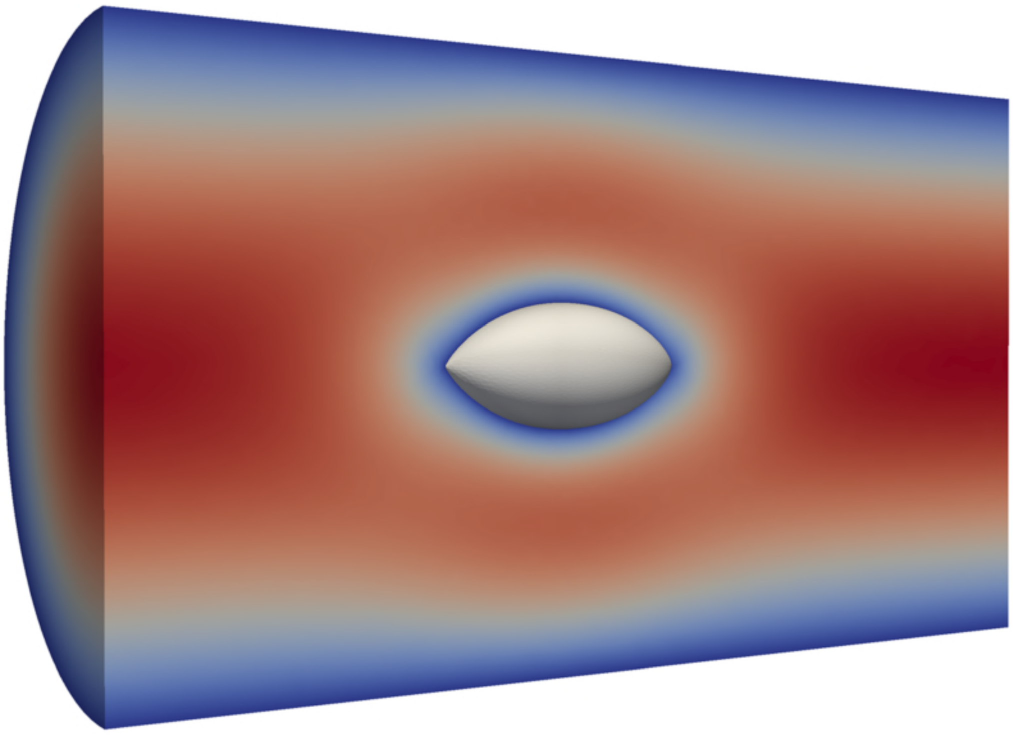
EPSRC Centre for Doctoral Training in

Industrially Focused Mathematical Modelling



Shape Optimization in Stokes Flow

Florian Wechsung

Contents

1. Introduction

Background

Aerodynamics play a crucial role in the design of modern cars, aeroplanes and many other objects subject to moving air and fluid flow. In particular, in the case of cars the minimization of drag forces is a key factor in order to make a vehicle that is as fuel efficient or as fast as possible.

The classic procedure when designing a car or aeroplane has been the following: an engineer uses his experience in aerodynamics and his intuition to design a prototype. This prototype is then tested in a wind tunnel and, based on the observations, the engineer changes his design - this process gets repeated until an optimal shape is believed to be found. Optimal means that the shape minimizes or maximizes a quantity or functional of interest subject to constraints; common examples are minimization of drag on an object or the maximization of lift or down force that is generated. With the improvements of Computational Fluid Dynamics (CFD), the prototypes in the wind tunnel have mostly been replaced by Computer-Aided design (CAD) and CFD simulations; however, the trial-and-error aspect often remains.

The goal of shape optimization is to automate the process of designing objects with ideal

Our goal, together with London Computational Solutions, is to use the theory of shape optimization to automate this procedure and to guarantee that the optimal shape is found by using techniques from PDE-constrained optimization and geometry.

Finite vs Infinite Dimensional optimization

One possible way to automate shape optimization is the parametrization of the object in a finite dimensional shape space, e.g. by a certain type of polynomials or splines. This enables us to use classic and well understood optimization routines in $\mathbb{R}^d$ to find the optimal coefficients for the parametrization of the shape. However, since every functional evaluation requires a costly solve of the governing PDE, using an optimization routine that uses not just functional values but also derivative information is crucial in order to reduce the number of iterations needed. This derivative can either be obtained via finite differencing, which is only feasible for very simple parametrizations, or by using the chain rule and first calculating the sensitivity of the functional with respect to shape deformations and then the sensitivity of the shape with respect to coefficient changes in the parametrization. Once this is done, fast optimization methods in $\mathbb{R}^d$ like the Conjugate Gradient algorithm or Quasi-Newton methods can be used to find the optimal coefficients in few iterations.

