

# Shape optimisation for linear stability



**Jack Brewster**

Department of Engineering  
University of Cambridge

First year report

Magdalene College

September 2016

For my parents and Ada

## **Declaration**

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 20,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Jack Brewster  
September 2016

## **Acknowledgements**

I would like to extend my gratitude to my supervisor, Professor Matthew Juniper. His advice and guidance have been invaluable and I am not sure how my first year would have gone without it!

Thanks must also go to Juan Guzman for helping me through the first months and putting up with my many, many questions. The Saplart-Allmaras turbulence model and the resolvent analysis would never have been implemented if it wasn't for his help.

I would like to thank my lab mates for providing such an enjoyable working environment: Alessando Orchini, Jose Aguilar, Berhnhard Semlitsch and Nick Jameson.

Ada now knows more about shape optimisation than she ever wanted to and I'd like to thank her for always being there and keeping me sane. Her support and guidance have helped me survive this first year.

Finally I would like to thank my family for their support and encouragement. This PhD would not have been possible without them and I cannot thank them enough.

## **Abstract**

The tools developed for performing shape optimisation and linear stability analyses are presented.

The shape optimisation tools are demonstrated on two problems - mixing of a scalar field in a laminar flow and minimisation of the viscous dissipation within a domain. The Hadamard structure of the gradient and its use in determining which regions of the shape are sensitive to changes is highlighted. A method for constraining the area of the shape via projection of the search direction is implemented.

The linear stability analysis tools are used to examine the stability of the flow past a cylinder and an axisymmetric disk. The shapes of the unstable modes compare favourably with the literature. A resolvent analysis is also performed and the pseudospectrum for the cylinder flow generated. The use of this pseudospectrum in exploring the difficulty of targeting certain eigenvalues in an optimisation routine is discussed.

The shape optimisation and stability tools are then combined in order to demonstrate the use of perturbation growth rate as an objective function. The shape gradient information is used to find the aspect ratio of an ellipse that gives marginal stability. A less restrictive parametrisation is then used in order to demonstrate the problem arising from a lack of smoothness of the shape gradient.

Finally the stability of the flow over a rotating cylinder is examined in order to demonstrate how the tools developed can be quickly used for other optimisation problems when stability forms part of the objective function.

# Table of contents

<b>List of figures</b>	<b>ix</b>
<b>Nomenclature</b>	<b>xi</b>
<b>1 Background</b>	<b>3</b>
1.1 Shape optimisation . . . . .	3
1.1.1 Historical overview . . . . .	3
1.1.2 Efficient gradient evaluation . . . . .	4
1.1.3 The adjoint method . . . . .	5
1.1.4 Shape calculus . . . . .	8
1.1.5 Parameter free optimisation . . . . .	9
1.1.6 Current state of the art . . . . .	10
1.2 Stability analysis . . . . .	10
1.2.1 Global analysis . . . . .	10
1.2.2 Resolvent analysis . . . . .	11
1.3 Combining shape optimisation and stability analysis . . . . .	12
<b>2 Computational methods</b>	<b>13</b>
2.1 Finite elements with FEniCS . . . . .	13
2.1.1 Solution of linear systems . . . . .	14
2.2 Finding the base-flow . . . . .	14
2.3 Shape parametrisation . . . . .	14
<b>3 Shape optimisation results</b>	<b>17</b>
3.1 Minimisation of viscous dissipation . . . . .	17
3.1.1 Motivation . . . . .	17
3.1.2 Problem statement . . . . .	17
3.1.3 Shape parametrisation and area constraint . . . . .	18
3.1.4 Gradient evaluation and optimisation routine . . . . .	19

3.1.5	Results . . . . .	20
3.2	An advection-diffusion problem . . . . .	23
3.2.1	Shape parametrisation . . . . .	26
3.2.2	Gradient evaluation and optimisation routine . . . . .	26
3.2.3	Results . . . . .	26
<b>4</b>	<b>Stability analysis</b>	<b>29</b>
4.1	Cylinder flow . . . . .	29
4.1.1	Motivation . . . . .	29
4.1.2	Global analysis . . . . .	30
4.1.3	Resolvent analysis . . . . .	30
4.2	Axisymmetric disk wake . . . . .	34
4.2.1	Motivation . . . . .	34
4.2.2	Modal analysis . . . . .	34
<b>5</b>	<b>Introducing stability to optimisation</b>	<b>37</b>
5.1	Shape optimisation for stability . . . . .	37
5.1.1	Motivation . . . . .	37
5.1.2	Finding an ellipse with marginal stability . . . . .	37
5.2	The problem of regularity . . . . .	39
5.3	Rotating cylinder . . . . .	41
5.3.1	Motivation . . . . .	41
5.3.2	Alterations to the optimisation code . . . . .	42
5.3.3	Results . . . . .	42
<b>6</b>	<b>Future Work and Research Plan</b>	<b>44</b>
6.1	Key work . . . . .	44
6.1.1	Swirl . . . . .	44
6.1.2	Separation metric . . . . .	44
6.1.3	Turbulence modelling . . . . .	45
6.2	Shape Hessian information . . . . .	45
	<b>References</b>	<b>48</b>
	<b>Appendix A Derivation of state equations for the featured optimisation problems</b>	<b>52</b>
A.1	Viscous dissipation in the flow . . . . .	52
A.2	Mixing of convected scalar on outlet plane . . . . .	55
A.3	Stability eigenvalue problem . . . . .	57

**Appendix B Automatic generation of geometric properties** **61**  
    B.1 Derivation of shape quantities . . . . . 61

# List of figures

2.1	Displacement fields for a Joukowsky airfoil . . . . .	16
3.1	Initial domain for viscous dissipation problem . . . . .	18
3.2	Projection method for area correction . . . . .	19
3.3	Percentage error in area during optimisation process . . . . .	21
3.4	Viscous dissipation convergence for variable maximum thickness shape . . . . .	21
3.5	Shape history for viscous dissipation with variable maximum thickness . . . . .	22
3.6	Comparison of base flows for initial and optimised shapes of the viscous dissipation problem . . . . .	23
3.7	Hadamard surface structure for initial steps . . . . .	24
3.8	Domain for advection-diffusion optimisation problem . . . . .	25
3.9	Convergence of objective function for mixing of scalar field . . . . .	27
3.10	Initial and end base flows and scalar fields for convected scalar problem . . . . .	27
3.11	Recirculation regions in the convected scalar problem . . . . .	28
3.12	Comparison between different parametrisations for convected scalar problem . . . . .	28
4.1	Vortex shedding behind a cylinder at $Re = 120$ . . . . .	30
4.2	Spectrum of global modes for cylinder flow, $Re = 50$ . . . . .	31
4.3	Direct and adjoint modes for a cylinder at $Re = 50$ . . . . .	32
4.4	Resolvent analysis for cylinder flow . . . . .	33
4.5	Pseudospectrum for cylinder flow, $Re = 60$ . . . . .	33
4.6	Disk wake base-flow, $Re = 120$ . . . . .	34
4.7	Spectrum for $m = 1$ azimuthal wavenumber in disk wake . . . . .	35
4.8	Mode shapes for the steady and unsteady $m = 1$ modes . . . . .	36
5.1	Determining ellipse aspect ratio for marginal stability . . . . .	39
5.2	Effect of shape gradient regularity on stability optimisation . . . . .	40
5.3	Surface gradient structure for stability optimisation of a cylinder . . . . .	41
5.4	Base-flow for rotating cylinder at $Re = 100$ . . . . .	43

---

5.5 Marginal stability curve for rotating cylinder . . . . . 43

6.1 Turbulence models on a flat plate boundary layer . . . . . 46

# Nomenclature

## Roman Symbols

$\mathbf{V}$	Displacement field
$g$	Hadamard structure
$i$	The imaginary unit
$\hat{\mathbf{n}}$	Unit normal pointing out of the domain
$J$	Objective function
$\mathbf{a}$	Vector of parameters defining the shape
$\mathcal{D}$	Set of admissible domains

## Greek Symbols

$\lambda$	The adjoint base-flow
$\xi$	The adjoint global mode
$\psi$	The adjoint scalar
$\beta$	Step distance for the steepest descent method
$\Gamma_{free}$	Free-stream boundary
$\Gamma_+$	Inlet boundary
$\Gamma_0$	Surface subject to optimisation
$\Gamma_-$	Outlet boundary
$\Gamma_w$	Wall boundary

$\omega$       Frequency of a perturbation

$\Omega$       The computational domain

$\sigma$       Growth rate of a perturbation

### **Superscripts**

†      Indicates adjoint of an operator

\*      Complex conjugate

# Motivation

Theoretical advances and the ready availability of computing resources have made computational fluid dynamics (CFD) usable as a design tool. Instead of using 'rules of thumb' and subsequent refinement to develop geometries of aircraft wings or turbomachinery, shape optimisation can be used to focus in on a design that maximises a performance criterion whilst satisfying the design constraints.

Whilst several modern CFD packages such as Ansys Fluent now have built in shape optimisation tools, they are often restricted to simple objective functions such as lift or drag. More complex objectives such as the stability of the flow or the mixing of an advected scalar field are beyond the reach of these solvers.

The stability of a flow can be an extremely important objective. The presence of an instability could cause the flow to transition to a different state when a control parameter is varied. This new state may have fundamentally different and undesirable properties such as unsteadiness, increased aerodynamic noise or an increased pressure loss. Targeting the stability of the flow by ensuring that any infinitesimal perturbation decays with time provides protection against this. Altering the geometry to achieve this aim results in a design that is physically simpler to implement than achieving it via motion of the surface of the shape or via blowing and sucking along the surface.

Stability can also be used as a constraint rather than the end objective. Suppose a duct has been optimised for a certain objective using a steady solution of the governing equations. If an instability is present this steady state may never be witnessed and an unsteady state that performs poorly may occur instead. If one constrains the flow to be stable then this problem will not arise.

The number of control parameters in a shape optimisation problem can range from a few parameters describing the general shape to the coordinates of each vertex that makes up the surface. When faced with such a large number of parameters an efficient means of calculating the gradient of the objective function is required. A method known as the adjoint approach provides an extremely efficient way of doing this.

---

The end goal of this PhD is to develop the tools required to perform shape optimisation with stability as an objective function and then apply this to a *cyclone separator*. Cyclone separators use a highly swirling flow to separate out dust particles. They are susceptible to a flow instability which causes the core of the vortex to precess. This precession reduces performance and is the cause of a considerable amount of unwanted noise. In view of this end goal work has been done on implementing an axisymmetric solver and optimisation of an objective function based on a scalar field, a possible proxy for a modelling the separation of dust particles.

# Chapter 1

## Background

### 1.1 Shape optimisation

#### 1.1.1 Historical overview

The use of computational fluid dynamics (CFD) as the basis of design methods first began with inverse methods. One of the first such methods was Lighthill's conformal mapping of potential flows to generate an aerofoil with a specified pressure distribution [20]. These methods are limited as they require advanced knowledge of what the optimal flow field most look like. For example, if one were use Lighthill's method to maximise lift the user would have to know the optimal pressure distribution in advance. Not only must the user be able to work this out, they require that such a pressure distribution can be physically achieved.

Shape optimisation overcomes these problems by trying to minimise a cost functions set by the user instead of directly targeting a specific flow state.

Shape optimisation problems take the general form:

$$\begin{aligned} \text{Find } \min_{\Omega \in \mathcal{D}} J(\mathbf{q}, \Omega) \\ \text{subject to } \mathbf{S}(\mathbf{q}, \Omega) = \mathbf{0} \end{aligned} \quad (1.1)$$

In the above,  $J$  represents a cost function based on the state,  $\mathbf{q}$ , and the domain,  $\Omega$ , that is to be minimised.  $\Omega$  is taken from a set of admissible domains,  $\mathcal{D}$ , that define the geometrical constraints. The state,  $\mathbf{q}$ , is a function of  $\Omega$  and is implicitly defined by  $\mathbf{S}$ , the system of partial differential equations (PDEs) and boundary conditions.  $\Omega$  is parametrised by the design variables,  $\Omega = \Omega(\mathbf{a})$ . The design variables are given by  $\mathbf{a} = \sum_{k=1}^K a_k \mathbf{e}_k$ , where  $\mathbf{e}_k$  is a basis for the design space and  $K$  represents the number of design variables. The number of design

variables may range from a few control points all the way to the coordinates of every point on the surface. If every surface point is free to move then the shape optimisation problem is termed *parameter-free* otherwise it is referred to as *parametrised* and the parametrisation of the shape must be specified.

Gradient-based methods are particularly efficient at finding local minima by using gradient information to calculate a search direction. For aerodynamic shape optimisation, gradient-based methods were first used by Hicks, Murman and Vanderplaats[12] where 2D airfoils were examined using a transonic potential flow code. Hicks and Henne then extended this technique to full wing design[11]. The techniques developed here used finite differencing (1.2) to approximate the gradient. This is computationally expensive and requires  $K + 1$  solutions of the state equation.

$$\frac{dJ}{da_k} \approx \frac{J(\mathbf{q}(\mathbf{a} + \varepsilon \mathbf{e}_k), \mathbf{a} + \varepsilon \mathbf{e}_k) - J(\mathbf{q}(\mathbf{a}), \mathbf{a})}{\varepsilon} \quad (1.2)$$

### 1.1.2 Efficient gradient evaluation

The finite difference has two key problems:

1.  $\mathbf{S}$  typically represents a solution of non-linear problems that are expensive to solve
2. The form of (1.2) is very sensitive to the choice of  $\varepsilon$  [28]. Too large a value gives a poor approximation of the gradient and too small a value causes large floating point errors due to the subtraction of two very similar terms in numerator

The  $\varepsilon$  dependency was resolved by the introduction of the complex-step method [23], (1.3).

$$\frac{dJ}{da_k} \approx \text{Im} \left( \frac{J(\mathbf{q}(\mathbf{a} + i\varepsilon \mathbf{e}_k), \mathbf{a} + i\varepsilon \mathbf{e}_k)}{\varepsilon} \right) \quad (1.3)$$

The complex-step method, in addition to requiring  $K + 1$  solutions of the state equation, mandates the use of a complex solver. It's accuracy at small values of  $\varepsilon$  made it a commonly used tool in the validation of other methods of obtaining the gradient in early work.

A potentially faster approach can be made by using the chain rule to express the derivative of the cost function and state equations

$$\frac{dJ}{da_k} = \frac{\partial J}{\partial \mathbf{q}} \frac{d\mathbf{q}}{da_k} + \frac{\partial J}{\partial a_k} \quad (1.4)$$

$$0 = \frac{\partial \mathbf{S}}{\partial \mathbf{q}} \frac{d\mathbf{q}}{da_k} + \frac{\partial \mathbf{S}}{\partial a_k} \quad (1.5)$$

A single solution of the state equation is required followed by  $K$  solutions of (1.6), the sensitivity equations. The sensitivities  $\frac{d\mathbf{q}}{da_k}$  are then substituted into (1.4) in order to get the shape gradient. As the sensitivity equations are linear and it is the same operator being inverted each time, this can often be more convenient than solving  $K$  non-linear systems[3].

$$\frac{d\mathbf{q}}{da_k} = - \left( \frac{\partial \mathbf{S}}{\partial \mathbf{q}} \right)^{-1} \frac{\partial \mathbf{S}}{\partial a_k} \quad (1.6)$$

The most efficient method for calculating the gradient extends the above and is known as the *adjoint method*.

### 1.1.3 The adjoint method

Inspired by the methods of optimal control developed by Lions[21], adjoint methods were introduced as a means of calculating the gradient much more efficiently. Piornneau used adjoints to find the minimum drag profile of a shape in Stokes flow[34] whilst Jameson used them for optimisation of aerodynamic surfaces [14] with both governed by both the Euler equations and the compressible Navier-Stokes equations.

The adjoint method uses Lagrange multipliers to convert the constrained optimisation problem in to an unconstrained one. An augmented function  $\mathcal{L}$  is defined with Lagrange multiplier,  $\mathbf{p}$ , and an appropriate inner product  $\langle \cdot, \cdot \rangle$ .

$$\mathcal{L} = \mathcal{J} - \langle \mathbf{p}, \mathbf{F} \rangle \quad (1.7)$$

A variational method is then employed by treating  $\mathbf{q}$ ,  $\mathbf{a}$  and  $\mathbf{p}$  as independent variables, The optimality point is found by setting variations with respect to  $\mathbf{q}, \mathbf{a}$  and  $\mathbf{p}$  to zero. The variation with respect to  $\mathbf{p}$  yields the state constraint whilst the variation with respect to  $\mathbf{q}$  leads to the adjoint equation, (1.8).

$$\frac{\partial \mathcal{J}}{\partial \mathbf{q}} - \left( \frac{\partial \mathbf{S}}{\partial \mathbf{q}} \right)^\dagger \mathbf{p} = \mathbf{0} \quad (1.8)$$

In the above, the definition of the adjoint of a linear operator  $L$ , equation (1.9) has been used where  $\dagger$  indicates the adjoint.

$$\langle u, Lv \rangle = \langle L^\dagger u, v \rangle \quad (1.9)$$

The gradient with respect to the design parameters is then found by taking variations with respect to the design parameters to get equation 1.10.

$$\frac{d\mathcal{J}}{da_k} = - \left( \frac{\partial \mathbf{S}}{\partial a_k} \right)^\dagger \mathbf{p} + \frac{\partial \mathcal{J}}{\partial a_k} \quad (1.10)$$

In the literature  $\mathbf{p}$  is termed the adjoint state or co-state. The adjoint equation is linear and specific to the objective function. In order to evaluate the gradient we require one solution of the state equation, one solution of the adjoint equation and then  $K$  evaluations of (1.10). The  $\frac{\partial \mathcal{J}}{\partial a_k}$  term is often known analytically and the only difficulty that may arise is with the so-called grid sensitivities,  $\frac{\partial \mathbf{S}}{\partial a_k}$ .

The grid sensitivities represent how the residual changes as points in the mesh move. The motion of the points on the surface of the deforming shape is well known and can be derived analytically based on the parametrisation of the shape. The motion of the interior points within the mesh depends on how the mesh is generated and can present a problem.

For structured meshes with an analytical expression for the location of the mesh coordinates then an analytical form for the grid sensitivities can often be found. This was the approach used by Jameson's early work with structured body-fixed meshes[35, 28]. The method for finding the grid sensitivities often depends on whether the adjoint method is applied to the governing equations before or after they have been discretised.

### The discrete adjoint

Applying the adjoint method to the discretised equations leads to the discrete adjoint approach. Discretisation leads to the state variable being represented by a finite dimensional vector and the linear operators being represented by matrices.

The appropriate inner product becomes the complex dot product and the adjoint of an operator becomes the conjugate transpose of its matrix representation. For certain solvers such as finite element solvers this allows the adjoint state to be found very quickly as the matrix representation of the linear operators are normally calculated when solving the state equations.

The key advantage of the discrete adjoint is that the gradient it produces is consistent with the discrete system[28]. Given a level of refinement,  $h$ , of the mesh then the discrete adjoint method finds  $\frac{d\mathcal{J}_h}{da_k}$ , the *exact* gradient of the approximate objective function,  $\mathcal{J}_h$ . This is referred to as consistency of the gradient. If the gradient is inconsistent then difficulties may arise in constrained optimisation problems [19].

