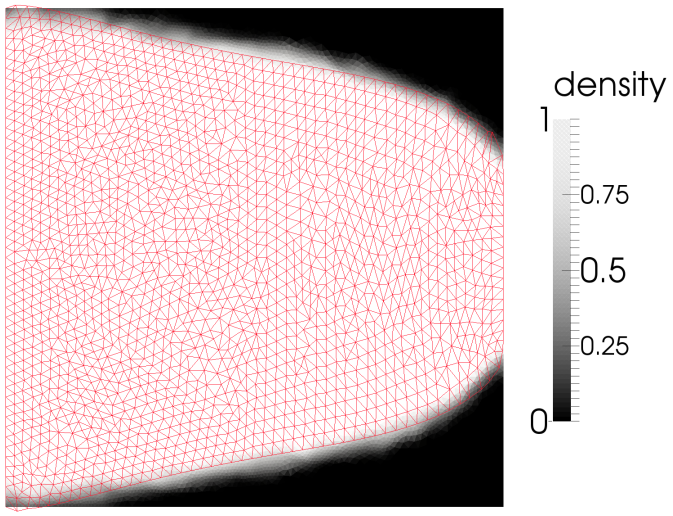


Figure 2.6: Comparison of the optimization results using SIMP method (black and white density function) and Shape optimization (red mesh).

Chapter 3

CFD two step optimization

To this chapter, two different CFD optimization methods have been studied. On one hand the topology optimization method using SIMP with the fluid domain defined implicitly by a density function ρ (Chapter 1). On the other hand a shape optimization method based on the derivative with respect to the domain (shape derivative) where only the fluid domain is taken into account and its boundary evolves to an optimum shape (Chapter 2). The purpose of this chapter is to combine this two methods to solve a problem by making a two step process, first to find an initial shape using SIMP, and then extract the fluid domain to apply the shape optimization process.

The use of these two methods for the same purpose has been studied by Pantz and Trabelsi (2007) for solids and Othmer (2008) and Kongress et al. (2006) for fluids. On both fluids works, topology and shape optimization methods are studied but they are not applied on the same optimization example. In this chapter, a complete two step process involving topology and shape derivative is presented.

3.1 Optimization process

The process can be rougly divided in three steps:

1. Solve the topology optimization process as done in section 1.2.
2. Extract from the resulting density function the fluid domain.
3. Modify the fluid boundaries using shape optimization (see section 2.3).

3.2 Fluid domain obtainment

To couple the topology with the shape optimization process, the fluid domain must be taken out from the density function distribution. This process will give a ragged boundary for the domain and the shape optimization process requires a sufficiently smooth boundary.

To take out the fluid domain, a isoline with $\rho = 0.5$ is created over the SIMP density solution, and then used as border of the new domain. The software used (FreeFem++) provides a function called `isoline` to create this contour from the density function, moreover this function can smooth the isoline to obtain a more continuous boundary. Then the fluid domain is discretized and used as input for the second step of optimization.

3.3 Implementation

To make the two step implementation a main function on FreeFem++ is created, the objective function is the power dissipation of the fluid and the state equation is defined by incompressible Stokes flow. The optimization problem is defined by equation 1.6 for topology oprimization and by equation 2.1 for shape optimization. It can be proved that both objective functions are equal when $\alpha \approx 0$, leading to the same optimization problem.

The first step is to define the space of design, then discretize it and run the topology optimization process. From the topology optimization, a scalar function called density is obtained. From this density function, a new domain is created according to section 3.2. This new domain is used as the intial state for the shape optimization process, the one proposed on chapter 2. Having the volume as a constraint and stopping criteria for the shape optimization process, brings the need for a clearance between the inital and maximum volume to perform the optimization. Then, the volume constraint should be slightly lower for the first step than for the second one.