Among the advantages of this method is that it is relatively easy and intuitive to use once software that can calculate the derivative as described above has been created. Furthermore, since we only search in the parametrization space, a good choice of parametrization can guarantee that only sensible shapes are found and it is relatively easy to include constraints like the minimum thickness of an object or maximum amount of deformation of the original shape into the problem.

There are, however, two disadvantages to this approach which motivate the following work. Using a parametrization means that we are only searching in a finite dimensional space of shapes, which cannot contain all possible shapes and is likely to not contain the optimal shape. As an example: if we parametrize a shape by a polynomial, we will never be able to find a shape that contains sharp edges or kinks; any solution will only be an approximation. The second problem is that the optimization routine is not aware that it is optimizing over a space of shapes, instead it thinks of the problem as a finite dimensional problem over real numbers. While this makes the optimization easier, since we can just reuse existing routines, it also means that we lose the inherent structure of shape spaces. This means that increasing the number of coefficients in our parametrization usually leads to poor conditioning of the problem, i.e. an increasing number of necessary iterations of the optimization routine as one increases the number of coefficients. To be able to solve large problems it is hence necessary to analyse the problem in a general, infinite dimensional shape space.

Instead of parametrizing the shape of interest and calculating the derivative of the functional with respect to finitely many coefficient changes, we calculate the derivative with respect to general domain deformations. This derivative can then be used to obtain the deformation direction of our shape that reduces the functional as much as possible. This means that the optimization is done over general, infinite dimensional shape spaces and does not depend on a parametrization. The approximation is only done at the last step, when the domain and the deformation functions are represented by meshes and finite element functions respectively. However, it is crucial that the analysis and the conditioning of our problem does not depend of the resolution of the chosen discretization.

The second advantage of this approach is that we search in a larger shape space and that we retain the structure of the optimization problem. Since we do not restrict ourselves to a particular parametrization, the space of shapes that we consider is significantly larger and the algorithm might suggest shapes that were not considered by the engineer and hence cannot be represented by the chosen parametrization.

The main disadvantage however is the analytical difficulty - while optimizing in finite dimensional spaces is very well understood, optimization over shape spaces is far more subtle and there remain many opportunities for fundamental research.

2. Energy Minimisation in Stokes flow

We consider the following test problem as introduced in [1]. Liquid flows through a channel containing an obstacle in the centre, as shown in Figure 1. Given an obstacle Ω_o, we denote the fluid containing part of the channel by Ω_c; this means that a deformation of the obstacle Ω_o directly corresponds to a change of Ω_c. Assuming either small velocity, high viscosity or small length scales in the problem we know that the fluid flow satisfies the Stokes equations (and so inertia is neglected). This means that, for a given geometry, we can use finite element methods to solve for the velocity field u of the fluid.

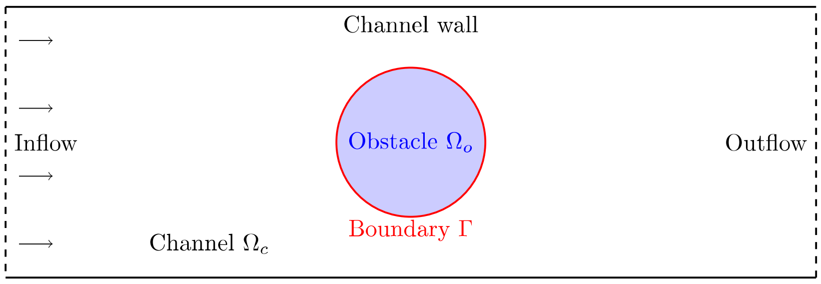


Figure 1: Sketch of the geometry under consideration in two dimensions. In three dimensions, we rotate this geometry along the central, horizontal axis.