In the discrete case the grid sensitivities require the derivative of the discretisation. If they haven't been eliminated analytically they can be calculated by:

1. *Finite differencing* - for each  $a_k$  the mesh is perturbed whilst keeping the value of the state at each vertex the same. The new residual field is calculated and this is then used to estimate the grid sensitivity. For large meshes this may be extremely expensive
2. *Automatic differentiation*[47] - the source code itself is differentiated and so the exact derivative of the discretisation with respect to each  $a_k$  is found. This requires access to the source code and is difficult to implement.
3. *Grid displacement model*[31] - the mesh is modelled as a linear elastic solid and the approximate displacement is calculated based on the solution of this. To make this process efficient the adjoint method may be applied to this displacement model as well
4. *Freezing of interior points* - the motion of interior points in the mesh is neglected. This method is computationally cheap but eliminates the advantage of the discrete adjoint approach

### The continuous adjoint

Applying the adjoint method directly to the governing equations leads to the continuous adjoint approach. This approach is inconsistent with the discrete system, providing an approximation to the gradient of the exact objective function,  $\left(\frac{dJ}{da_k}\right)_h$ .

For the continuous adjoint approach the relevant inner product is normally taken to be:

$$\langle \mathbf{p}, \mathbf{q} \rangle = \int_{\Omega} \mathbf{p}(\mathbf{x})^* \cdot \mathbf{q}(\mathbf{x}) d\Omega \quad (1.11)$$

The governing equation and boundary conditions of the adjoint is then found by successive applications of the divergence theorem. For flow problems the resulting adjoint equation has a similar form to the original governing equations except with the sign of the advection term reversed[37]. These equations have the advantage that, with a finite volume or finite difference solver, they can be solved using the same method as the direct system except with time-stepping now going backwards in time instead of forwards. For this reason the continuous adjoint framework has been very popular and developed extensively [29, 41]. Formulating the continuous adjoint can be difficult and has proved problematic for turbulence models, especially those with wall functions and so until recently the discrete adjoint approach has chiefly been used for turbulent flows.

The interior contribution to the grid sensitivities still remains in the formulation of the continuous adjoint problems and the methods used to find them for the discrete adjoint approach all apply here. Early authors noted[16, 32], however, that the interior contribution

could be eliminated by application of the divergence theorem. The expression for the shape gradient can then be arranged as a single surface integral over the shape being optimised. This is the product of a result from shape calculus - the *Hadamard structure theorem*[6]. The theorem states that for a displacement vector,  $\mathbf{V}_k$ , associated with parameter  $a_k$  then after a certain amount of manipulation gradient can be expressed in the form of (1.12). If the gradient is expressed in this form then it is said to be in *Hadamard form*.

$$\frac{d\mathcal{J}}{da_k} = \int_{\Gamma} \langle \mathbf{V}_k, \hat{\mathbf{n}} \rangle g d\Gamma \quad (1.12)$$

The Hadamard structure,  $g$ , can be used to calculate the value of  $\frac{d\mathcal{J}}{da_k}$  for each  $a_k$  or can even be used to update the surface directly when using a parameter free approach. In sections A.1 and A.3 we derive the Hadamard form for two different optimisation problems.

It is worth noting that the Hadamard form is derived in the continuous limit. For particularly coarse meshes then a more accurate gradient may be obtained by dealing with the interior points in the same manner as the discrete adjoint [18].

### Admissability

It has been noted that for certain objective functions (1.8) cannot be solved. This occurs only in objective functions based on boundary integrals and arises because the operator  $\left(\frac{\partial \mathbf{S}}{\partial \mathbf{q}}\right)^\dagger$  may not put appropriate terms on to the boundaries that are able to eliminate the boundary terms of  $\frac{\partial \mathcal{J}}{\partial \mathbf{q}}$ . Such objective functions are termed inadmissible.

A problem can be made admissible either by adding auxiliary boundary conditions [1] or by posing the adjoint governing equations in the weak form[19]. This would suggest the use of finite elements as a spatial discretisation as they work directly with the weak forms of the governing equations.

### 1.1.4 Shape calculus

Shape calculus is the rigorous way of deriving the Hadamard form for the gradient. It's origins are discussed in [8] and recent developments have been driven predominantly by Zolesio and Delfour [6].

Shape calculus introduces a perturbed domain  $\Omega_t$  defined by the bijective mapping:

$$T_t(\Omega) := \{T_t(\mathbf{x}) : \mathbf{x} \in \Omega\} \quad (1.13)$$

There are two chief methods for defining  $T_t$  - the *perturbation of the identity* and the *speed method*. Both approaches agree for first order shape derivatives but disagree for higher derivatives [38]. The perturbation of the identity is defined in equation (1.14).

$$T_t[\mathbf{V}](\mathbf{x}) = \mathbf{x} + t\mathbf{V}(\mathbf{x}) \quad (1.14)$$

The shape derivative in direction  $\mathbf{V}$  is then defined as:

$$d\mathcal{J}[\mathbf{V}] = \lim_{t \rightarrow 0} \frac{\mathcal{J}(\Omega_t) - \mathcal{J}(\Omega)}{t} \quad (1.15)$$

We examine only objective functions such as (1.16). Objective functions whose form is a surface integral over the deforming shape are much harder due to difficulties in evaluating the behaviour of the surface normal during the deformation of the shape [38].

$$\mathcal{J} = \int_{\Omega} F(\mathbf{q}) d\Omega \quad (1.16)$$

The shape derivative of such an objective function, where  $\Gamma_0$  indicates the region of the surface being deformed, is [39]:

$$d\mathcal{J}[\mathbf{V}] = \int_{\Omega} \frac{\partial F}{\partial \mathbf{q}} \mathbf{q}' d\Omega + \int_{\Gamma_0} \langle \mathbf{V}, \hat{\mathbf{n}} \rangle F d\Gamma \quad (1.17)$$

$$\text{where } \frac{\partial \mathbf{S}}{\partial \mathbf{q}} \mathbf{q}' = 0 \text{ in } \Omega \quad (1.18)$$

$$\mathbf{q}' = -\langle \mathbf{V}, \hat{\mathbf{n}} \rangle \frac{\partial \mathbf{q}}{\partial n} \text{ on } \Gamma_0 \quad (1.19)$$

The sensitivity system is given by (1.18) and (1.19). It is this system that we get the adjoint operator of in order to derive the governing equations for the adjoint state.

By choosing the adjoint governing equations to eliminate  $\mathbf{q}'$  and substituting in (1.19) where possible we can find any admissible objective function in Hadamard form.

### 1.1.5 Parameter free optimisation

For parametrised problems the Hadamard form can be used to evaluate  $\frac{d\mathcal{J}}{da_k}$ . For each  $a_k$  we calculate the displacement field  $\mathbf{V}_k$  and integrate over the surface.

Alternatively the Hadamard form could be used as to directly update the shape by applying the displacement field  $-\beta g \hat{\mathbf{n}}$ . This approach is used in parameter-free approaches. It can be shown, however, that the Hadamard structure lacks the smoothness of the original shape [26]. This results in a loss of smoothness and smoothing methods such as the traction

method[2] or Sobolev smoothing[15] are required in order to maintain the regularity of the shape. This is often not a problem for parametrised shapes as the parametrisation can be chosen to guarantee their smoothness.

### 1.1.6 Current state of the art

Besides a steady stream of new applications, recent research in shape optimisation has focused on three fields:

1. *Turbulence modeling*[48, 49] - continuous adjoint formulations for turbulence models and wall functions have recently been developed. Previously these problems required a discrete formulation and so couldn't take advantage of the Hadamard form.
2. *Shape Hessian*[40] - it has been shown that the shape Hessian has a similar Hadamard like structure as the shape gradient[5]. Knowledge of the Hessian would allow faster convergence and can be used to develop new preconditioners. Difficulties arise however when trying to use the shape Hessian with standard optimisation algorithms as the Hessian is usually not symmetric.
3. *Automatic differentiation* [27],- this eliminates the difficulties in finding the grid-sensitivities for the discrete adjoint formulation through direct differentiation of the source code.

New fields of shape optimisation are beginning to emerge as well. These include topological optimisation, done by allowing the porosity of regions of the mesh to change[33] and robust optimisation where geometric certainties are introduced into the formalism[36].

## 1.2 Stability analysis

The objective of this PhD is to combine the above shape optimisation framework with a linear stability analysis. A brief overview of both global and resolvent analysis is covered here.

### 1.2.1 Global analysis

The incompressible Navier–Stokes equation, (1.20), is perturbed about a steady state,  $\mathbf{u}_b$  through  $(\mathbf{u}, p) \rightarrow (\mathbf{u}_b + \varepsilon \mathbf{u}', p_b + \varepsilon p')$ .

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} = 0 \quad (1.20)$$

At order epsilon we get the governing equation of the perturbation

$$\frac{\partial \mathbf{u}'}{\partial t} + (\mathbf{u}' \cdot \nabla) \mathbf{u}_b + (\mathbf{u}_b \cdot \nabla) \mathbf{u}' + \nabla p' - \nu \nabla^2 \mathbf{u}' = 0 \quad (1.21)$$

Global modes [17] are found by looking for solutions of the form

$$(\mathbf{u}', p') = (\hat{\mathbf{u}}, \hat{p})(x, y) e^{(\sigma + i\omega)t} \quad (1.22)$$

In (1.22)  $\sigma$  is the growth rate of the global mode whilst  $\omega$  is its frequency. Substituting this Ansatz into (1.21) leads to the generalised eigenvalue problem:

$$\mathbf{A} \begin{pmatrix} \hat{\mathbf{u}} \\ \hat{p} \end{pmatrix} = s \mathbf{B} \begin{pmatrix} \hat{\mathbf{u}} \\ \hat{p} \end{pmatrix} \quad (1.23)$$

$$\mathbf{A} = \begin{pmatrix} ((\cdot) \cdot \nabla) \mathbf{u}_b + (\mathbf{u}_b \cdot \nabla)(\cdot) - \nu \nabla^2(\cdot) & \nabla(\cdot) \\ \nabla \cdot (\cdot) & 0 \end{pmatrix} \quad (1.24)$$

$$\mathbf{B} = \begin{pmatrix} -1 & 0 \\ 0 & 0 \end{pmatrix} \quad (1.25)$$

$$s = \sigma + i\omega \quad (1.26)$$

If any of the eigenvalues have a positive real component these are referred to as unstable global modes. As the time tends to infinity these unstable perturbations will grow exponentially.

## 1.2.2 Resolvent analysis

If no unstable global modes are present then the perturbation will decay to zero as the time tends to infinity. In finite time, however, there may be some amplification due to non-normality of the eigenvectors[46]. This is the case for flat plate boundary layers[43].

As opposed to examining the behaviour of the flow when a perturbation is applied, the resolvent analysis applies a harmonic forcing at order  $\varepsilon$  to the governing equations.

$$(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} = \varepsilon \mathbf{f} \quad (1.27)$$

Looking for harmonic responses with a zero growth-rate, we can use a prolongation operator,  $P$ , to express them in the form

$$\hat{\mathbf{u}} = C(\omega)\hat{\mathbf{f}} \quad (1.28)$$

$$\text{where } C(\omega) = P^T(i\omega B - A)P$$

The singular values of this give the amplification of the forcing. By decomposing  $\hat{\mathbf{f}}$  into its leading singular modes a series of optimal forcings and responses can be found. The singular values give the amplification of the response corresponding to each forcing. By sweeping through the frequency of the forcing the optimal frequency for forcing the system can be found. For systems governed by normal matrices this frequency will coincide with a direct global mode. If there is a large degree of non-normality the resonant frequency may be a long way from any mode present in the spectrum.

### 1.3 Combining shape optimisation and stability analysis

Few authors have combined shape optimisation and linear stability analysis.

Marquet et al. [24] used adjoint methods to find how the growth rate of an unsteady global mode responds to a forcing in the base-flow. This was then used to find the optimal position for a small control cylinder in order to suppress vortex shedding.

At about the same time Strauss and Heuveline [45, 10] examined shape optimisation for hydrodynamic stability of a cylinder. They examined the use of perturbation growth rate as both an objective function and a constraint. Their approach, however, relied on finite differencing and the use of a bionic algorithm.

The only group to have published on a gradient-based method for shape optimisation using the adjoint method to efficiently calculate the shape gradient are Nakazawa and Azegami [30]. In their paper they examine optimisation of a suddenly expanding duct. A flow instability whereby the flow will suddenly switch to a new steady state and attach to the wall is present in that configuration. They found that the addition of a small ramp eliminates this instability.

This group have also done work on shape optimisation of other eigenvalue problems such as the resonant frequency of a brake disk[42] in order to suppress brake squeal.

# Chapter 2

## Computational methods

### 2.1 Finite elements with FEniCS

As discussed in the background text in section 1.1.3, a spatial discretisation using continuous Galerkin finite elements is advantageous. This allows the governing equations of the adjoint to be expressed in their weak form, eliminating the problem of inadmissible objective functions.

The finite element package FEniCS[22] was chosen as it provides an easy to use Python interface and has automatic parallelisation, enabling much faster development of the parallel codes required for large scale optimisation problems. A large number of different element types have been implemented by the developers and updates are released regularly. This allows us to focus on implementing the shape optimisation code rather than having to develop a finite element solver.

The Python interface allows many other useful packages such as *scipy*, *numpy* and *sympy* to be used in conjunction with the solver. In addition Python interfaces are available for PETSc and SLEPc, two well known linear algebra and eigensolver packages for sparse matrices. This allows greater flexibility than the interfaces provided by FEniCS alone.

One disadvantage of using FEniCS is its lack of support for complex matrices. These occur when examining swirling flows finding the adjoint base-flow. For swirling flows  $i$  now appears in the governing equations when a non-zero azimuthal wavenumber is used. Complex terms also arise solving for the adjoint base flow because the complex eigenvector of the stability problem now acts as a source term to the governing equations. This problem has been resolved by separating out the real and imaginary parts of the governing equations and solving the  $N$  equations simultaneously as a  $2N \times 2N$  matrix.

### 2.1.1 Solution of linear systems

The finite element method produces a linear system governed by a sparse matrix. The direct solver MUMPS has been chosen as the linear solver. MUMPS is a parallel code for the solution of linear systems governed by large sparse matrices. MUMPS can be used through the PETSc interface and so is available without needing to install additional packages.

Whilst direct solvers do not scale as well as iterative solvers such as GMRES, they don't require preconditioning of the problem. As the problems dealt with here are sufficiently small, finding the appropriate preconditioner would entail additional development effort for little gain.

## 2.2 Finding the base-flow

In this report a base-flow is considered to be any steady solution of the incompressible Navier-Stokes equations. When performing an optimisation study the base-flow is the flow about which we optimise. For a linear stability analysis the base-flow is the steady state we apply the perturbation to.

Steady solutions are found using a Newton method with an initial guess of a uniform flow with speed 1. For the laminar flow problems featured here this process typically converges within ten iterations. For higher Reynolds number flows or for unusual shapes this process may diverge. If the solution fails to converge within a certain number of iterations, or divergence is detected, then the Reynolds number is reduced by raising the viscosity. The viscosity is then gradually reduced until the target Reynolds number is achieved, each time using the solution of previous viscosity as the initial guess for the next viscosity.

## 2.3 Shape parametrisation

Whilst the end goal is to use a CAD-like parametrisation for geometries in order to allow a fast transition between the optimisation and manufacturing stages of the design, other parametrisations have been used in order to explore the behaviour of certain shape optimisation problems.

An automated code for calculating the required geometric properties has been developed by combining a tensor based approach[9] with the symbolic algebra package *sympy*. Shapes with large numbers of parameters or where there is a complex relationship between the end shape and its control parameters e.g. a Joukowski airfoil, can involve a large amount of tedious and error-prone hand differentiation when deriving these properties by hand. The

automated approach allows a range of different parametrisations to be used with minimal additional development time.

The tensor derivation is given in appendix B but the method can be quickly used for any 2D shape that can be defined by a surface coordinate. The simplicity is demonstrated below with a snippet of code for defining a Joukowski airfoil.

```

from shape import Shape2D
import sympy as sp

class JoukowskiAirfoil(Shape2D):
    def generate_shape_expressions(self):
        x0,y0,R, theta = sp.symbols("x0_yo_R_theta")
        self.surfaceCoordinate=theta
    self.params = [x0,y0,R]
        self.xExpression = (x0+R*sp.cos(theta))*(1+1/((x0+R*sp.
            cos(theta))**2 + (y0+R*sp.sin(theta))**2))
    self.yExpression = (y0+R*sp.sin(theta))*(1-1/((x0+R*sp.cos(theta)
        ))**2 + (y0+R*sp.sin(theta))**2))

```

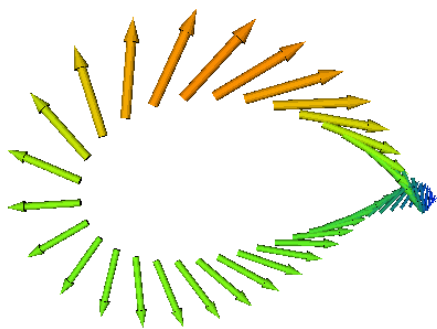
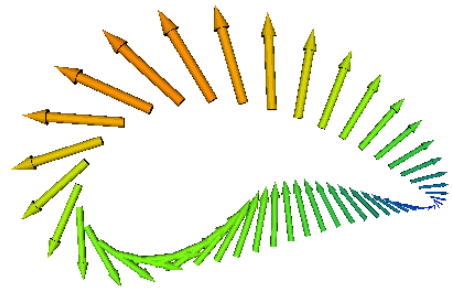
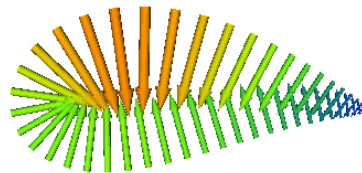
The Shape2D class then automatically generates the derived fields and returns them via the following calls:

```

gen_vertex_list(mesh)
gen_curvature_field(mesh)
gen_normal_field(mesh)
gen_displacement_field(mesh, paramIndex)

```

The three parameters that govern the shape of the Joukowski airfoil are the two centre coordinates and the radius of the circle to which the Joukowski conformal mapping is applied. The automatically generated displacement fields for the three parameters are shown in figure 2.1.

(a) Displacement field due to  $x$  coordinate(b) Displacement field due to  $y$  coordinate

(c) Displacement field due to radius

Fig. 2.1 Automatically generated displacement fields for a Joukowski airfoil

# Chapter 3

## Shape optimisation results

Two shape optimisation problems were undertaken in order to demonstrate the progress made in developing the relevant tools. The first involves minimising the viscous dissipation, a proxy for drag, in the fluid around a shape. The second involves optimising a duct to increase mixing of a scalar field.