3.3.1 Diffuser

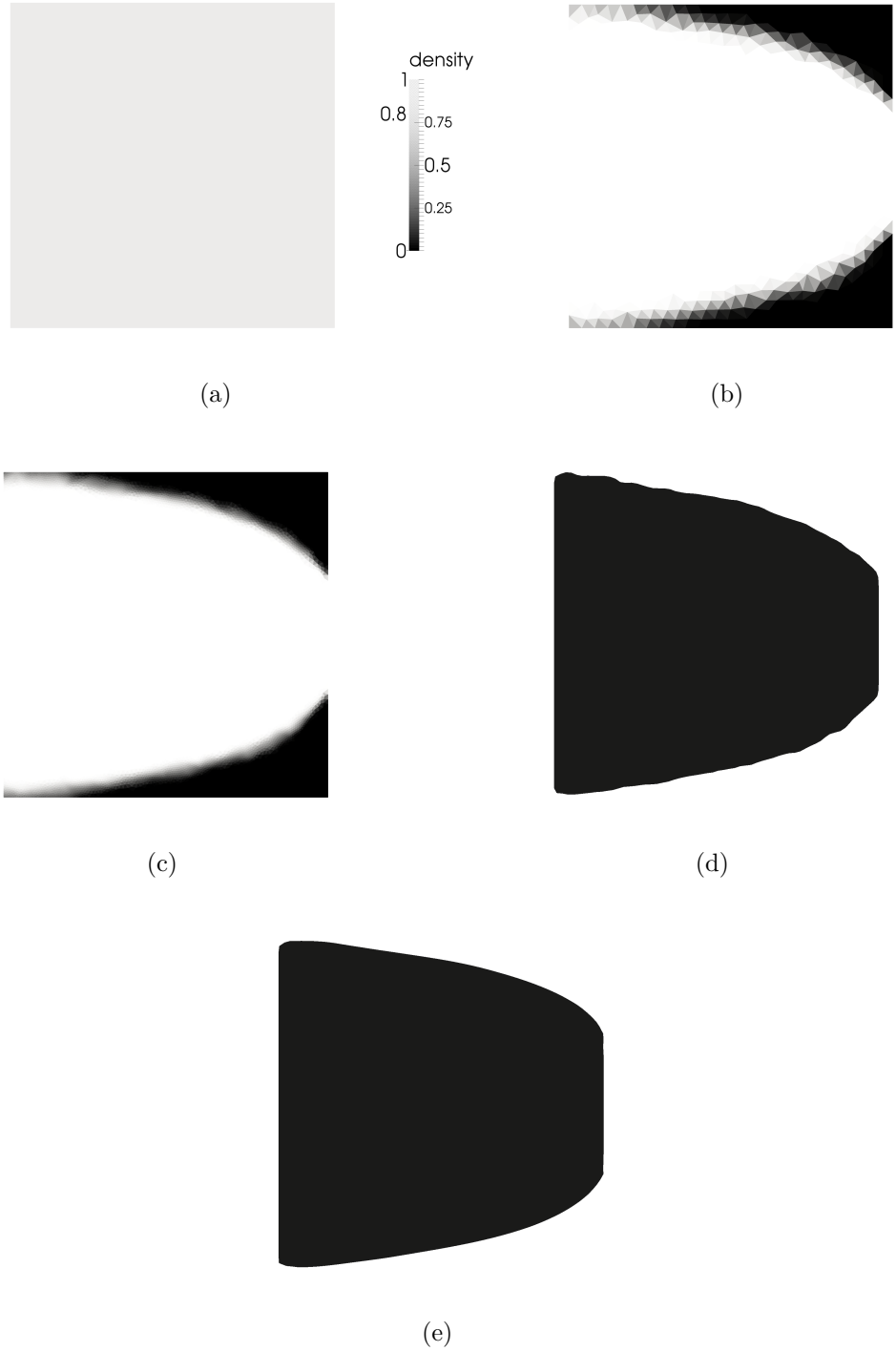
The optimization of the diffuser with the inital conditions defined in figure 1.2 is done. The volume constraint for the SIMP step is defined as 0.8 and for the second step is increased to 0.85. A mesh adaptation step is implemented on the SIMP process as defined on section 1.4. The process is summarized in table 3.1 and the results in figure 3.1. The objective