Once we have obtained the velocity field it is possible to calculate the energy that is dissipated in the fluid by integrating the square of the norm of the gradient of the velocity field over the channel. Our goal is to deform the object in the centre of the channel to minimize this energy. However, it is intuitively clear that without any constraints one can minimize the energy lost around the obstacle by simply making it smaller and smaller until it vanishes. Hence we add the constraint that the volume should remain constant.

The Shape Derivative

There are various options how a volume constraint can be incorporated and we will not focus on this here. However, in order to minimize a function or functional quickly we need derivative information, as this will enable us to find the best descent directions at every iteration. In this case it is not immediately clear how the derivative with respect to a shape should be defined. In shape calculus, we consider deformations represented by vector fields W and the shape derivative in direction of a deformation is defined by how much the functional changes if the domain is deformed in direction of the vector. An example for such a deformation can be seen in Figure 2.

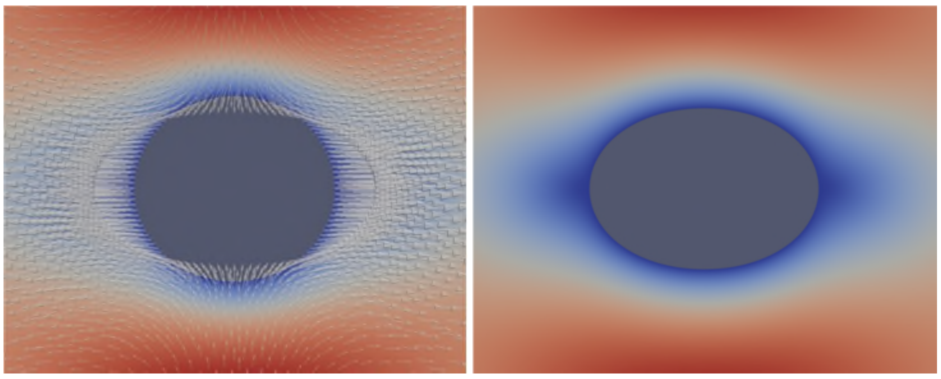


Figure 2: Left: Initial domain (circle) with arrows indicating the deformation direction W. Right: Deformed domain. The colours indicate the magnitude of the velocity field; red represents faster flow, blue stands for slower flow.

In the case of the energy functional for Stokes flow, the shape derivative in direction W can be expressed by two equivalent formulas. The first one is called the volume form of the derivative, and can be expressed by an integral over the channel, and the second form is the surface form and is given by an integral over the surface of the obstacle.

It is possible to calculate the shape derivative for other quantities, e.g. lift or downforce.

Assume that our current geometry is given by Ω_c and we want to know how the drag changes if we apply a deformation W. Then we need to solve Stokes equations to obtain the velocity fields u and then evaluate either the volume integral or the surface integral for the shape derivative. Though theoretically equivalent, the second form is often the preferred form, as it is more intuitive since it supports our intuition that a deformation field that does not change the boundary of the object has no influence on the objective. The volume form appears is often more complicated but it has been noticed (cf. [2]) in the past that it has numerical advantages and is more accurate than the surface form when implemented using finite elements.

Comments

- Since the velocity field u depends on the domain we need to include the fact that u solves the Stokes equations implicitly when deriving the shape derivative. This means, that for other fluid models like Navier-Stokes, Bernoulli or Reynolds-averaged Navier-Stokes the expression for the shape derivative will be different.

3. Steepest Descent and Limited Memory BFGS

Once we have obtained the shape derivative we are able to calculate the shape gradient using the structure on the shape space defined in [1, 3]. The shape gradient is a function defined on the boundary of the domain that tells us how much to deform the boundary in the normal direction to reduce the objective as much as possible. Since we do not want to regenerate the mesh in our domain, we extend this deformation function into the rest of the volume to obtain a deformation vector field that we can apply onto the mesh nodes.

A simple steepest descent algorithm now works as follow: solve the Stokes equations to obtain the fluid flow; calculate the shape gradient using either the surface or the volume form of the shape derivative; extend the deformation into the volume; deform the mesh; repeat.