### 3.1 Minimisation of viscous dissipation

#### 3.1.1 Motivation

Minimising the drag of a body is a common objective for many engineering applications. As discussed in the previous section, objective functions with the form of a surface integral over the deforming shape are much more difficult than field integrals and can have the problem of admissibility. The viscous dissipation within the domain is therefore taken as a volume integral proxy to the drag. This leads to an objective function of the form:

$$\mathcal{J} = \int_{\Omega} \sum_{i,j=1}^2 \left( \frac{\partial u_i}{\partial x_j} \right)^2 d\Omega \quad (3.1)$$

#### 3.1.2 Problem statement

The problem statement is given by

$$\text{Find } \min_{\Omega \in \mathcal{D}} \mathcal{J}(\mathbf{u}, \Omega) \quad (3.2)$$

$$\text{subject to: } (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} = 0 \text{ in } \Omega \quad (3.3)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \quad (3.4)$$

$$\mathbf{u} = \mathbf{u}_+ \text{ on } \Gamma_+ \quad (3.5)$$

$$p \mathbf{n} - \nu \frac{\partial \mathbf{u}}{\partial n} = 0 \text{ on } \Gamma_- \quad (3.6)$$

$$\mathbf{u} = 0 \text{ on } \Gamma_0 \quad (3.7)$$

$$\frac{\partial u}{\partial y}, v = 0 \text{ on } \Gamma_{free} \quad (3.8)$$

Where  $\Gamma_+$ ,  $\Gamma_-$ ,  $\Gamma_{free}$  and  $\Gamma_0$  represent the inlet, outlet, free-stream and optimising boundaries respectively. If there were no restrictions on  $\mathcal{D}$  then the solution of the above would be a shape of vanishing area. We therefore require that  $\mathcal{D}$  is the set of all shapes with specified area,  $A_0$ , and appropriate smoothness. This constraint is enforced by a projection method performed at each iteration.

The cross-stream thickness of the shape is allowed to vary and the viscosity is set to give an initial Reynolds number of 50 in order to allow comparison with the stability problem of section 5.2.

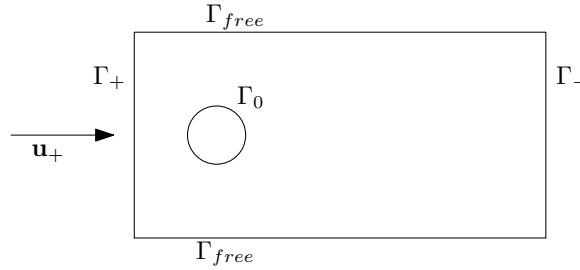


Fig. 3.1 Initial domain for the viscous dissipation optimisation problem.

### 3.1.3 Shape parametrisation and area constraint

The parametrisation of the shape is given by a Fourier series in polar coordinates, equation 3.9. This parametrisation provides an easy to implement orthogonal basis for the design space and does not place excessive restrictions on the form of the final shape.

$$r = a_1 + \sum_{n=1}^{\frac{N-1}{2}} a_{2n} \sin(n\theta) + a_{2n+1} \cos(n\theta) \quad (3.9)$$

To apply the area constraint, every iteration of the optimisation we project the search direction onto the plane given by equation 3.10. This performs an area correction accurate to first order. This is visualised in figure 3.2.

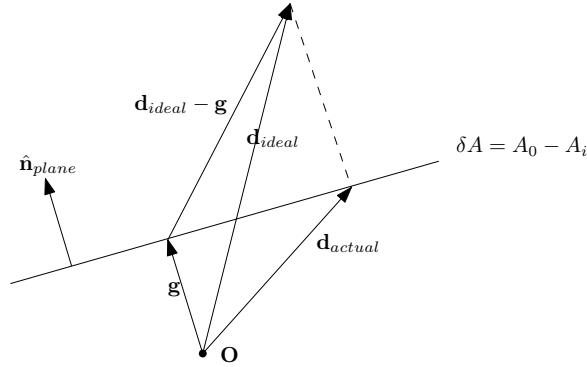


Fig. 3.2 Correcting the area by a projection method.  $\mathbf{d}_{ideal}$  is the ideal search direction for the optimisation problem whilst  $\mathbf{g}$  is the distance of the hyperplane from the origin. The hyperplane is the set of all search directions that gives the first order area change  $\delta A$ . The actual search direction is  $\mathbf{d}_{actual} = \mathbf{d}_{ideal} - (\mathbf{d}_{ideal} - \mathbf{g}) \cdot \hat{\mathbf{n}}_{plane}$ .

$$\sum_{i=1}^N \frac{\partial A}{\partial a_k} \delta a_k = A_0 - A_i \quad (3.10)$$

The quantities  $\frac{\partial A}{\partial a_k}$  are calculated by equation 3.11. This is a straightforward results found by taking the domain volume as the objective function  $\mathcal{J} = \int_{\Omega} 1 d\Omega$ .

$$\frac{\partial A}{\partial a_k} = - \int_{\Gamma_0} \langle V_k, \hat{\mathbf{n}} \rangle d\Gamma \quad (3.11)$$

It must be noted that, in the initial state, the first shape derivative of the area is always zero for the sine and cosine terms whilst the second derivatives are not. The first iteration will therefore violate the area constraint due to a non-zero second order change in area. The correction applied by subsequent iterations will eventually eliminate this.

### 3.1.4 Gradient evaluation and optimisation routine

Finding the gradient requires finding the adjoint base-flow  $\boldsymbol{\lambda}$ . The governing equations are derived in section A.1 and are found to be:

$$\mathbf{v}\nabla^2\boldsymbol{\lambda} + \nabla\lambda_p + (\mathbf{u}\cdot\nabla)\boldsymbol{\lambda} + (\nabla\boldsymbol{\lambda})^T\mathbf{u} = 2\nu\nabla^2\mathbf{u} \text{ in } \Omega \quad (3.12)$$

$$\nabla\cdot\boldsymbol{\lambda} = 0 \text{ in } \Omega \quad (3.13)$$

$$\boldsymbol{\lambda} = 0 \text{ on } \Gamma_+ \quad (3.14)$$

$$\boldsymbol{\lambda} = 0 \text{ on } \Gamma_0 \quad (3.15)$$

$$\lambda_p\hat{\mathbf{n}} + \nu\frac{\partial\boldsymbol{\lambda}}{\partial n} + \langle\boldsymbol{\lambda},\mathbf{u}\rangle\hat{\mathbf{n}} + \langle\hat{\mathbf{n}}\mathbf{u}\rangle\boldsymbol{\lambda} - 2\nu\frac{\partial\mathbf{u}}{\partial n} = 0 \text{ on } \Gamma_- \quad (3.16)$$

The resulting expression for the gradient with respect to  $V_k$ , the perturbation field associated with  $a_k$ , is given by (3.17)

$$d\mathcal{J}[V] = \int_{\Gamma_0} \langle V, \hat{\mathbf{n}} \rangle \sum_{i=1}^2 \left[ -\nu \left( \frac{\partial u_i}{\partial x_j} \right)^2 + \nu \frac{\partial \lambda_i}{\partial n} \frac{\partial u_i}{\partial n} \right] \quad (3.17)$$

Every iteration the base-flow and adjoint flow are found for the current shape. The ideal search direction is then given by the steepest descent method:

$$\mathbf{d}_{ideal} = -\beta \sum_{i=1}^N d\mathcal{J}[V_k] \mathbf{e}_k \quad (3.18)$$

The quantity  $\beta$  is small and positive to ensure that the objective function decreases. Before applying the update we use the method of figure 3.2 and project  $\mathbf{d}_{ideal}$  to get the actual search direction  $\mathbf{d}_{actual}$ .

### 3.1.5 Results

The optimisation program was run with both  $N = 65$  and  $N = 49$  from the same initial configuration of a circle with unit diameter. The initial Reynolds number based on the maximum thickness was 50.

The projection method for satisfying the area constraint was found to work well but required a large number of iterations,  $\approx 50$ , before the area violation from the initial step was cancelled as seen in figure 3.3.

Both parametrisations achieved a reduction in the viscous dissipation in the fluid of  $\approx 36\%$  and had similar rates of convergence as seen in figure 3.4. The shape at various instances during the optimisation process can be seen in figure 3.5.

Examining the base-flows of the initial and final configuration, figure 3.6, we can see that by moving to a thinner streamlined shape the length of the wake has been greatly reduced

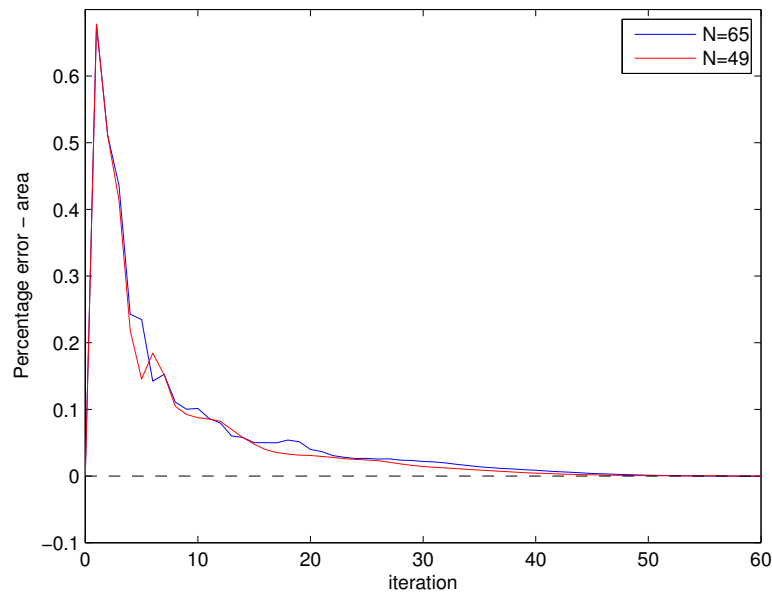


Fig. 3.3 Percentage error in area,  $100\frac{A_i - A_0}{A_0}$ , during the optimisation process for different shape resolutions. In both cases the error from the initial step is mostly eliminated after  $\approx 50$  iterations.

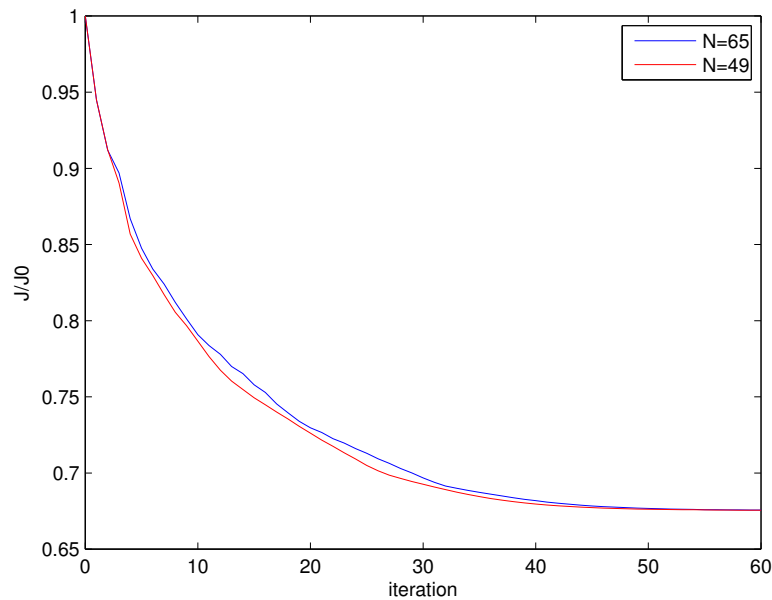


Fig. 3.4 Convergence of the objective function for viscous dissipation using the two different parametrisations. Similar rates of convergence are achieved for both.

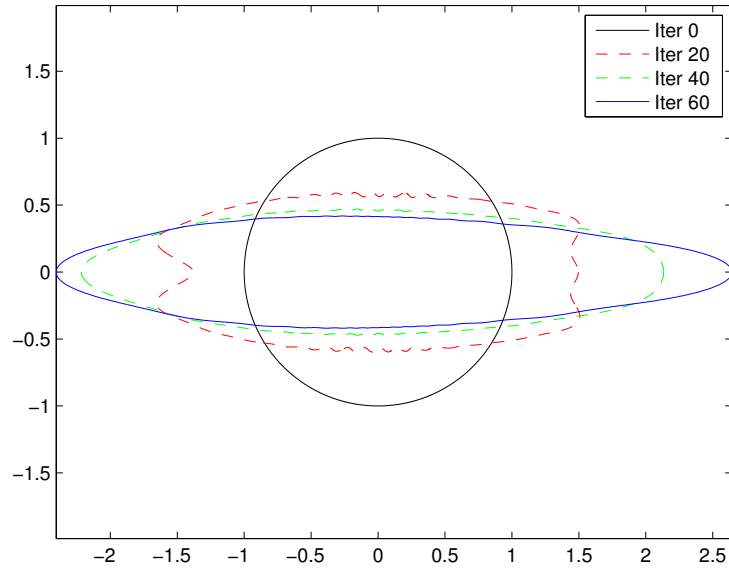


Fig. 3.5 Initial, interim and final shapes for an objective function based on viscous dissipation using the  $N = 65$  parametrisation.

and the recirculation bubble has been eliminated. The solution here is a compromise between form drag and skin friction - as the shape gets thinner and thinner the form drag decreases but the dissipation due to skin friction increases thanks to the increased surface area of the shape.

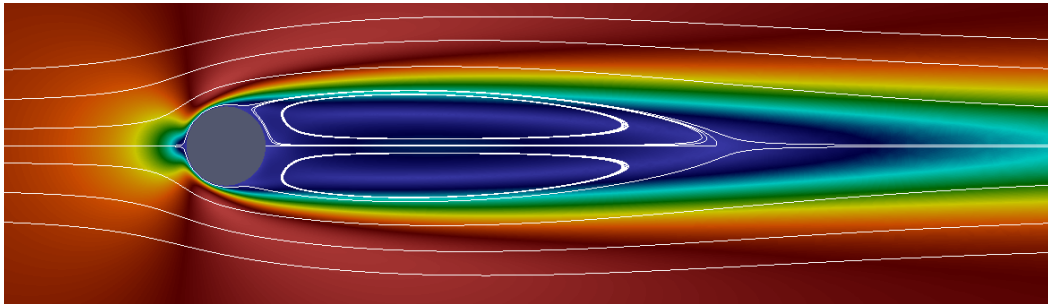
As the expression for  $dJ[V_k]$  is in Hadamard form, by plotting

$$\sum_{i=1}^2 \left[ -\nu \left( \frac{\partial u_i}{\partial x_j} \right)^2 + \nu \frac{\partial \lambda_i}{\partial n} \frac{\partial u_i}{\partial n} \right]$$

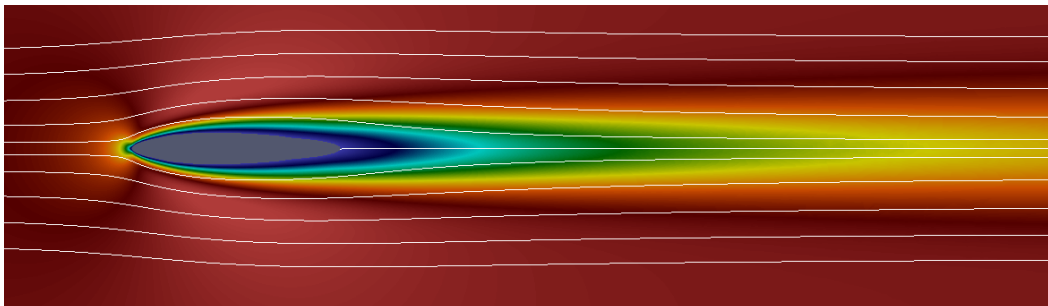
along the surface of the shape we can visualise how the surface will move in order to reduce the cost function.

The Hadamard structure for the first few iterations is shown in figure 3.7. The largest reduction in dissipation occurs when the top and bottom of the cylinder are moved towards the centre. This compression then causes the shape to elongate due to the area constraint. The objective function is most sensitive to changes in regions at the front of the shape and where the shape protrudes into the flow. Viscous dissipation appears to be relatively insensitive to the geometry at the rear of the shape as would be expected for a bluff body.

Due to the nature of the viscous dissipation problem, issues over the regularity and smoothness of the Hadamard structure are alleviated. Any protrusion of the shape into the



(a) Initial base flow - the wake extends far downstream and a large recirculation bubble is present.



(b) Final base flow - the wake has been reduced and the recirculation bubble eliminated.

Fig. 3.6 Comparison of the base flows for the initial and optimised shapes of the viscous dissipation problem.

flow due to a lack of smoothness in the gradient is penalised in the next iteration and causes the shape to be smoothed.

This Hadamard structure provides useful insight for a designer as it immediately highlights the regions of the shape that have most influence on the objective function.

## 3.2 An advection-diffusion problem

### Motivation

The end goal of this PhD is to apply shape optimisation to a cyclone separator in order to ensure stable flow without compromising the separation of dust particles. Potential dust particle models may be based on a scalar field governed by the advection-diffusion equation. An objective function based on the scalar at the outlet surface might therefore be an effective model.

In order to demonstrate that the shape optimisation tools developed here can be applied to such a problem a model problem involving a temperature field was developed.

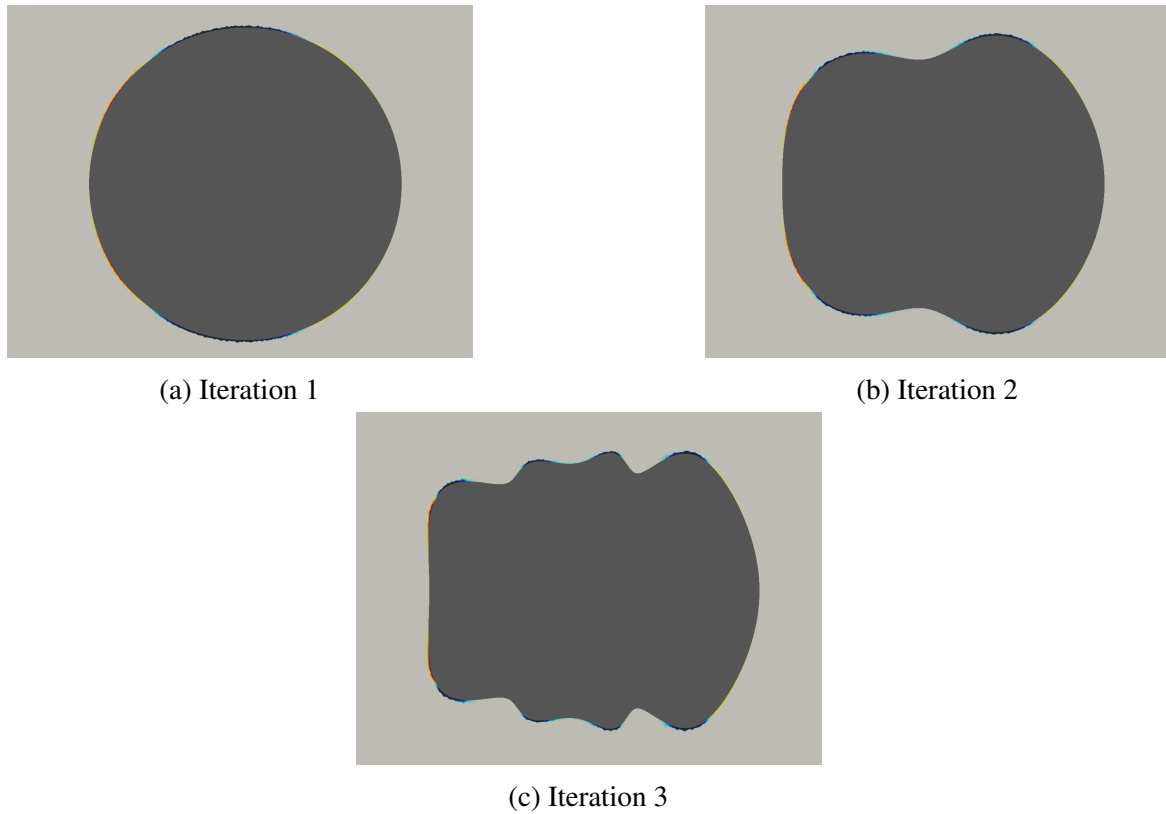


Fig. 3.7 The Hadamard structure on the surface for the first 3 iterations. Red indicates that the objective function will be reduced by displacing the surface out away from the shape in a direction normal to the surface. Blue indicates the objective function will be reduced by an inwards displacement. The brightness of the colour indicates the magnitude of the objective function change

### Problem statement

A pipe section with an expansion is chosen as the domain for this problem and is shown in figure 3.8. As the scalar,  $T$ , is conserved and the problem is applied to the steady state solution we can calculate its ideal value,  $T^*$ , that it would obtain if perfect mixing occurs.