Iteration	Event	Objective function value	Fluid domain volume
0	SIMP optimization with rough mesh start.	389.4	0.8
8	SIMP optimization with rough mesh end.	26.9	0.8
9	SIMP optimization with adapted mesh start.	26.5	0.8
11	SIMP optimization with adapted mesh end.	26.3	0.8
12	Shape optimization start.	24.6	0.8
96	Shape optimization end.	23.7	0.85

Table 3.1: Two step optimization process of the diffuser.

function evolution over iterations is plotted in figure 3.2 where the gray line marks the change of optimization step.

It can be seen that the objective function is always decreasing. The shape optimization step does not reduce significantly the objective function but produces the smooth boundary as desired. The global performance of the process is satisfactory.



(a) (b)

(c) (d)

(e)

Figure 3.1: Diffuser initial state for SIMP optimization (a), final state for SIMP with rough mesh (b), final state for SIMP with adapted mesh (c), inital fluid domain for shape otimization obtained from SIMP density result (d) and shape optimization final result (e).

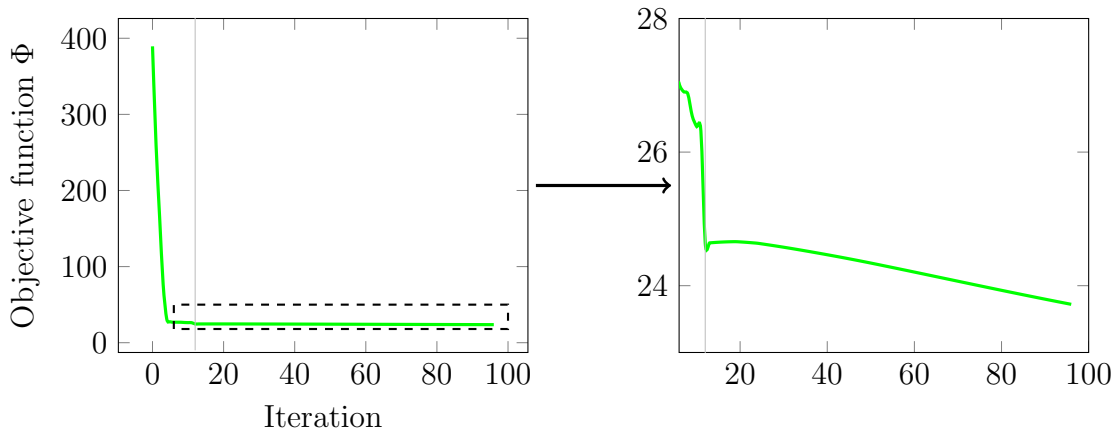


Figure 3.2: Objective function over iterations with zoom to the shape optimization step (from iteration 12).

Chapter 4

Conclusions and perspectives

The CFD optimization method proposed on this project shows satisfactory results. As inital step, the SIMP method is implemented with the purpose of producing an inital design. Obtaining the fluid domain from the resulting density function of the topology optimization will generate an undesired ragged surface. Therefore, the shape optimization method using a shape derivative of the objective function is used to correct the ribbed surface.

The topology optimization method presented in chapter 1 shows convergence in the iterative process. The inclussion of the MMA to update the density function is essential for the optimization process. The double pipe benchmark solution proves the capability of the procedure to reach different topologies. In search of efficiency, the mesh adaptation step is inserted in the process, remarkable results were obtained in terms of computational time to reach a desired solution.

Although it is very popular for solid optimization, the shape optimization using shape derivatives is still not used widely for CFD optimizations. The procedure of differentiating with respect to the domain is ilustrated in a generic form and implemented to reduce the Stokes flow energy dissipation (See chapter 2). The variation of a discretized domain boundary is not an easy task, to avoid remeshing on every iteration is the main objective of this implementation. The continuous displacement field found by solving a linear elasticity equation over the domain along with the `movemesh` function provided by the FreeFem++ software are the basis of the domain update. To keep a smooth boundary and converge to an optimum result, small boundary displacements should be defined, then a noticeable shape variation would require a large number of iterations.