For optimization algorithms over real numbers, it is possible to significantly increase convergence speeds to superlinear convergence by including information about the second derivative of the function being minimised. However, often calculating or storing the inverse of the second derivative is not feasible and in those cases one can build an estimate of the second derivatives using the previous iteration steps. One of the most popular algorithms that does this is the BFGS algorithm. The problem with this algorithm is that, as the algorithm progresses and the number of past iterations increases, the complexity and memory requirements of the algorithm increase since each iteration must take into account all previous iterations. The Limited-Memory BFGS algorithms counters this problem by only storing a fixed number of previous iteration steps and hence does not suffer from ever increasing iteration cost. This algorithm is well understood and has been used successfully for optimization problems over real numbers and has been previously adapted to our case of optimization over shape spaces (see [1, 3]).

4. Numerical Results

We have implemented the solver for Stokes equations and the two optimization algorithms using the firedrake software. In Figure 2, we show, both the initial shape (left), and the optimal shape that minimized the energy dissipation (right), where we have held the area of the obstacle constant.

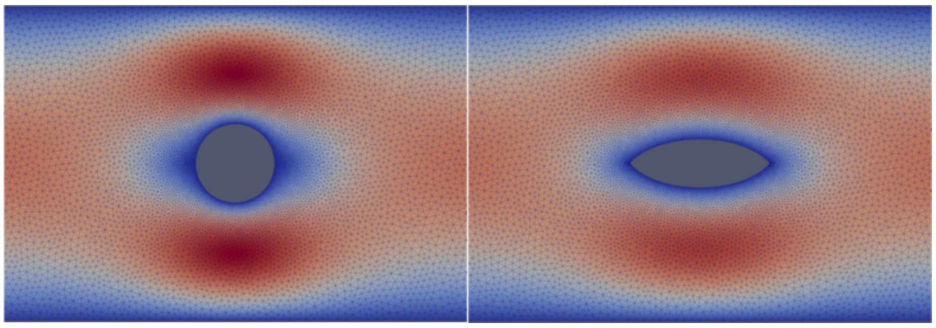


Figure 2: Initial shape (left) and final, optimal shape (right). The colours indicate the magnitude of the velocity field; red represents faster flow, blue stands for slower flow.

Steepest Descent vs L-BFGS

L-BFGS increases convergence speed significantly at small extra cost per iteration.

We begin by comparing the convergence speed of the two optimization algorithms. In Figure 3, we show that L-BFGS has essentially converged after about 40 iterations. It makes significantly more progress per iteration when compared against the steepest descent method, which we see requires significantly more iterations and still has not converged after more than 300 iterations. Since L-BFGS is a more complicated algorithm requiring multiple calculations to find the descent direction, each iteration is computationally more expensive than a simple steepest descent iteration. However, in practice this is negligible as the main cost of each iteration lies in the solve of the fluid equations, especially when the governing equations and the geometry under consideration become more complicated.

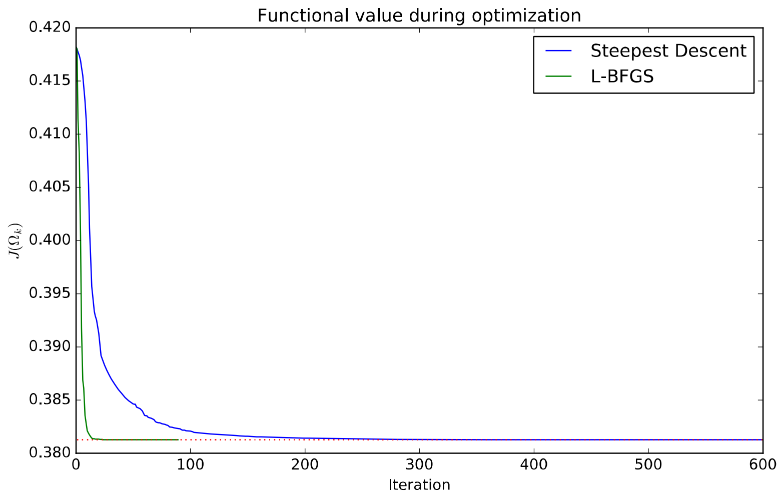


Figure 3: Functional value at each iteration of the two optimization algorithms.

Surface vs Volume Form of the Shape Derivative

We now compare the two different formulations of the shape derivative of the energy functional. As already explained, the surface form is more commonly used but the volume form often has better numerical properties when approximated using finite elements. It turns out that in our case the same holds true, especially for geometries that are not smooth but contain kinks. We illustrate this by considering as an initial shape a square that has been turned by 45 degrees so that it appears to stand on one of its kinks.