This leads to the objective function:

$$\mathcal{J} = \int_{\Gamma_-} (T - T^*)^2 d\Gamma \quad (3.19)$$

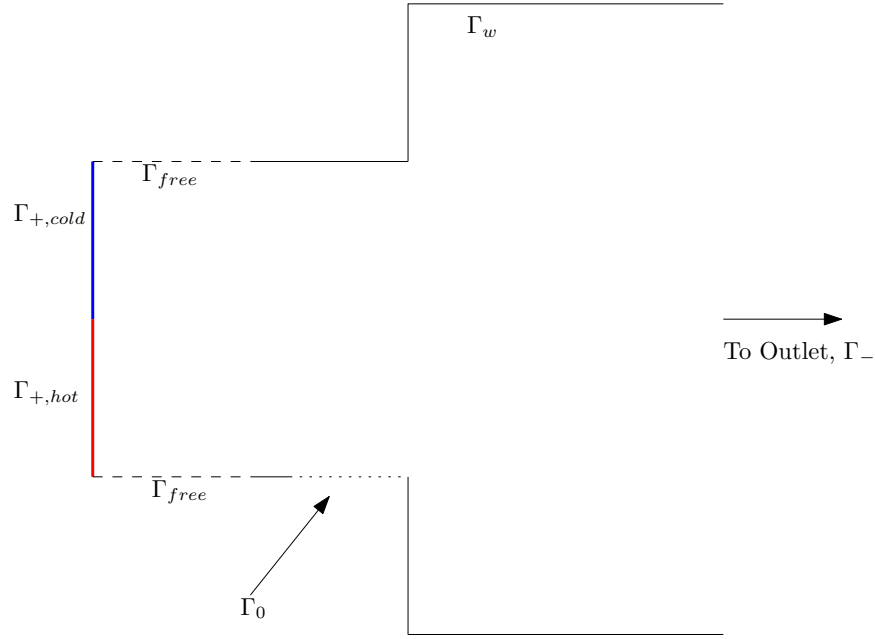


Fig. 3.8 Domain for advection-diffusion optimisation problem. Only the small region  $\Gamma_0$  of the surface is deformed during the optimisation procedure

The problem statement becomes:

$$\text{Find } \min_{\Omega \in \mathcal{D}} \mathcal{J}(\mathbf{u}, \Omega) \quad (3.20)$$

$$\text{subject to } (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} = 0 \text{ in } \Omega \quad (3.21)$$

$$\mathbf{u} \cdot \nabla T - \alpha \nabla^2 T = 0 \text{ in } \Omega \quad (3.22)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \quad (3.23)$$

$$\mathbf{u} = \mathbf{u}_+ \text{ on } \Gamma_+ \quad (3.24)$$

$$p \mathbf{n} - \nu \frac{\partial \mathbf{u}}{\partial n} = 0 \text{ on } \Gamma_- \quad (3.25)$$

$$\mathbf{u} = 0 \text{ on } \Gamma_0 \quad (3.26)$$

$$\frac{\partial u}{\partial y}, v = 0 \text{ on } \Gamma_{free} \quad (3.27)$$

$$T = 0 \text{ on } \Gamma_{+,cold} \quad (3.28)$$

$$T = T_{hot} \text{ on } \Gamma_{+,hot} \quad (3.29)$$

$$\frac{\partial T}{\partial n} = 0 \text{ on } \Gamma_w \cup \Gamma_0 \cup \Gamma_{free} \cup \Gamma_- \quad (3.30)$$

### 3.2.1 Shape parametrisation

In a similar way to the viscous dissipation problem, the surface of  $\Gamma_0$  is given by a Fourier series of the form given in 3.31 where  $x_0$  and  $x_f$  represent the  $x$  coordinates of the start and end of the optimisation region respectively.

$$y = y + 0 + \sum_{k=1}^N a_{2k-1} \sin\left(2\pi k \frac{x_f - x}{x_f - x_0}\right) + a_{2k} \cos\left(2\pi k \frac{x_f - x}{x_f - x_0}\right) \quad (3.31)$$

### 3.2.2 Gradient evaluation and optimisation routine

In order to evaluate the gradient we must introduce an adjoint base-flow  $\boldsymbol{\lambda}$  and an adjoint temperature  $\psi$ . The governing equations for these are given in section A.2.

The resulting expression for the gradient becomes:

$$dJ[\mathbf{V}] = \int_{\Gamma_0} \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \sum_{i=1}^d \frac{\partial u_i}{\partial n} \left\{ \mathbf{v} \frac{\partial \lambda_i}{\partial n} + T \psi n_i \right\} - \int_{\Gamma_0} \alpha \psi \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \frac{\partial^2 T}{\partial n^2} d\Gamma + \int_{\Gamma_0} \alpha \psi \langle \nabla_{\Gamma} T, \nabla_{\Gamma} \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \rangle d\Gamma \quad (3.32)$$

Whilst this expression is a surface integral it is not in true Hadamard form. We therefore cannot draw the Hadamard structure to get plots such as those in 3.7a.

### 3.2.3 Results

The optimisation routine was performed with parametrisations using the first 10 and first 25 harmonics. After only a few iterations the optimisation procedure had to be stopped as base-flow solutions could no longer be found. In these few iterations the objective function is dramatically reduced. When using both the first 25 harmonics of the Fourier series the objective function drops to less than 10% of its original value as seen in figure 3.9.

The base flows and scalar fields for the initial and final shapes are shown for the  $N = 25$  case in figure 3.10.

As the flow used here is laminar, all mixing takes place only by molecular diffusion. This requires large gradients in the scalar field and a large interface between hot and cold regions.

At the inlet the blockage introduced by the optimisation routine causes a large temperature gradient with the cold fluid flowing over the relatively stagnant hot fluid. Forcing the fluid through a small gap generates a series of counter-rotating vortices shown in figure 3.11. The temperature is relatively uniform in these recirculation regions as seen in figure 3.10d. This causes a large temperature gradient between the recirculation region and the fluid flowing

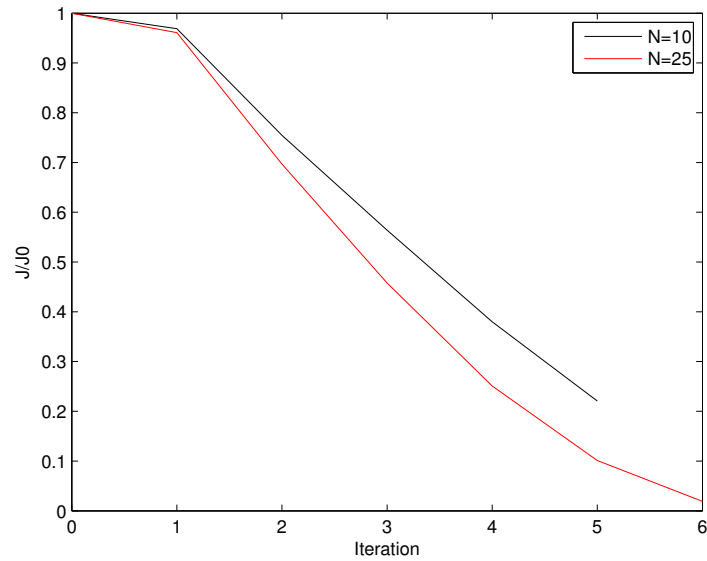


Fig. 3.9 Objective time against iteration for  $N = 10$  and  $N = 25$  harmonics in the surface parametrisation

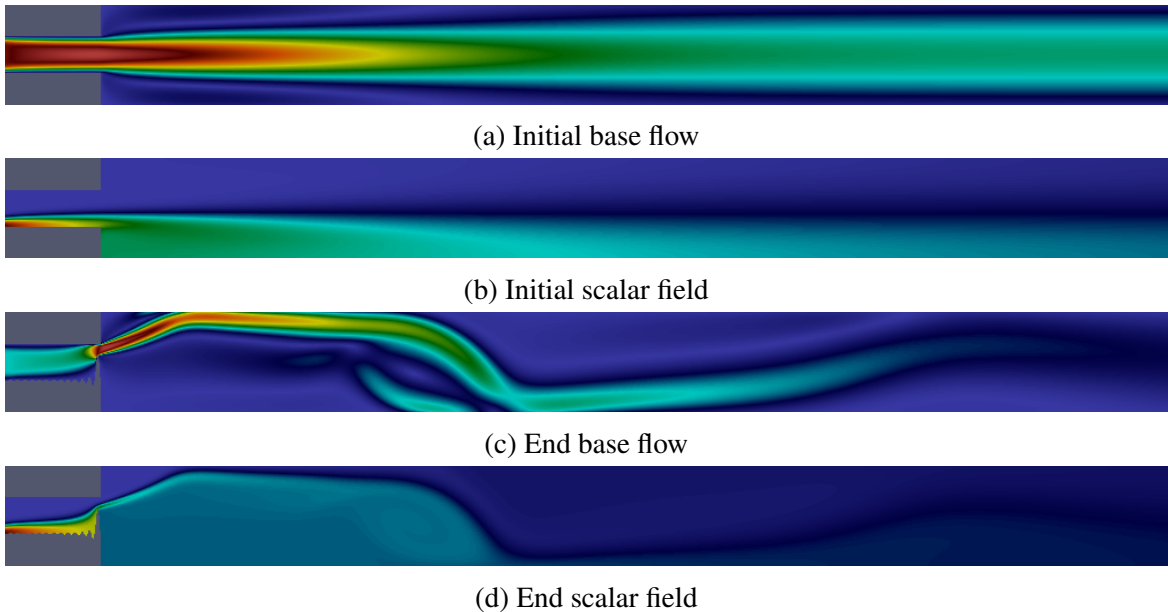


Fig. 3.10 The initial and end base flows and scalar fields for the  $N = 25$  parametrisation

past it. The recirculation regions further increase the mixing by producing a long windy interface with a high surface area between the hot and cold regions of the flow.

As with the viscous dissipation case the Fourier series parametrisation introduces wobbles in the optimised surface. By examining the difference between the final shapes obtained

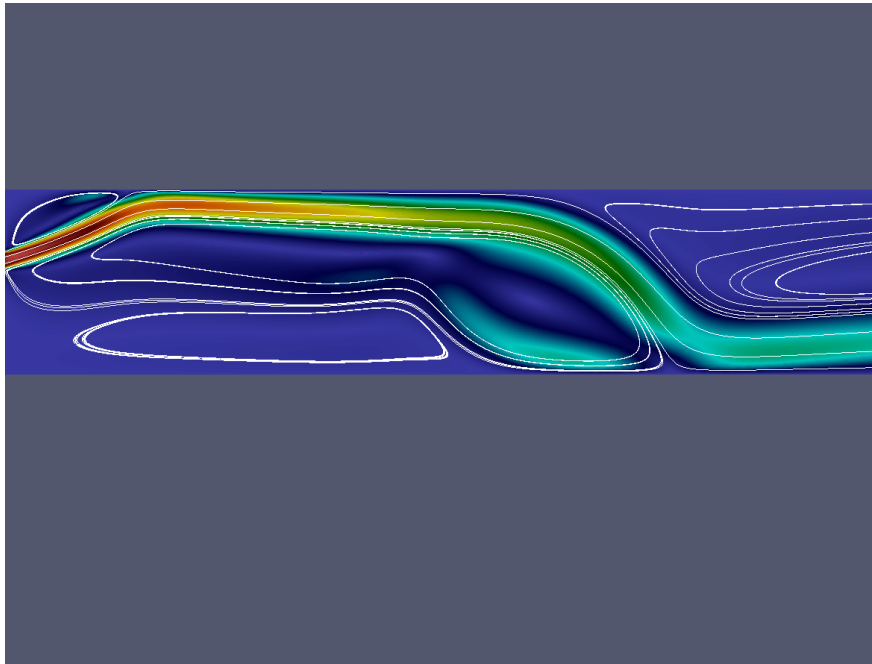
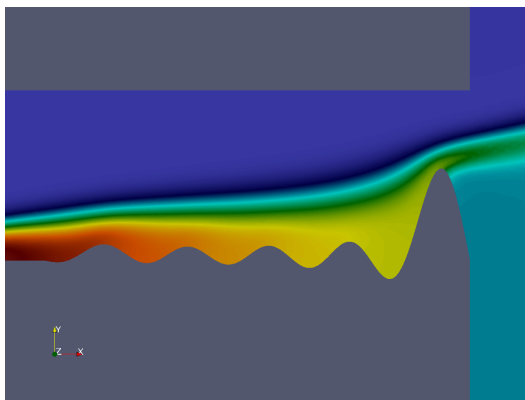
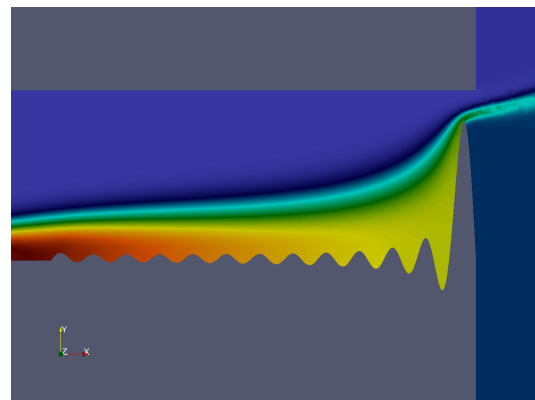


Fig. 3.11 Large counter rotating vortices caused by the emerging jet

using the  $N = 10$  and  $N = 25$  parametrisation we can see that the higher harmonics serve to flatten out the rest of the inlet, figure 3.12. In the limit of the parameter free approach one would expect the surface to be perfectly flat except for the last vertex where there is a step jump which blocks off the inlet.



(a)  $N = 10$



(b)  $N = 25$

Fig. 3.12 Comparison of final shapes for the  $N = 10$  and  $N = 25$  parametrisations of the convected scalar problem

# Chapter 4

## Stability analysis

In order to demonstrate the progress made so far in developing the stability analysis techniques used in this project, linear stability analyses have been performed on the flows past a cylinder and a disk. The example of the disk also allows the demonstration of the axisymmetric code.

### 4.1 Cylinder flow

#### 4.1.1 Motivation

Around a Reynolds number of 50 the steady laminar flow behind a cylinder becomes unstable and the flow starts to oscillate with a series of counter rotating vortices being shed behind the cylinder as seen in 4.1. Linear stability analyses about the steady solution of the Navier-Stokes equations around this Reynolds number show that this base-flow is unstable and provides a good estimate of the frequency at which vortices are shed. Implementing these analyses is an important step in the path of stability optimisation.

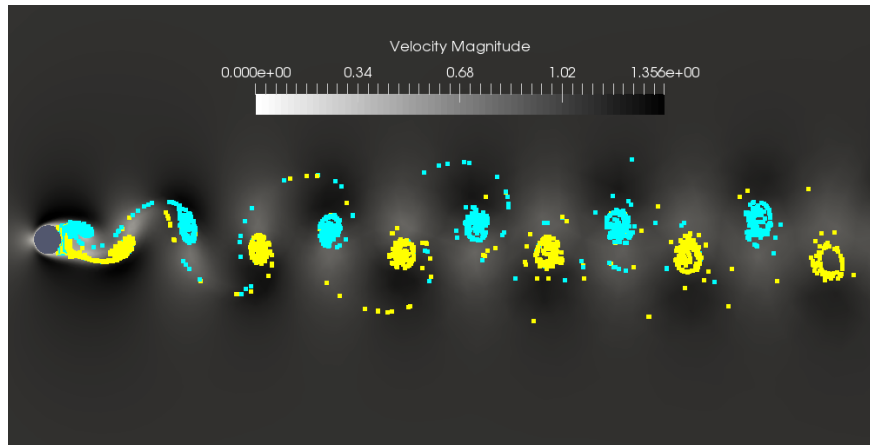


Fig. 4.1 Vortex shedding behind a cylinder from an unsteady solution of Navier-Stokes using a fractional step method derived from Donea et al.[7]. Particles have been injected into the flow to visualise the vortex street with colours indicating the side of the cylinder they originate from

### 4.1.2 Global analysis

The global analysis of section 1.2.1 was performed on the base-flow of the flow over a cylinder using the same domain as the viscous dissipation case, section 3.1. For a Reynolds number of 50 the base-flow is the same as the initial flow for the viscous dissipation problem, figure 3.6a.

The resulting spectrum, figure 4.2 shows that there is one unstable mode with a non-zero frequency. This is to be expected as the vortex shedding that emerges is periodic.

For the unstable global mode the direct and adjoint mode shapes, figure 4.3, were calculated and good agreement was found with [24]. The real and imaginary parts of the direct global mode are in spatial quadrature indicating that the perturbations are advected downstream.