The two step process involving topology and shape optimization is the core of this project (chapter 3). The coupling of these two methods with a isovalue domain obtainment worked as expected. an intial design is obtained using the SIMP method and the surface is smoothed sufficiently by the shape derivative based method while the objective function is still decreasing. A optimum design with explicit boundary definition is obtained.

As future work, a 3D Navier-Stokes optimization method is proposed. The non-linear term of this equation must be treated carefully in order to obtain valid sensitivities. The domain update with the shape derivative can be done taking advantage of the MMA as done on the SIMP method, this way local minima can be avoided and better convergence reached.

Bibliography

G. Allaire. *Conception optimale de structures: Majeure Sciences de l'ingénieur, simulation et modélisation*. École polytechnique, Département de Mathématiques appliquées, 2006. ISBN 9782730213042.

Thomas Borrvall and Joakim Petersson. Topology optimization using regularized intermediate density control. *Computer Methods in Applied Mechanics and Engineering*, 190(37–38):4911 – 4928, 2001.

Thomas Borrvall and Joakim Petersson. Topology optimization of fluids in stokes flow. *International Journal for Numerical Methods in Engineering*, 41:77–107, 2003.

Jean Céa. Conception optimale ou identification de formes: calcul rapide de la dérivée directionnelle de la fonction coût. *Modélisation mathématique et analyse numérique*, 20:371–402, 1986.

Charles Dapogny. *Shape optimization, level set methods on unstructured meshes and mesh evolution*. PhD thesis, Ecole Doctorale Paris Centre, 2013.

M. C. Delfour and J.-P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, 2nd ed.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2011.

Manuel J. Garcia and G. P. Steven. Fixed grid finite elements in elasticity problems. *Engineering Computations*, 16:145–164, 1999.

Jacques Hadamard. *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*. Paris : Imprimerie nationale., 1909.

F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012. ISSN 1570-2820.

Amy Henderson. *The ParaView Guide: A Parallel Visualization Application*. Kitware, 2007.

Khodor Khadra, Philippe Angot, Sacha Parneix, and Jean-Paul Caltagirone. Fictitious domain approach for numerical modelling of navier–stokes equations. *International Journal for Numerical Methods in Fluids*, 34(8):651–684, 2000.

Internationaler Kongress, Cfd topologie Und Formoptimierung Mit Adjungierten-methoden, Carsten Othmer, and Volkswagen Ag. Cfd topology and shape optimization with adjoint methods, 2006.

Kyungjun Lee. *Topology optimization of convective cooling system designs*. PhD thesis, University of Michigan, 2012.

B. Mohammadi and O. Pironneau. Applied optimal shape design. *Journal of Computational and Applied Mathematics*, 149(1):193 – 205, 2002. Scientific and Engineering Computations for the 21st Century - Methodologies and Applications Proceedings of the 15th Toyota Conference.

Bijan Mohammadi and Olivier Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press Inc., 2010.

Laurits Højgaard Olesen, Fridolin Okkels, and Henrik Bruus. A high-level programming-language implementation of topology optimization applied to steady-state navier–stokes flow. *International Journal for Numerical Methods in Engineering*, 65(7):975–1001, 2006.

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.

O. Pantz and K. Trabelsi. Simultaneous shape, topology, and homogenized properties optimization. *Structural and Multidisciplinary Optimization*, 34(4):361–365, 2007.

O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64: 97–110, 6 1974. ISSN 1469-7645.

Krister Svanberg. The method of moving asymptotes-a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987. ISSN 1097-0207.

Dominique Thvenin and Gbor Janiga. *Optimization and Computational Fluid Dynamics*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 3540721525, 9783540721529.