We use the L-BFGS algorithm to compare the final shape obtained when using the surface form and the volume form of the derivative and we show the results in Figure 4. We can see that the surface form struggles to flatten out the top and bottom kinks whereas the volume form has no problems. We have observed similar behaviour when facing the opposite task of creating a kink in a smooth surface. In that case the volume form was able to create a kink consistently, the surface form however would have stump kinks or overly sharp kinks, depending on parameter choice.

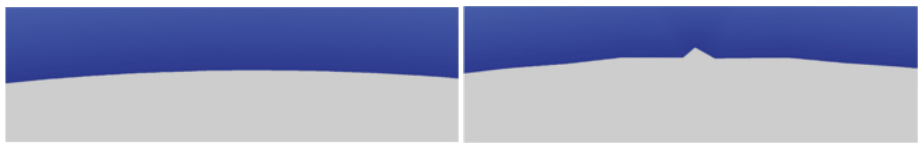


Figure 4: Top side of the final shape when using the volume form (left) or the surface form (right) of the shape derivative.

Extension to three dimensions

In order to show applicability of our work to real-world problems, we now extend the problem under consideration to three dimensions. This means that we consider fluid flow in a pipe around an object whose energy we want to minimize. In Figure 5, we show the optimal shape found using the volume form of the derivative. We ran this simulation on a 16 core CPU within a few hours and only about 3% of the computational effort was spent in the optimization. This supports our previous statement that, for realistic geometries, the extra cost due to L-BFGS compared to Steepest Descent is negligible.

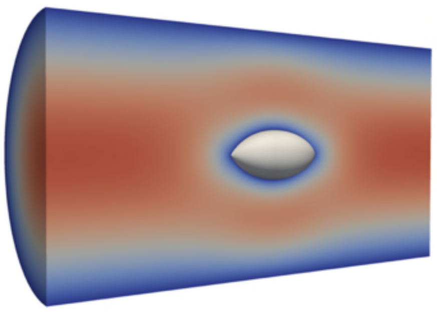


Figure 5: Optimal shape for the energy minimization problem in three dimensional Stokes flow, where the volume of the obstacle is preserved.

5. Discussion, conclusions and recommendations

To summarize, we draw two main conclusions: the L-BFGS optimization algorithm converges significantly faster than the Steepest Descent algorithm at little extra cost per iteration. Furthermore, the volume form generally appears to be a better choice than the surface form, since in our studies we have demonstrated that it is significantly more stable than the surface formulation.

We have identified a variety of future goals that can be broadly grouped into two categories. On the one hand we need to do more fundamental work into understanding the structure of the shape space better which will then hopefully enable us to formulate full Newton optimization schemes. This should lead to faster converging optimization algorithms. Deflation has previously been used successfully to find multiple solutions to a range of optimization problems that have more than one solution, but has yet to be used for shape spaces. Formulating a Newton method for shape optimization should allow us to adapt the theory of Deflation to shapes. The goal is to be able to find not just one but a range of optimal shapes (if more than one exists) which can then be suggested to an engineer who can pick the shape that is the most cost efficient, easiest to manufacture or pleasant to look at.

On the other hand, we need to make the fluid model and the geometries more realistic. In a first step this means deriving the shape derivative for the full Navier-Stokes equations on parts of a car and eventually the goal is to model the flow using the Reynolds-averaged Navier-Stokes equations with a turbulence model for the complete geometry of a car.

Potential Impact

Mark Taylor, CEO of London Computational Solutions, commented: "*Florian has shown an immediate grasp of the engineering problem we are trying to solve. LCS are determined to bring these powerful mathematical approaches to bear on real-world engineering problems which we know will be based on fundamental understanding of the equations and how they behave when being solved efficiently. I am really excited about working with Florian on his PhD project and feel certain that this will yield a commercial impact and see these techniques improving engineering products in the real world.*"

References

1. V. Schulz and M. Siebenborn (2016) *Computational comparison of surface metrics for PDE constrained shape optimization.* arXiv:1509.08601

2. R. Hiptmair, A. Paganini and S. Sargeheini (2015) *Comparison of approximate shape gradients.* BIT, 55(2):459-485.

3. V. Schulz, M. Siebenborn and K. Welker (2016) *Efficient PDE constrained shape optimization based on Steklov-Poincaré type metric.* arXiv:1506.02244.