### 4.1.3 Resolvent analysis

The resolvent analysis of section 1.2.2 was performed on the cylinder flow at Reynolds numbers 20 and 60. The first two optimal forcings and the amplitude amplification of the corresponding responses were tracked. The relationship between the amplification,  $\lambda$ , and the frequency of the forcing is shown in 4.4

It can be seen that the amplification due to the first mode completely dwarfs that of the second mode. The correspondence between the peak amplification and the frequency of the unstable global mode is observed as expected. Much like the effect of damping on a

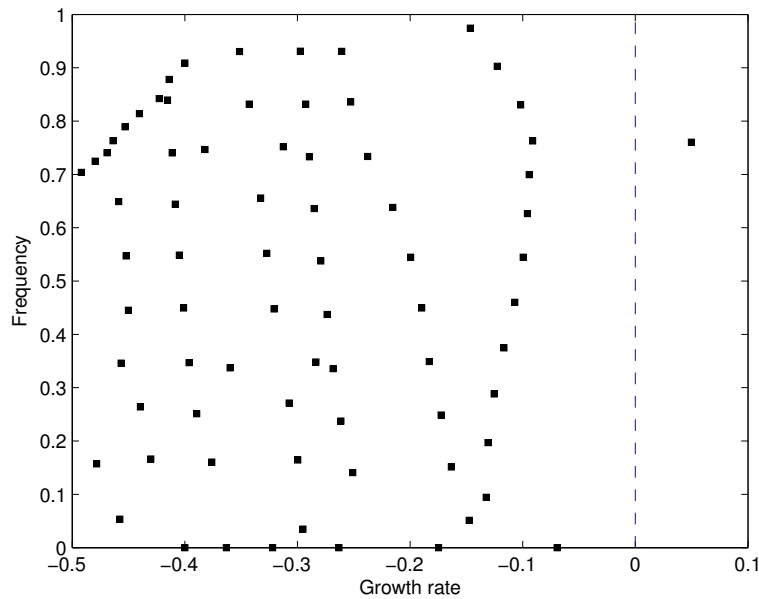


Fig. 4.2 Spectrum of global modes for cylinder flow at  $Re = 50$ . The blue line indicates the line of zero growth rate

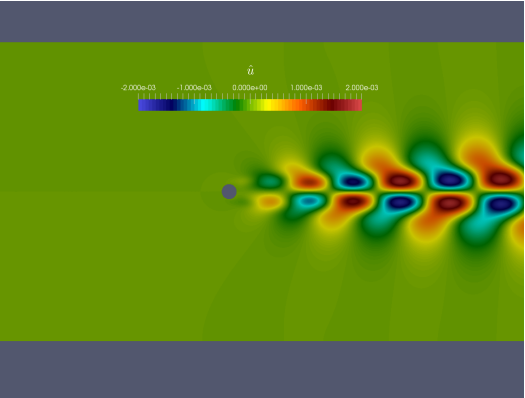
mechanical system, increasing the viscosity causes the amplitude and frequency of the peak response to drop.

### Pseudospectrum

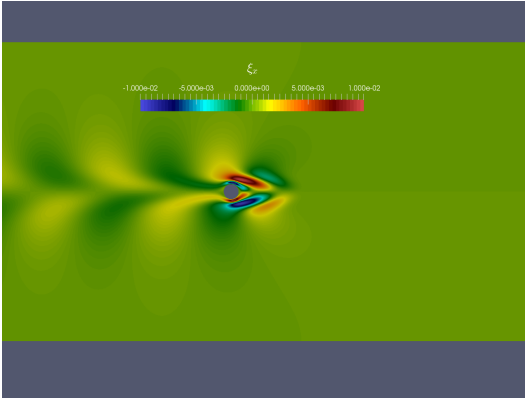
The resolvent analysis can also be used to find the pseudospectrum[46],  $\sigma_\varepsilon$  of the system. The pseudospectrum provides a measure of how sensitive the eigenvalues are to perturbations in the matrix representing the linear operator. If a perturbation matrix with norm  $\varepsilon$  is added to the system then the contour of level  $\varepsilon$  will contain all possible perturbed eigenvalues.

The pseudospectrum for the  $Re = 60$  case is shown in figure 4.5. It can be seen that the stable global modes are enclosed by contours of low  $\varepsilon$  and so are very sensitive to perturbations. This would suggest that they are not physically meaningful as slight changes in the discretisation will result in significant changes in these eigenvalues.

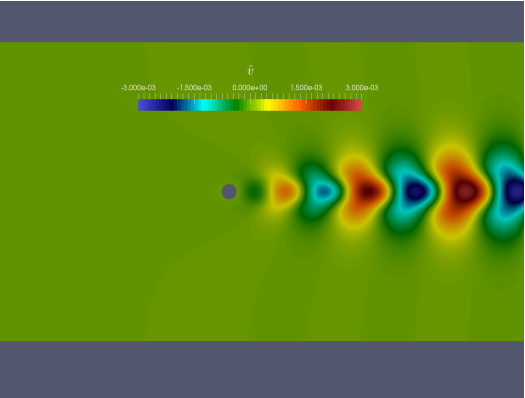
Once a discretisation has been applied and provided that the number of elements and their connectivity is unchanged, shape optimisation can be thought of as simply applying such a perturbation matrix to the original system. The contours of the pseudospectra will therefore indicate how difficult it is to alter the eigenvalues. Finding pseudospectra is very computationally expensive and so the use of the pseudospectrum to diagnose why an eigenvalue may be difficult to change would likely be a step of last resort.



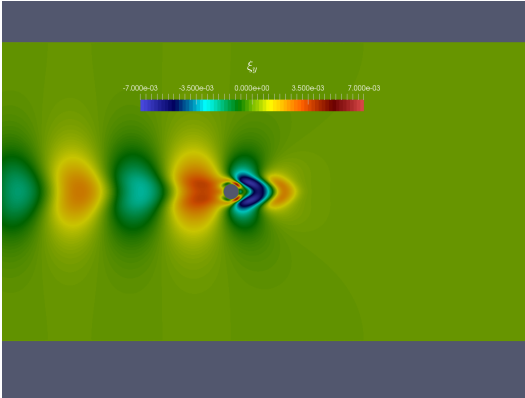
(a) Direct mode x component



(b) Adjoint mode x component



(c) Direct mode y component



(d) Adjoint mode y component

Fig. 4.3 Direct and adjoint modes for a cylinder at  $Re = 50$

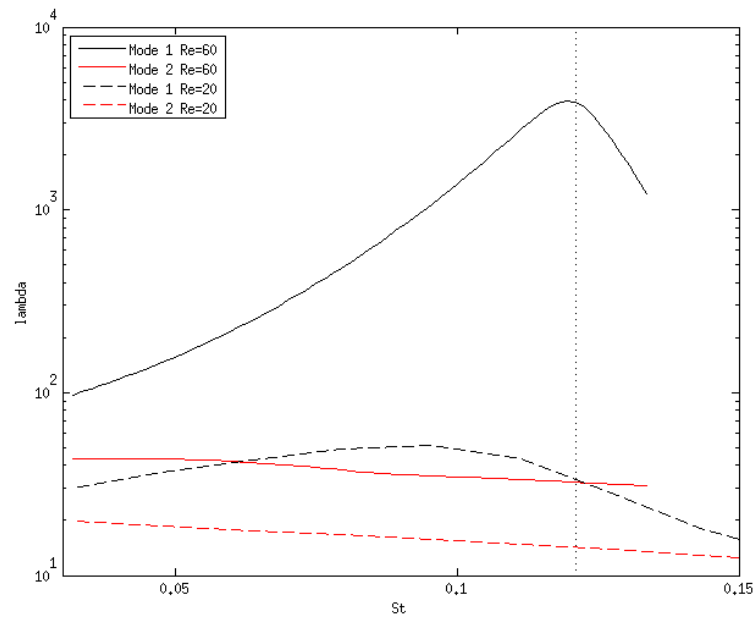


Fig. 4.4 Resolvent analysis for cylinder flows at  $Re = 20$  and  $Re = 60$ . The dotted line indicates the frequency of the unstable mode in the  $Re = 60$  base-flow

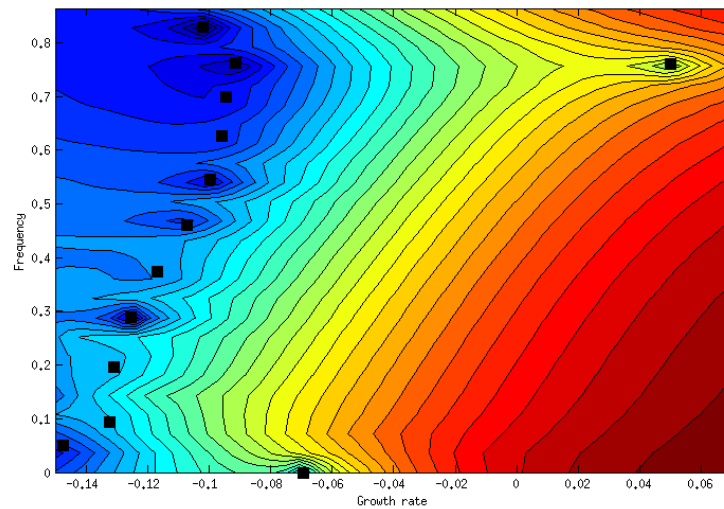


Fig. 4.5 Pseudospectrum for cylinder flow at  $Re = 60$ . Black squares indicate the eigenvalues whilst the colours indicate the value of  $\epsilon$  enclosed by each contour.  $\epsilon$  ranges from  $10^{-2}$  (red) to  $10^{-6}$  (blue).

## 4.2 Axisymmetric disk wake

### 4.2.1 Motivation

The cyclone separator problem is axisymmetric. By taking advantage of this and applying a stability analysis to the axisymmetric Navier-Stokes equations we avoid having to perform a full 3D stability analysis. The stability of the flow past a disk has been investigated by Meliga et al. [25] and so this is a good case to validate the axisymmetric code against.

### 4.2.2 Modal analysis

For an axisymmetric stability analysis we assume that the perturbation is of the form:

$$\mathbf{u}' \rightarrow \hat{\mathbf{u}}(r, z) e^{st} e^{im\theta} \quad (4.1)$$

The eigenvalue problem is then solved at different azimuthal wavenumbers,  $m$ . For the disk wake at  $Re = 120$  with base-flow shown in figure 4.6, unstable modes appear only for the  $m = 1$  azimuthal wavenumber. The spectrum is seen in figure 4.7.

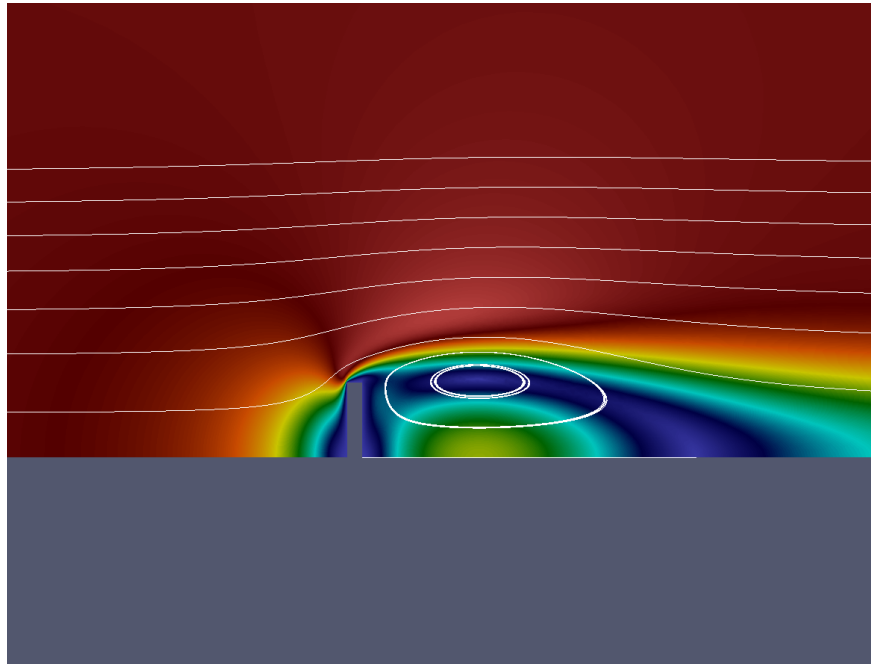


Fig. 4.6 Base-flow for disk wake at  $Re = 120$

There are two modes which cross the marginal stability line at roughly the same Reynolds number. This has an important impact on the resulting flow around the cylinder and has been

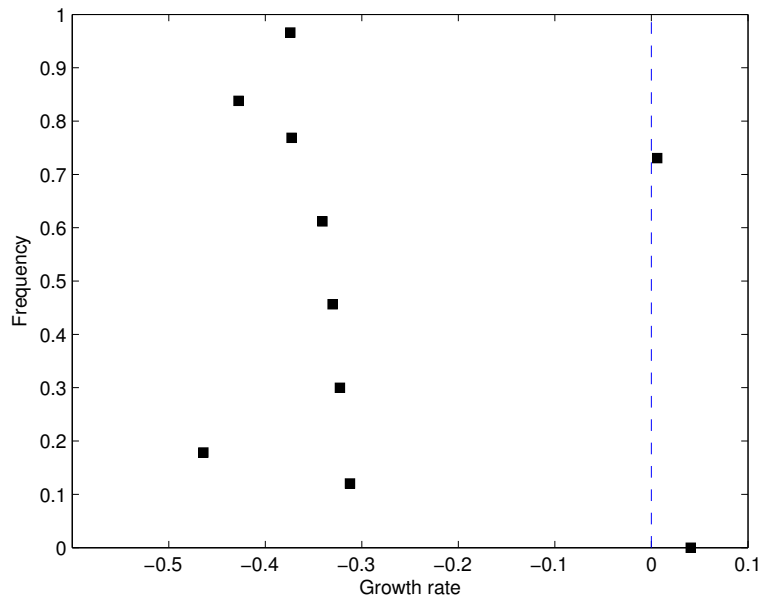
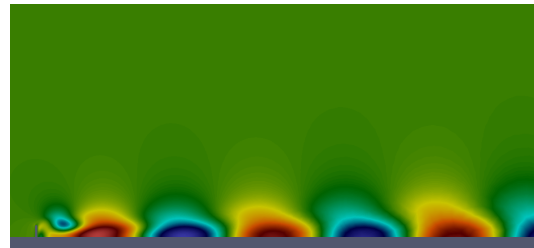
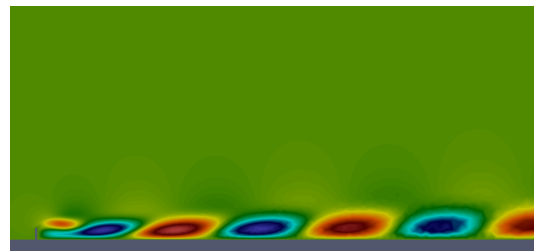
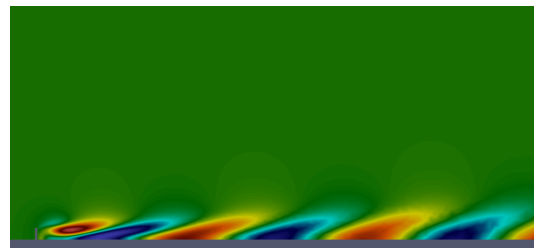


Fig. 4.7 Spectrum for  $m = 1$  azimuthal wavenumber in the disk wake. Two unstable modes are observed: one is steady and the other is unsteady

examined extensively by Meliga et al [25]. One of these modes is steady and represents a deflected wake. The other is spiralling and unsteady.

The mode shapes for the steady and unsteady are given in figure 4.8 and are found to be in good agreement with [25]. This validates the axisymmetric code and stability analysis.

(a) Steady mode  $r$  component(b) Unsteady mode  $r$  component(c) Steady mode  $z$  component(d) Unsteady mode  $z$  component(e) Steady mode  $\theta$  component(f) Unsteady mode  $\theta$  componentFig. 4.8 Mode shapes for the steady and unsteady mode shapes for  $m = 1$

# Chapter 5

## Introducing stability to optimisation

The shape optimisation and stability analysis from the last two chapters are now combined. We first demonstrate the use of shape optimisation in a simple stability problem: finding the aspect ratio of the ellipse that is marginally stable for a given Reynolds number. We then relax the restrictions on the shape and use the same parametrisation that we did for the viscous dissipation example. The problem of gradient regularity is then observed. Finally we demonstrate that the shape optimisation for stability can be easily changed to investigate other stability problems such as control of vortex shedding by rotating cylinders.

### 5.1 Shape optimisation for stability

#### 5.1.1 Motivation

Optimising the shape of the domain to help to ensure that it won't spontaneously transition to a new undesirable state. Additionally, the shape gradient information for the stability problem can be used to enforce a stability constraint via a projection method in much the same way as we enforce the area constraint in section 3.1

#### 5.1.2 Finding an ellipse with marginal stability

The problem statement for a marginal stability problem becomes:

$$\text{Find } \Omega \in \mathcal{D} \quad (5.1)$$

$$\text{such that } \sigma(\Omega) = 0 \quad (5.2)$$

$$\text{subject to: } (\mathbf{u}_b \cdot \nabla) \mathbf{u}_b + \nabla p_b - \nu \nabla^2 \mathbf{u}_b = 0 \text{ in } \Omega \quad (5.3)$$

$$\hat{\mathbf{u}} + (\hat{\mathbf{u}} \cdot \nabla) \mathbf{u}_b + (\mathbf{u}_b \cdot \nabla) \hat{\mathbf{u}} - \nabla \hat{p} + \nu \nabla^2 \hat{\mathbf{u}} = 0 \text{ in } \Omega \quad s \in \mathcal{C} \quad (5.4)$$

$$\mathbf{u}_b = \mathbf{u}_+ \text{ on } \Gamma_+ \quad (5.5)$$

$$p_b \mathbf{n} - \nu \frac{\partial \mathbf{u}_b}{\partial n} = 0 \text{ on } \Gamma_- \quad (5.6)$$

$$\mathbf{u}_b = 0 \text{ on } \Gamma_0 \quad (5.7)$$

$$\frac{\partial u_b}{\partial y}, \nu = 0 \text{ on } \Gamma_{free} \quad (5.8)$$

$$\hat{\mathbf{u}} = 0_+ \text{ on } \Gamma_+ \cup \Gamma_0 \quad (5.9)$$

$$s \hat{p} \mathbf{n} - \nu \frac{\partial \hat{\mathbf{u}}}{\partial n} = 0 \text{ on } \Gamma_- \quad (5.10)$$

$$\frac{\partial \hat{\mathbf{u}}}{\partial y}, \hat{\nu} = 0 \text{ on } \Gamma_{free} \quad (5.11)$$

### Gradient evaluation and optimisation routine

Reducing the set of admissible shapes to those consisting purely of ellipse with unit thickness reduces the problem to a single control variable:  $w$ , the length of the major-axis. This means there is only one deformation vector,  $V$ , and so  $dJ[V] = \frac{d\sigma}{dw}$ .

As we are attempting to find the  $w$  at which the flow is marginally stable we can now perform a Newton method with, at each iteration,  $i$

$$w_{i+1} = - \frac{\sigma(w_i)}{\left. \frac{d\sigma}{dw} \right|_{w_i}} \quad (5.12)$$

The governing equations for the adjoint base flow  $\boldsymbol{\lambda}$  and the adjoint global mode  $\boldsymbol{\xi}$  have been derived in section A.3.

The expression for the gradient with in the direction of deformation field  $V$  has been found in true Hadamard form and is:

$$dJ[V] = \int_{\Gamma_0} \sum_i^d \langle V, \hat{\mathbf{n}} \rangle \frac{\partial \hat{u}_i}{\partial n} \left\{ \nu \frac{\partial \xi_i^*}{\partial n} \right\} d\Gamma + \int_{\Gamma_0} \sum_{i=1}^d \langle V, \hat{\mathbf{n}} \rangle \frac{\partial u_{b,i}}{\partial n} \left[ \nu \frac{\partial \lambda_i}{\partial n} \right] d\Gamma \quad (5.13)$$

## Results

The change in shape and movement within the spectrum is shown in figure 5.1.

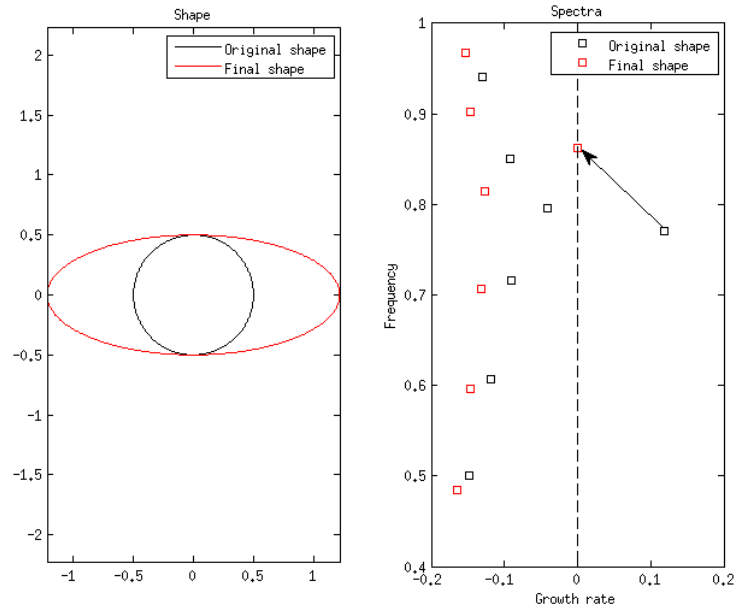


Fig. 5.1 On the left hand side the original and final, marginally stable shapes are shown. The left hand side shows the spectra of the two shapes with the motion of the leading direct global mode highlighted

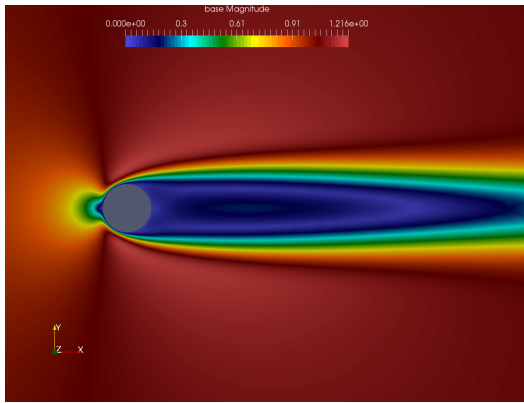
By stretching the cylinder the strength of the shear layers is reduced, preventing them from coupling and behaving as an oscillator.

## 5.2 The problem of regularity

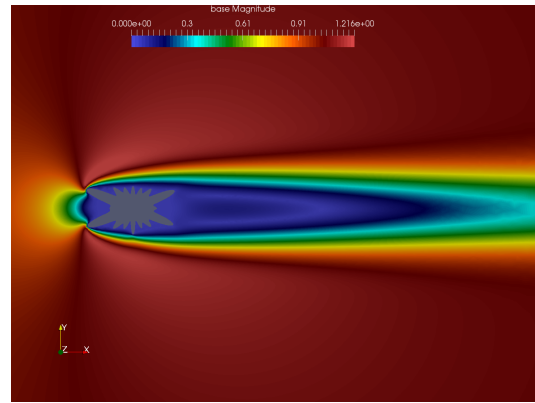
We now look at a growth rate minimisation problem using the steepest descent method and with the less constrained parametrisation of the radial Fourier series from section 3.1.3.

The problem of the shape gradient being not as smooth as the surface can now be seen in figure 5.2. Whilst the objective function is smaller for the end shape, the lower regularity of the shape gradient causes it to lose smoothness. This has an effect on the convergence as well, figure 5.2c, where the minimisation of the objective function is interspersed with sudden spikes.

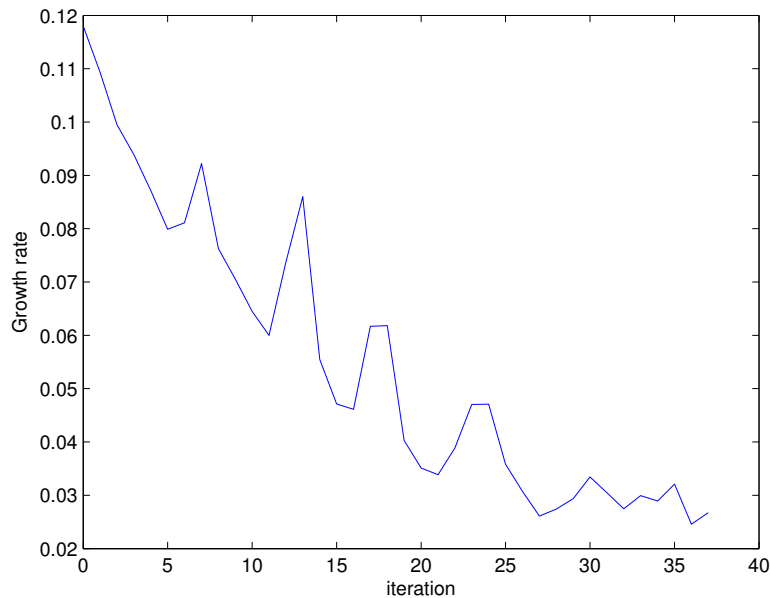
As the form of the shape gradient for this problem has been expressed in a Hadamard form we can plot the gradient structure on the surface as shown in figure 5.3.



(a) Initial shape



(b) End shape



(c) Objective function convergence

Fig. 5.2 A lack of regularity of the shape gradient causes the final shape to lose smoothness. The effect of this on the convergence of the objective function can be seen

When compared with the surface gradient for the same initial shape in the viscous dissipation problem, figure 3.7a, several immediate differences are seen. For both cases, moving the surface outwards or inwards has same effect on the objective function. The relative importance of each part of the surface differs significantly however. For the viscous dissipation case the initial cylinder has a very strong front-back asymmetry where changes at the rear of the cylinder produce little effect on the objective function. For the stability case, however, both the front and back of the cylinder produce a similar level of change in the objective function.

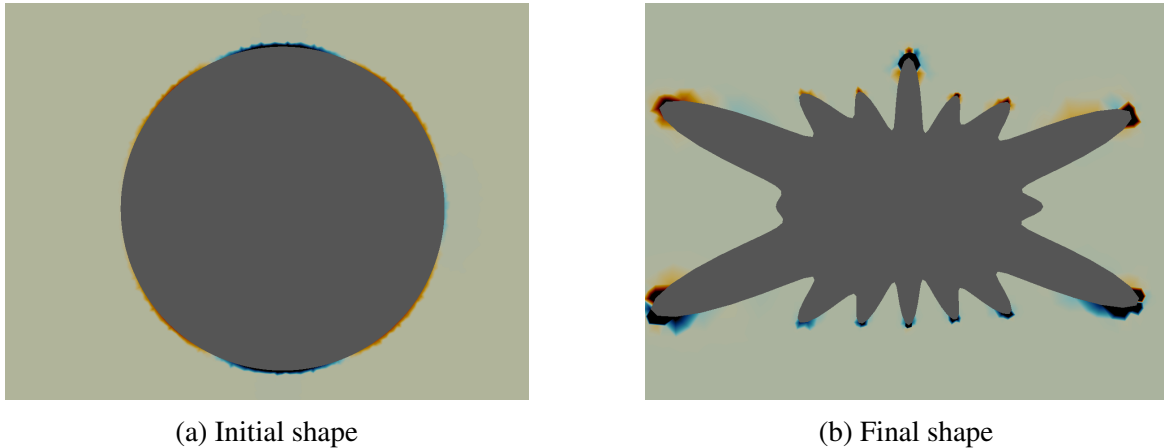


Fig. 5.3 The surface gradient structure for the initial and final shape

As the optimisation procedure continues and the shape becomes spikier the Hadamard structure is dominated more and more by its values at the tips of the spikes. With the viscous dissipation problem any increase in spikiness would increase the objective function and subsequent iterations would try and smooth the shape. For the stability case this feedback loop is no longer present. Subsequent shape changes tend to increase the spikiness of the shape and so, even though the objective function on the whole decreases, the smoothness of the shape decreases rapidly. This causes difficulties in finding a base-flow and problems with the convergence.

## 5.3 Rotating cylinder

### 5.3.1 Motivation

With only a small modification to the code the tools developed here can be applied to other optimisation problems where perturbation growth rate is an objective.

Active means for suppressing the onset of vortex shedding behind a cylinder has had a long history in the literature. These range from actively blowing or sucking the flow on the surface, the use of a control cylinder and even rotating cylinders.

The literature for rotating cylinders typically finds the minimum non dimensional rotation rate  $\alpha = \frac{\Omega a}{U}$  that suppresses vortex shedding either via a direct search method for minimisation or by using gradient information for an unsteady objective function whose minimum was assumed to coincide with a non-oscillating flow. The first approach is slow as it does not use any of the gradient information available. The second approach will converge quickly to a solution due to the gradient information found via the adjoint state but is very expensive

computationally as the direct system must be solved forwards in time first before the adjoint system is subsequently solved backwards in time.

The purpose of this problem is to demonstrate the versatile nature of the optimisation code developed and how using gradient information in stability optimisation allows a large reduction of computation time.

### 5.3.2 Alterations to the optimisation code

The adjoint method used for shape optimisation can be quickly adapted to this optimal control method with only a couple of lines being changed in the code. The governing equations of the adjoint states remain the same whilst the form of the gradient expression changes due to new boundary conditions for the sensitivities on the surface of the cylinder:

$$u'_b = \frac{d}{d\Omega}(-\Omega ay) = -y \text{ on } \Gamma_{cyl} \quad (5.14)$$

$$v'_b = \frac{d}{d\Omega}(\Omega ax) = x \text{ on } \Gamma_{cyl} \quad (5.15)$$

$$\hat{\mathbf{u}}' = 0 \text{ on } \Gamma_{cyl} \quad (5.16)$$

The new sensitivity boundary condition for the unstable mode means that the adjoint mode no longer features in the gradient expression and so no longer needs evaluating. The gradient expression is given in equation 5.17.

$$d\mathcal{J} = - \int_{\Gamma_{cyl}} \left\langle \lambda_p \hat{\mathbf{n}} + \mathbf{v} \frac{\partial \lambda}{\partial n}, \begin{pmatrix} -y \\ x \end{pmatrix} \right\rangle d\Gamma \quad (5.17)$$

### 5.3.3 Results

The code was implemented and the marginal stability curve showing the critical value of alpha for each Reynolds number was plotted, figure 5.5. The base-flow (see figure ) and angular velocity,  $\alpha_{crit}$ , found from each Reynolds number were used as the initial guesses in the base-flow solver and optimisation code.

In the paper by Homescu et al. [13] the calculation of  $\alpha_{crit}$  for a single Reynolds number takes about two hours. The unoptimised single processor code used to find  $\alpha_{crit}$  here took 2.5 hours to find data points for 12 different Reynolds numbers whilst using a larger mesh with higher resolution. This demonstrates the large computational savings obtainable with this method.

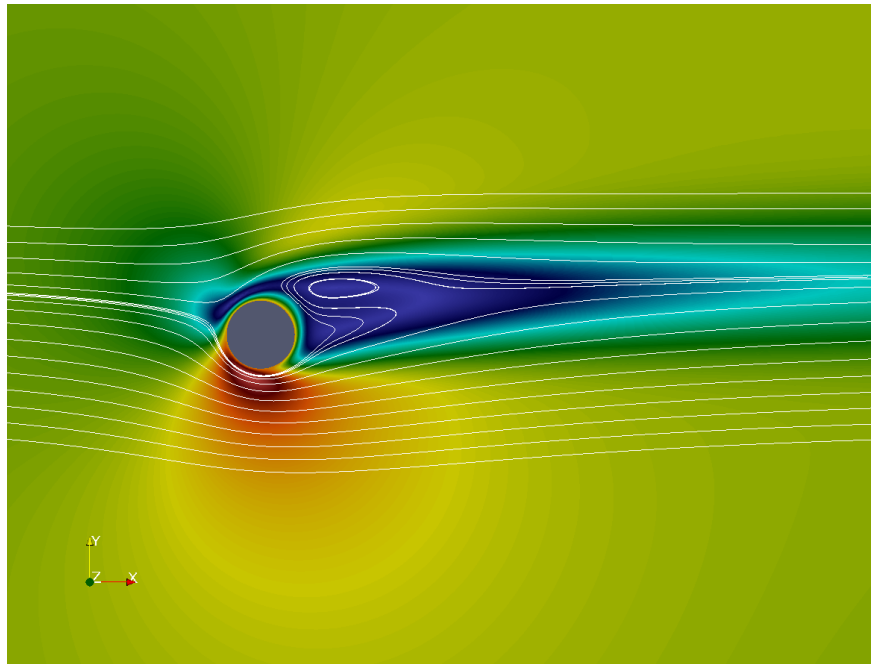
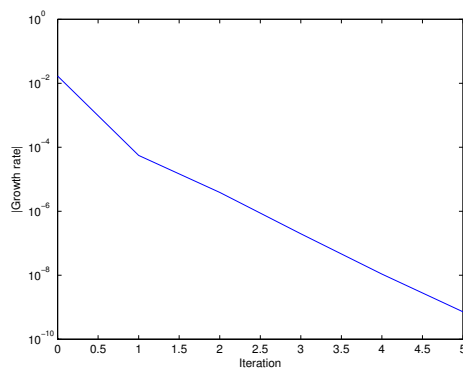
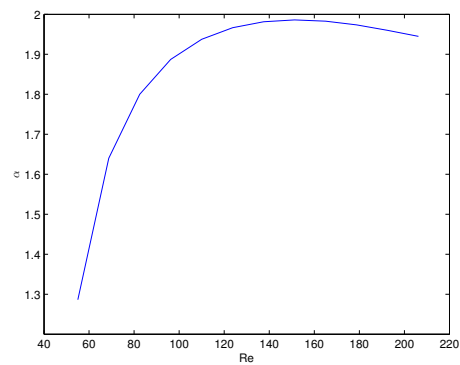


Fig. 5.4 Base-flow for rotating cylinder at  $Re = 100$



(a) Typical convergence for  $\alpha_{crit}$



(b)  $\alpha_{crit}$  for  $60 \leq Re \leq 250$

Fig. 5.5 The marginal stability curve for a rotating cylinder for a range of Reynolds numbers. A typical convergence plot for  $\alpha_{crit}$  is shown to demonstrate the rapid convergence possible when using gradient information

# Chapter 6

## Future Work and Research Plan

### 6.1 Key work

The main underlying methods and analyses required for the optimisation of the cyclone separator have now been implemented. To move to the target flow configuration the following work must be performed:

1. Introduction of swirl
2. Separation of metrics
3. Turbulence modelling
4. Shape Hessian information

#### 6.1.1 Swirl

The cyclone separator uses a highly swirling flow in order to separate out particles. Carried by their own inertia, the particles spiral outwards before colliding with the walls, losing their energy and subsequently falling into the collector.

A swirling base-flow is therefore needed in order to optimise and perform stability analyses about. This will be introduced to the axisymmetric code developed for section 4.2.

#### 6.1.2 Separation metric

Once swirl has been introduced to the base-flow then a separation metric can be introduced. This allows us to use separation performance as a constraint during the stability optimisation.

A number of potential models are available ranging from empirical models, such as the lambda model provided by Dyson, based on flow properties within the volume, all the way through to scalar fields representing the concentration of different sized dust particles. Part of the PhD work arranged with Dyson will involve going to their site and comparing the separation models developed with empirical measurements.

The performance of the separator is an engineering compromise between stability, separation performance, size and pressure drop. Multi-objective optimisation techniques or appropriate weightings applied to each objective will need to be introduced in order to produce an acceptable final design.

### 6.1.3 Turbulence modelling

The Reynolds number of a cyclone separator is much higher than the low Reynolds number laminar cases we have considered here. A turbulence model must be introduced and the stability framework be based around the closure model and RANS equations combined instead of just the steady incompressible Navier-Stokes.

For the simple 2D case the Spalart-Allmaras turbulence model [44] has already been implemented with the modifications and solution method suggested by Crivellini et al. [4] albeit using a continuous Galerkin discretisation.

The Spalart-Allmaras code has been tested for a flat plate boundary layer with the results shown in figure 6.1.

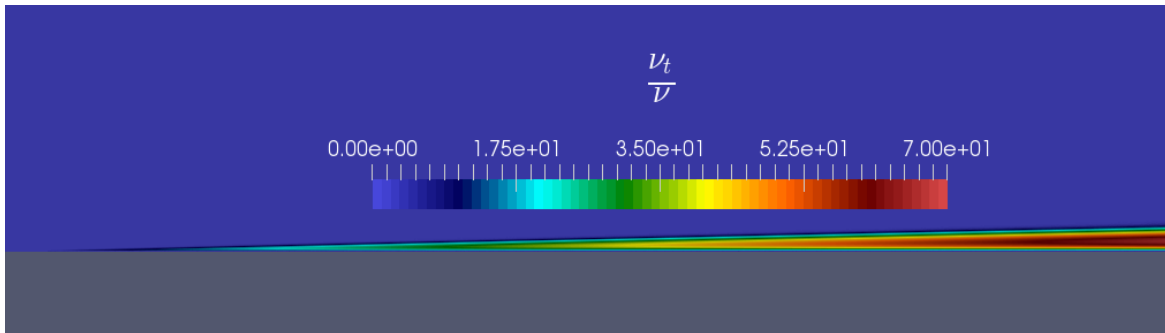
Whilst Spalart-Allmaras works well for simple 2D cases, it is not appropriate for swirling flows. These flows require a full Reynolds-stress model to be implemented. The additional equations required for this model will greatly increase the computational expense and so ensuring the good performance of the code becomes an important priority.

Despite its shortcomings the Spalart-Allmaras model does provide a good test-bed for shape optimisation techniques in conjunction with turbulence models.

## 6.2 Shape Hessian information

The steepest descent method here takes a step of size  $\beta$  down the gradient vector in order to find the minimum. The value of  $\beta$  has so far been chosen by trial and error in order to get good performance over the first few iterations.

More advanced optimisation methods use knowledge of the second derivatives of the system in order to determine the optimal step size. For shape optimisation this information is known as the shape Hessian.



(a) Ratio of turbulent to molecular viscosity

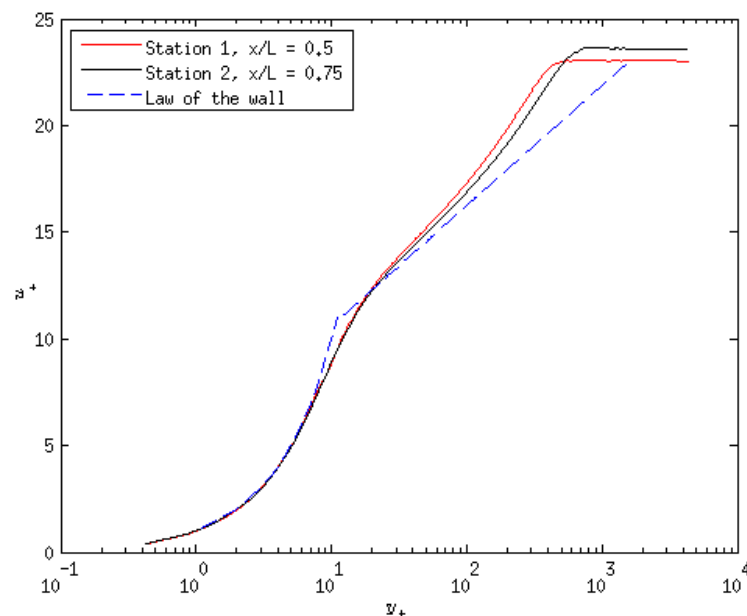
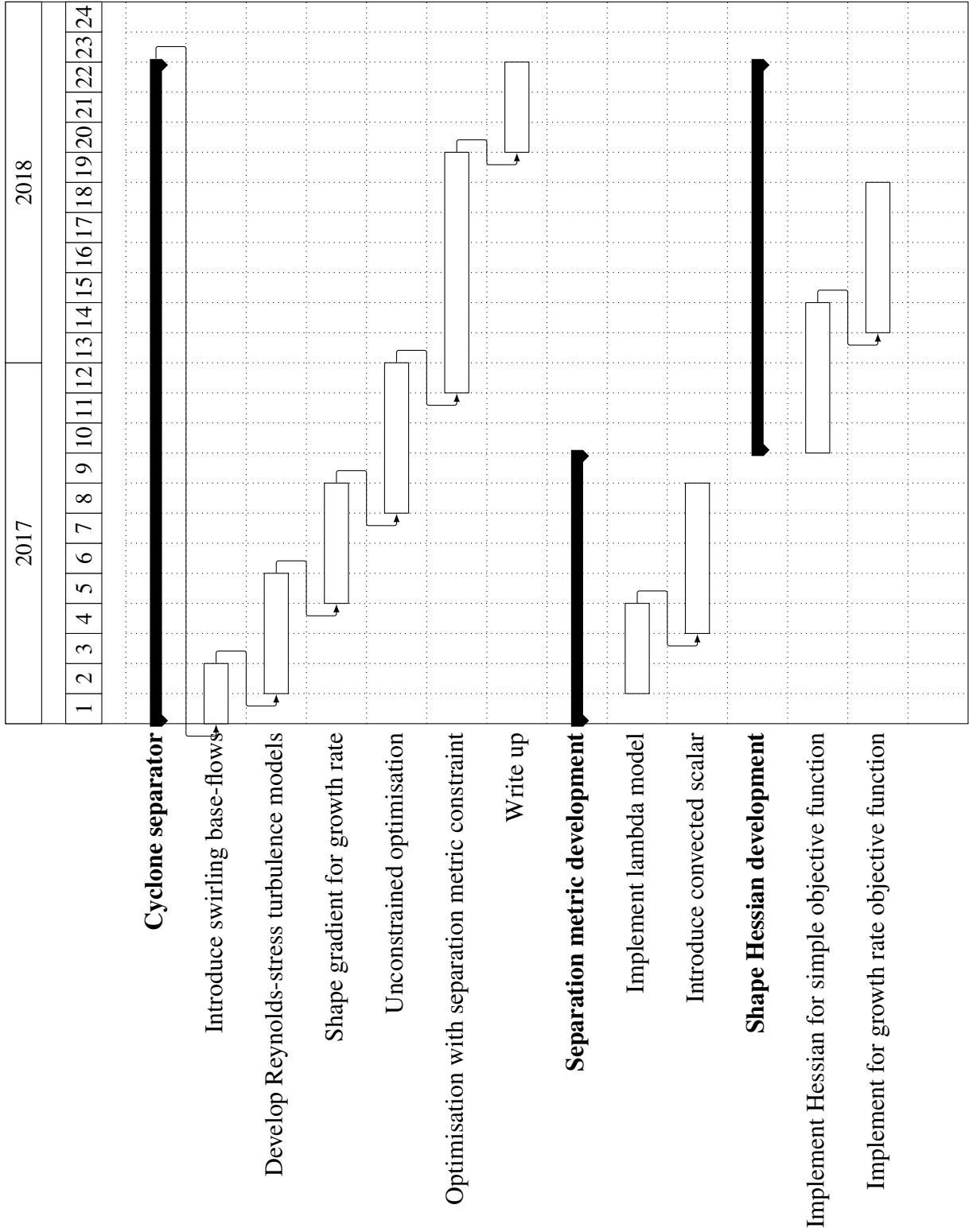
(b) Non-dimensional velocity against non-dimensional wall distance at two different stations. The theoretical law of the wall with  $\kappa = 0.41$  and  $C = 5.0$  is shown for comparison

Fig. 6.1 Results from the Spalart-Allmaras code for a flat plate boundary layer. Plotting the dimensionless velocity profile at two different locations shows the self similar behaviour of the boundary layer is captured. The analytical *law of the wall* agrees well

It has been shown that, like the gradient, the shape Hessian can be expressed in a Hadamard form along the surface of the deforming shape. This means that similar design plots as those in figure 3.7 can be made to aid design before the Hessian is introduced to the optimisation algorithm. The shape Hessian, is in general not symmetric, and so optimisation methods which can deal with an asymmetric Hessian need to be implemented.



# References

- [1] Arian, E. and M., S. (1999). Admitting the Inadmissible: Adjoint Formulation for Incomplete Cost Functionals in Aerodynamic Optimization . *AIAA Journal*, 37(1):37–44.
- [2] Azegami, H., Shimoda, M., Katamine, E., and Wu, Z. C. (1995). A Domain Optimization Technique for Elliptic Boundary Value Problems. *In Computer Aided Optimization Design of Structures IV, Structural Optimization*, pages 51–58.
- [3] Cossu, C. (2014). An Introduction to Optimal Control Lecture notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Appl. Mech. Rev.*, 66(2):021001.
- [4] Crivellini, A., D’Alessandro, V., and Bassi, F. (2013). A Spalart-Allmaras turbulence model implementation in a discontinuous Galerkin solver for incompressible flows. *Journal of Computational Physics*, 241(May 2016):388–415.
- [5] Delfour, M. and Zolesio, J. (1991). Anatomy of the Shape Hessian. *Annali di Matematica pura ed applicata*, CLIX(Iv):315–339.
- [6] Delfour, M. and Zolesio, J. (2001). *Shapes and geometries: analysis, differential calculus, and optimization*. SIAM.
- [7] Donea, J., Giuliani, S., Laval, H., and Quartapelle, L. (1982). Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. *Computer Methods in Applied Mechanics and Engineering*, 30(1):53–73.
- [8] Grinfeld, P. (2010). *Hadamard’s Formula Inside and Out*, volume 146.
- [9] Grinfeld, P. (2013). *Introduction to tensor analysis and the calculus of moving surfaces*. Springer.
- [10] Heuveline, V. and Strauß, F. (2009). Shape optimization towards stability in constrained hydrodynamic systems. *Journal of Computational Physics*, 228(4):938–951.
- [11] Hicks, R. M. and Henne, P. A. (1978). Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412.
- [12] Hicks, R. M., Murman, E. M., and Vanderplaats, G. N. (1974). An assessment of airfoil design by numerical optimization.
- [13] Homescu, C., Navon, I. M., and Li, Z. (2002). Suppression of vortex shedding for flow around a circular cylinder using optimal control. *Int. J. Numer. Meth. Fluids*, 38(March 2001):43–69.

- [14] Jameson, a. (1988). Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260.
- [15] Jameson, a. (1990). Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag. *Proceedings of the 31st Israel Annual Conference on Aviation and Aeronautics, Tel Aviv*, pages 5–17.
- [16] Jameson, A. and Kim, S. (2003). Reduction of the Adjoint Gradient in the Continuous Limit. *Aiaa-2003-0040*, i.
- [17] Juniper, M. P., Hanifi, A., and Theofilis, V. (2014). Modal Stability Theory. *Applied Mechanics Reviews*, 66(2):021003.
- [18] Kavvadias, I., Papoutsis-Kiachagias, E., and Giannakoglou, K. (2015). On the proper treatment of grid sensitivities in continuous adjoint methods for shape optimization. *Journal of Computational Physics*, 301:1–18.
- [19] Lewis, R. M. (1997). Numerical Computation of Sensitivities and the Adjoint Approach Institute for Computer Applications in Science and Engineering Operated by Universities Space Research Association. *Engineering*, (97).
- [20] Lighthill, M. J. (1945). *A New Method of Two-dimensional Aerodynamic Design*.
- [21] Lions, J. L. (1971). *Optimal control of systems governed by partial differential equations*. Springer-Verlag Berlin Heidelberg.
- [22] Logg, A., Mardal, K.-A., Wells, G., and Al., E. (2012). *Automated Solution of Differential Equations by the Finite Element Method*. Springer.
- [23] Lyness, J. and Moler, C. (1967). Numerical differentiation of analytic functions. *SIAM J. Numer. Anal.*, 4:202–210.
- [24] MARQUET, O., SIPP, D., and JACQUIN, L. (2008). Sensitivity analysis and passive control of cylinder flow. *Journal of Fluid Mechanics*, 615:221.
- [25] Meliga, P., Chomaz, J.-M., and Sipp, D. (2009). Global mode interaction and pattern selection in the wake of a disk: a weakly nonlinear expansion. *Journal of Fluid Mechanics*, 633:159.
- [26] Mohammadi, B., Pironneau, O., Mohammadi, B., and Pironneau, O. (2001). *Applied shape optimization for fluids*, volume 28. Oxford Science Publications, 2 edition.
- [27] Mykhaskiv, O., Banovic, M., and Walther, A. (2016). Shape optimisation with differentiated CAD-kernel for U-bend testcase. pages 1–2.
- [28] Nadarajah, S. and Jameson, A. (2000). A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization. *38th AIAA Aerospace Sciences Meeting and Exhibit*, (c).
- [29] Nadarajah, S. K. and Jameson, A. (2007). Optimum Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjoint Method. *AIAA Journal*, 45(7):1478–1491.

- [30] Nakazawa, T. and Azegami, H. (2015). Shape optimization of flow field improving hydrodynamic stability. *Japan Journal of Industrial and Applied Mathematics*.
- [31] Nielsen, E. J., Park, M. A., Introduction, I., Modeling, C., and Branch, S. (2005). Mesh Sensitivities in Computational Design. *Simulation*, (January):1–10.
- [32] Papadimitriou, D. I. and Giannakoglou, K. C. (2007). A continuous adjoint method with objective function derivatives based on boundary integrals, for inviscid and viscous flows. *Computers and Fluids*, 36(2):325–341.
- [33] Papoutsis-Kiachagias, E. M. and Giannakoglou, K. C. (2014). Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications. *Archives of Computational Methods in Engineering*, pages 255–299.
- [34] Pironneau, O. (1973). On optimum profiles in Stokes flow. *Journal of Fluid Mechanics*, 59:117.
- [35] Reuther, J. and Jameson, A. (1995). Aerodynamic shape optimization of wing and wing-body configurations using control theory.
- [36] Schillings, C., Schmidt, S., and Schulz, V. (2011). Efficient shape optimization for certain and uncertain aerodynamic design. *Computers and Fluids*, 46(1):78–87.
- [37] Schmid, P. J. and Brandt, L. (2014). Analysis of Fluid Systems: Stability, Receptivity, Sensitivity Lecture notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Applied Mechanics Reviews*, 66(2):021003.
- [38] Schmidt, S. (2010). Efficient Large Scale Aerodynamic Design Based on Shape Calculus.
- [39] Schmidt, S. and Schultz, V. (2009). Priority Program 1253. *Optimization with Partial Differential Equations*, (February):1–20.
- [40] Schmidt, S. and Schulz, V. (2009). Impulse response approximations of discrete shape Hessians with application in cfd. *Society*, 48(4):2562–2580.
- [41] Schramm, M., Stoevesandt, B., Peinke, J., Optimization, C., En, W., and Turbine, W. (2014). Adjoint Optimization of 2D-Airfoils in Incompressible Flows. (Wccm Xi):1–12.
- [42] Shintani, K. and Azegami, H. (2013). Shape Optimization for Brake Squeal. 2(780):1–10.
- [43] Sipp, D. and Marquet, O. (2012). Characterization of noise amplifiers with global singular modes: the case of the leading-edge flat-plate boundary layer. *Theoretical and Computational Fluid Dynamics*, pages 1–19.
- [44] Spalart, P. R. and Allmaras, S. R. (1994). A one-equation turbulence model for aerodynamic flows. *La Recherche Aeronautique*, 1(1):5–21.
- [45] Strauss, F., Desideri, J.-a., Heuveline, V., Strauss, F., Desideri, J.-a., and Heuveline, V. (2008). multiobjective optimization in hydrodynamic stability control To cite this version : Multiobjective optimization in hydrodynamic stability control.

- 
- [46] Trefethen, L. and Embree, M. (2005). *Spectra and pseudospectra: the behaviour of nonnormal matrices and operators*. Princeton University Press.
- [47] Yu, G., Müller, J. D., Jones, D., and Christakopoulos, F. (2011). CAD-based shape optimisation using adjoint sensitivities. *Computers and Fluids*, 46(1):512–516.
- [48] Zymaris, A. S., Papadimitriou, D. I., Giannakoglou, K. C., and Othmer, C. (2009). Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows. *Computers and Fluids*, 38(8):1528–1538.
- [49] Zymaris, A. S., Papadimitriou, D. I., Giannakoglou, K. C., and Othmer, C. (2010). Adjoint wall functions: A new concept for use in aerodynamic shape optimization. *Journal of Computational Physics*, 229(13):5228–5245.

# Appendix A

## Derivation of state equations for the featured optimisation problems

This appendix derives the strong form of the state equations governing the continuous adjoint for the problems presented in this report.

For all sections in this appendix the general problem statement is that of equation 1.1

### A.1 Viscous dissipation in the flow

The objective function is:

$$J = \int_{\Omega} \sum_{i,j=1}^2 \left( \frac{\partial u_i}{\partial x_j} \right)^2 d\Omega \quad (\text{A.1})$$

The incompressible Navier-Stokes equations form the governing state equations:

$$\begin{aligned} (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} &= 0 \text{ in } \Omega \\ \nabla \cdot \mathbf{u} &= 0 \text{ in } \Omega \end{aligned}$$

With corresponding boundary conditions:

$$\begin{aligned} \mathbf{u} &= 0 \text{ on } \Gamma_w \\ \mathbf{u} &= \mathbf{u}_{in} \text{ on } \Gamma_+ \\ \frac{\partial u}{\partial y}, \nu &= 0 \text{ on } \Gamma_{free} \\ p \hat{\mathbf{n}} - \nu \frac{\partial \mathbf{u}}{\partial n} &= 0 \text{ on } \Gamma_- \end{aligned}$$

Taking the shape derivative of the state equation yields the sensitivity equations and boundary conditions with surface motion,  $V$ .

$$(\mathbf{u}' \cdot \nabla) \cdot \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}' + \nabla p' - \nu \nabla^2 \mathbf{u}' = 0 \quad (\text{A.2})$$

$$\nabla \cdot \mathbf{u}' = 0 \quad (\text{A.3})$$

$$\mathbf{u}' = 0 \text{ on } \Gamma_+ \cup \Gamma_w \quad (\text{A.4})$$

$$\frac{\partial u'}{\partial y}, v' = 0 \text{ on } \Gamma_{free} \quad (\text{A.5})$$

$$p' \hat{\mathbf{n}} - \nu \frac{\partial \mathbf{u}'}{\partial n} = 0 \text{ on } \Gamma_- \quad (\text{A.6})$$

$$\mathbf{u}' + \langle V, \hat{\mathbf{n}} \rangle \frac{\partial \mathbf{u}}{\partial n} = 0 \text{ on } \Gamma_0 \quad (\text{A.7})$$

We then take the dot product of the momentum equation with an adjoint base flow,  $\boldsymbol{\lambda}$ , and multiply the continuity equation by the adjoint pressure,  $\lambda_p$ . After integrating over the domain and successive applications of the divergence theorem we obtain:

$$\int_{\Omega} \sum_{i=1}^2 \left[ -\nu \nabla^2 \lambda_i - \frac{\partial \lambda_p}{\partial x_i} + \sum_{j=1}^2 \left[ -u_j \frac{\partial \lambda_i}{\partial x_j} - u_j \frac{\partial \lambda_j}{\partial x_i} \right] \right] u'_i d\Omega \quad (\text{A.8})$$

$$- \int_{\Omega} \sum_{i=1}^2 p' \frac{\partial \lambda_i}{\partial x_i} d\Omega \quad (\text{A.9})$$

$$+ \int_{\Gamma} \sum_{i=1}^2 \left[ \lambda_p n_i + \nu \frac{\partial \lambda_i}{\partial n} + \sum_{j=1}^2 [\lambda_j u_j n_i + \lambda_i u_j n_j] \right] u'_i d\Gamma \quad (\text{A.10})$$

$$+ \int_{\Gamma} \sum_{i=1}^2 \lambda_i n_i p' d\Gamma - \int_{\Gamma} \sum_{i=1}^2 \nu \lambda_i \frac{\partial u'_i}{\partial n} \quad (\text{A.11})$$

$$= 0 \quad (\text{A.12})$$

Taking the shape derivative of  $\mathcal{J}$  yields:

$$d\mathcal{J}[V] = \int_{\Gamma} \langle V, \hat{\mathbf{n}} \rangle \sum_{i,j=1}^2 \nu \left( \frac{\partial u_i}{\partial x_j} \right)^2 d\Gamma + \int_{\Omega} \sum_{i,j=1}^2 \nu \frac{\partial u_i}{\partial x_j} \frac{\partial u'_i}{\partial x_j} d\Omega \quad (\text{A.13})$$

Applying the divergence theorem yields:

$$\begin{aligned}
d\mathcal{J}[V] &= \int_{\Gamma} \langle V, \hat{\mathbf{n}} \rangle \sum_{i,j=1}^2 \nu \left( \frac{\partial u_i}{\partial x_j} \right)^2 d\Gamma \\
&\quad - \int_{\Omega} 2\nu \sum_{i=1}^2 u'_i \nabla^2 u_i d\Omega + \int_{\Gamma} 2\nu \sum_{i=1}^2 u'_i \frac{\partial u_i}{\partial n} d\Gamma
\end{aligned} \tag{A.14}$$

We add equation A.8 to the left hand side of equation A.14. As  $\boldsymbol{\lambda}$  and  $\lambda_p$  have so far been arbitrary functions, we can use them to eliminate the sensitivity terms  $u'_i$  and  $p'$  by choosing them such that they satisfy:

$$\nu \nabla^2 \boldsymbol{\lambda} + \nabla \lambda_p + (\mathbf{u} \cdot \nabla) \boldsymbol{\lambda} + (\nabla \boldsymbol{\lambda})^T \mathbf{u} = 2\nu \nabla^2 \mathbf{u} \tag{A.15}$$

$$\nabla \cdot \boldsymbol{\lambda} = 0 \tag{A.16}$$

$$\boldsymbol{\lambda} = 0 \text{ on } \Gamma_+ \tag{A.17}$$

$$\boldsymbol{\lambda} = 0 \text{ on } \Gamma_w \cup \Gamma_0 \tag{A.18}$$

$$\lambda_p \hat{\mathbf{n}} + \nu \frac{\partial \boldsymbol{\lambda}}{\partial n} + \langle \boldsymbol{\lambda}, \mathbf{u} \rangle \hat{\mathbf{n}} + \langle \hat{\mathbf{n}}, \mathbf{u} \rangle \boldsymbol{\lambda} - 2\nu \frac{\partial \mathbf{u}}{\partial n} = 0 \text{ on } \Gamma_- \tag{A.19}$$

The boundary conditions for the above have been found by splitting the integrals over  $\Gamma$  into integrals over the subsections  $\Gamma_0, \Gamma_w, \Gamma_+, \Gamma_-$  and  $\Gamma_0$ . The boundary conditions of the primal and sensitivity variables are substituted in and the boundary condition of the adjoint state is chosen to cancel out any term involving a sensitivity variable.

For  $\boldsymbol{\lambda}$  and  $\lambda_p$  that satisfy the above, we then obtain the following expression for the gradient  $d\mathcal{J}[V]$  in Hadamard form:

$$d\mathcal{J}[V] = \int_{\Gamma_0} \langle V, \hat{\mathbf{n}} \rangle \sum_{i=1}^2 \left[ -\nu \left( \frac{\partial u_i}{\partial x_j} \right)^2 + \nu \frac{\partial \lambda_i}{\partial n} \frac{\partial u_i}{\partial n} \right] \tag{A.20}$$

The above has made use of the fact that both  $\sum_{i=1}^2 \frac{\partial u_i}{\partial x_j} n_i = \nabla \cdot \mathbf{u}$  and  $\sum_{i,j=1}^2 \left( \frac{\partial u_i}{\partial x_j} \right)^2 = \sum_{i=1}^2 \left( \frac{\partial u_i}{\partial n} \right)^2$  on  $\Gamma_0$ .

## A.2 Mixing of convected scalar on outlet plane

A convected scalar,  $T$  e.g. temperature, has been introduced and the objective function based on a target uniform outlet temperature,  $T^*$ , results in an objective function of:

$$\mathcal{J} = \int_{\Gamma_-} (T - T^*)^2 d\Gamma \quad (\text{A.21})$$

We split the inlet boundary into a 'hot' and 'cold' region,  $\Gamma_+ = \Gamma_{+,hot} \cup \Gamma_{+,cold}$ . The governing equation and boundary conditions for  $T$  are therefore:

$$(\mathbf{u} \cdot \nabla) T - \alpha \nabla^2 T = 0 \quad (\text{A.22})$$

$$T = 0 \text{ on } \Gamma_{+,cold} \quad (\text{A.23})$$

$$T = T_{hot} \text{ on } \Gamma_{+,hot} \quad (\text{A.24})$$

$$\frac{\partial T}{\partial n} = 0 \text{ on } \Gamma_w \cup \Gamma_- \cup \Gamma_0 \quad (\text{A.25})$$

In the above  $\alpha$  is the scalar's diffusivity and  $\mathbf{u}$  satisfies the incompressible Navier-Stokes equations.

Taking the shape derivative of the governing equations and boundary conditions generates the state equations for the sensitivities,  $T'$ .

$$(\mathbf{u}' \cdot \nabla) T + (\mathbf{u} \cdot \nabla) T' - \alpha \nabla^2 T' = 0 \text{ in } \Omega \quad (\text{A.26})$$

$$T' = 0 \text{ on } \Gamma_{+,hot} \cup \Gamma_{+,cold} \quad (\text{A.27})$$

$$\frac{\partial T'}{\partial n} = 0 \text{ on } \Gamma_w \cup \Gamma_w \quad (\text{A.28})$$

$$\frac{\partial T'}{\partial n} + \langle V, \hat{\mathbf{n}} \rangle \frac{\partial^2 T}{\partial n^2} - \langle \nabla_{\Gamma} T, \nabla_{\Gamma} \langle V, \hat{\mathbf{n}} \rangle \rangle = 0 \text{ on } \Gamma_0 \quad (\text{A.29})$$

In the above  $\nabla_{\Gamma}$  is the tangential gradient operator and we define  $\frac{\partial^2 T}{\partial n^2}$  by:

$$\frac{\partial^2 T}{\partial n^2} = \langle \nabla (\nabla T), \hat{\mathbf{n}}, \hat{\mathbf{n}} \rangle$$

The equations governing the sensitivity of the base flow,  $\mathbf{u}'$ , are the same as those in equation A.2.

We take the product of the governing sensitivity equation with the adjoint scalar  $\psi$  and integrate over the whole domain. By applying the divergence theorem we get:

$$\begin{aligned} & \int_{\Omega} -(\mathbf{u}' \cdot \nabla \psi) T + (\mathbf{u} \cdot \nabla \psi) T' - \alpha T' \nabla^2 \psi d\Omega \\ & + \int_{\Gamma} T \psi \langle \mathbf{u}', \hat{\mathbf{n}} \rangle + T' \psi \langle \mathbf{u}, \hat{\mathbf{n}} \rangle - \alpha \frac{\partial T'}{\partial n} \psi + \alpha \frac{\partial \psi}{\partial n} T' d\Gamma = 0 \end{aligned} \quad (\text{A.30})$$

The shape derivative of  $\mathcal{J}$  gives:

$$d\mathcal{J}[\mathbf{V}] = \int_{\Gamma_-} 2(T - T^*) T' d\Gamma \quad (\text{A.31})$$

To the left hand side of equation A.31 we now add equations A.8 and A.30. This gives

$$\begin{aligned} & \int_{\Omega} \left( -\alpha \nabla^2 \psi - \sum_{i=1}^2 u_i \frac{\partial \psi}{\partial x_i} \right) T' d\Omega + \int_{\Omega} \sum_{i=1}^2 \left[ -T \frac{\partial \psi}{\partial x_i} \right] u'_i d\Omega \\ & \int_{\Omega} \sum_{i=1}^d \left[ -v \nabla^2 \lambda_i - \frac{\partial \lambda_p}{\partial x_i} + \sum_{j=1}^d \left\{ -u_j \frac{\partial \lambda_i}{\partial x_j} - u_j \frac{\partial \lambda_j}{\partial x_i} \right\} \right] u'_i d\Omega + \\ & \int_{\Gamma} \left( \psi \sum_{i=1}^d \{u_i n_i\} + \alpha \frac{\partial \psi}{\partial n} \right) T' d\Gamma + \int_{\Gamma} -\alpha \psi \frac{\partial T'}{\partial n} d\Gamma + \int_{\Gamma} \sum_{i=1}^d \{T \psi n_i\} u'_i d\Gamma + \\ & - \int_{\Omega} \sum_{i=1}^d p' \frac{\partial \lambda_i}{\partial x_i} d\Omega + \int_{\Gamma} \sum_{i=1}^d \left[ \lambda_p n_i + v \frac{\partial \lambda_i}{\partial n} + \sum_{j=1}^d \{ \lambda_j u_j n_i + \lambda_i u_j n_j \} \right] u'_i d\Gamma + \\ & + \int_{\Gamma} \sum_{i=1}^d \lambda_i n_i p' d\Gamma - \int_{\Gamma} \sum_{i=1}^d v \lambda_i \frac{\partial u'_i}{\partial n} d\Gamma + \\ & d\mathcal{J}[\mathbf{V}] = \int_{\Gamma_{outlet}} 2(T - T^*) T' d\Gamma \end{aligned} \quad (\text{A.32})$$

We now choose the system of equations that must be satisfied by  $\psi$  and  $\boldsymbol{\lambda}$  in order to eliminate any dependence on the sensitivities.

The governing equations for the adjoint scalar are:

$$(\mathbf{u} \cdot \nabla) \psi + \alpha \nabla^2 \psi = 0 \text{ in } \Omega \quad (\text{A.33})$$

$$\psi = 0 \text{ on } \Gamma_{inlet} \quad (\text{A.34})$$

$$\alpha \frac{\partial \psi}{\partial n} + \psi \langle \mathbf{u}, \hat{\mathbf{n}} \rangle = 0 \text{ on } \Gamma_w \cup \Gamma_0 \quad (\text{A.35})$$

$$\alpha \frac{\partial \psi}{\partial n} + \psi \langle \mathbf{u}, \hat{\mathbf{n}} \rangle = 2(T - T^*) \text{ on } \Gamma_- \quad (\text{A.36})$$

The governing equations for the adjoint scalar remain the same except for a new source term and a change of the outlet boundary condition:

$$v \nabla^2 \boldsymbol{\lambda} + \nabla \lambda_p + (\mathbf{u} \cdot \nabla) \boldsymbol{\lambda} + (\nabla \boldsymbol{\lambda})^T \mathbf{u} = T \nabla \psi \text{ in } \Omega \quad (\text{A.37})$$

$$\lambda_p \hat{\mathbf{n}} + v \frac{\partial \boldsymbol{\lambda}}{\partial n} + \langle \boldsymbol{\lambda}, \mathbf{u} \rangle \hat{\mathbf{n}} + \langle \hat{\mathbf{n}} \mathbf{u} \rangle \boldsymbol{\lambda} = T \psi \hat{\mathbf{n}} \text{ on } \Gamma_- \quad (\text{A.38})$$

The final equation for the shape gradient is given in equation A.39. This is not in complete Hadamard form due to the last term. As all the field integrals have been eliminate and the shape paramaterisation code can generate the tangential gradient of  $\langle \mathbf{V}, \hat{\mathbf{n}} \rangle$  we do not attempt to bring it into full Hadamard form. If a parameter free shape optimisation approach were to be used then this would need to be performed.

$$\begin{aligned} dJ[\mathbf{V}] = & \int_{\Gamma_{opt}} \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \sum_{i=1}^d \frac{\partial u_i}{\partial n} \left\{ v \frac{\partial \lambda_i}{\partial n} + T \psi n_i \right\} - \int_{\Gamma_{opt}} \alpha \psi \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \frac{\partial^2 T}{\partial n^2} d\Gamma \\ & + \int_{\Gamma_{opt}} \alpha \psi \langle \nabla_{\Gamma} T, \nabla_{\Gamma} \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \rangle d\Gamma \end{aligned} \quad (\text{A.39})$$

### A.3 Stability eigenvalue problem

In the previous cases the objective function has been a field or surface integral. For the stability case out objective is now:

$$\mathcal{J} = s \quad s \in \mathbb{C} \quad (\text{A.40})$$

The linear stability eigenvalue problem defines  $s$  and is given by

$$s\hat{\mathbf{u}} + (\hat{\mathbf{u}} \cdot \nabla) \mathbf{u}_b + (\mathbf{u}_b \cdot \nabla) \hat{\mathbf{u}} + \nabla \hat{p} - \nu \nabla^2 \hat{\mathbf{u}} = 0 \text{ in } \Omega \quad (\text{A.41})$$

$$\nabla \cdot \hat{\mathbf{u}} = 0 \text{ in } \Omega \quad (\text{A.42})$$

$$\hat{\mathbf{u}} = 0 \text{ on } \Gamma_+ \cup \Gamma_0 \cup \Gamma_w \quad (\text{A.43})$$

$$\frac{\partial \hat{u}}{\partial y}, \hat{v} = 0 \text{ on } \Gamma_{free} \quad (\text{A.44})$$

$$\hat{p} \hat{\mathbf{n}} - \nu \frac{\partial \hat{\mathbf{u}}}{\partial n} = 0 \text{ on } \Gamma_- \quad (\text{A.45})$$

In above the base-flow  $\mathbf{u}_b$  is governed by the incompressible Navier-Stokes equations. Taking the shape derivative of the governing equations yields the sensitivity equations:

$$s\hat{\mathbf{u}}' + s'\hat{\mathbf{u}} + (\hat{\mathbf{u}}' \cdot \nabla) \mathbf{u}_b + (\hat{\mathbf{u}} \cdot \nabla) \mathbf{u}_b' + (\mathbf{u}_b' \cdot \nabla) \hat{\mathbf{u}} + (\mathbf{u}_b \cdot \nabla) \hat{\mathbf{u}}' + \nabla \hat{p}' - \nu \nabla^2 \hat{\mathbf{u}}' = 0 \text{ in } \Omega \quad (\text{A.46})$$

$$\nabla \cdot \hat{\mathbf{u}}' = 0 \text{ in } \Omega \quad (\text{A.47})$$

$$\hat{\mathbf{u}}' = 0 \text{ on } \Gamma_+ \cup \Gamma_w \quad (\text{A.48})$$

$$\hat{p}' \hat{\mathbf{n}} - \nu \frac{\partial \hat{\mathbf{u}}'}{\partial n} = 0 \text{ on } \Gamma_- \quad (\text{A.49})$$

$$\frac{\partial \hat{u}'}{\partial y}, \hat{v}' = 0 \text{ on } \Gamma_{free} \quad (\text{A.50})$$

$$\hat{\mathbf{u}}' + \langle V, \hat{\mathbf{n}} \rangle \frac{\partial \hat{\mathbf{u}}}{\partial n} = 0 \text{ on } \Gamma_0 \quad (\text{A.51})$$

The sensitivities and primal variables in the above are now complex quantities. We therefore take the complex dot product with the adjoint stability variable,  $\boldsymbol{\xi}$ , and the adjoint stability pressure  $\xi_p$ . Integrating over the whole domain and applying the divergence theorem yields:

$$\begin{aligned}
& \int_{\Omega} \sum_i^d \{\hat{u}_i \xi_i^*\} s' + \sum_i^d \left\{ s \xi_i^* - v \nabla^2 \xi_i^* + \frac{\partial \xi_p^*}{\partial x_i} - \sum_j^d \left[ u_{b,j} \frac{\partial \xi_i^*}{\partial x_j} + u_{b,j} \frac{\partial \xi_j^*}{\partial x_i} \right] \right\} \hat{u}'_i - \sum_{i,j}^d \left\{ \hat{u}_j \frac{\partial \xi_i^*}{\partial x_j} + \hat{u}_j \frac{\partial \xi_j^*}{\partial x_i} \right\} u'_{b,i} d\Omega \\
& \quad - \int_{\Omega} \sum_{i=1}^d \left\{ \frac{\partial \xi_i^*}{\partial x_i} \right\} \hat{p}' d\Omega \\
& \quad + \int_{\Gamma} \sum_i^d \left\{ v \frac{\partial \xi_i^*}{\partial n} + \sum_j^d [u_{b,j} \xi_i^* n_j + u_{b,j} \xi_j^* n_i] - \xi_p^* n_i \right\} \hat{u}'_i + \sum_{i,j}^d \left\{ \hat{u}_j n_j \xi_i^* + \hat{u}_j \xi_j^* n_i \right\} u'_{b,i} d\Gamma \\
& \quad + \int_{\Gamma} \sum_i^d \xi_i^* n_i \hat{p}' d\Gamma - \int_{\Gamma} v \sum_i^d \xi_i^* \frac{\partial \hat{u}'_i}{\partial n} d\Gamma \\
& \hspace{20em} = 0 \\
& \hspace{15em} \text{(A.52)}
\end{aligned}$$

We add this and equation A.8 to the left hand side of equation A.40 to get:

$$\begin{aligned}
& \int_{\Omega} \sum_i^d \{\hat{u}_i \xi_i^*\} s' + \sum_i^d \left\{ s \xi_i^* - v \nabla^2 \xi_i^* + \frac{\partial \xi_p^*}{\partial x_i} - \sum_j^d \left[ u_{b,j} \frac{\partial \xi_i^*}{\partial x_j} + u_{b,j} \frac{\partial \xi_j^*}{\partial x_i} \right] \right\} \hat{u}'_i - \sum_{i,j}^d \left\{ \hat{u}_j \frac{\partial \xi_i^*}{\partial x_j} + \hat{u}_j \frac{\partial \xi_j^*}{\partial x_i} \right\} u'_{b,i} d\Omega \\
& \quad - \int_{\Omega} \sum_{i=1}^d \left\{ \frac{\partial \xi_i^*}{\partial x_i} \right\} \hat{p}' d\Omega \\
& \quad + \int_{\Gamma} \sum_i^d \left\{ v \frac{\partial \xi_i^*}{\partial n} + \sum_j^d [u_{b,j} \xi_i^* n_j + u_{b,j} \xi_j^* n_i] - \xi_p^* n_i \right\} \hat{u}'_i + \sum_{i,j}^d \left\{ \hat{u}_j n_j \xi_i^* + \hat{u}_j \xi_j^* n_i \right\} u'_{b,i} d\Gamma \\
& \quad + \int_{\Gamma} \sum_i^d \xi_i^* n_i \hat{p}' d\Gamma - \int_{\Gamma} v \sum_i^d \xi_i^* \frac{\partial \hat{u}'_i}{\partial n} d\Gamma \\
& \hspace{15em} \text{(A.53)} \\
& \quad + \int_{\Omega} \sum_{i=1}^d \left[ -v \nabla^2 \lambda_i - \frac{\partial \lambda_p}{\partial x_i} + \sum_{j=1}^2 \left\{ -u_{b,j} \frac{\partial \lambda_i}{\partial x_j} - u_{b,j} \frac{\partial \lambda_j}{\partial x_i} \right\} \right] u'_{b,i} d\Omega \\
& \quad - \int_{\Omega} \sum_{i=1}^d p'_b \frac{\partial \lambda_i}{\partial x_i} d\Omega + \int_{\Gamma} \sum_{i=1}^2 \left[ \lambda_p n_i + v \frac{\partial \lambda_i}{\partial n} + \sum_{j=1}^d \left\{ \lambda_j u_{b,j} n_i + \lambda_i u_{b,j} n_j \right\} \right] u'_{b,i} d\Gamma \\
& \quad + \int_{\Gamma} \sum_{i=1}^d \lambda_i n_i p'_b d\Gamma - \int_{\Gamma} \sum_{i=1}^d v \lambda_i \frac{\partial u'_{b,i}}{\partial n} d\Gamma \\
& \hspace{15em} + d^{\mathcal{J}}[\mathbf{V}] \\
& \hspace{20em} = s' \\
& \hspace{15em} \text{(A.54)}
\end{aligned}$$

We choose the governing equations and boundary conditions that  $\boldsymbol{\xi}$  and  $\boldsymbol{\lambda}$  in order to eliminate the remaining sensitivity terms.

$$s\xi_i^* - \mathbf{v}\nabla^2\xi_i^* + \frac{\partial\xi_p^*}{\partial x_i} - \sum_j \left[ u_{b,j} \frac{\partial\xi_i^*}{\partial x_j} + u_{b,j} \frac{\partial\xi_j^*}{\partial x_i} \right] = 0 \text{ in } \Omega \quad (\text{A.55})$$

$$\boldsymbol{\xi} = 0 \text{ on } \Gamma_+ \cup \Gamma_0 \cup \Gamma_w \quad (\text{A.56})$$

$$\frac{\partial\xi_1}{\partial y}, \xi_2 = 0 \text{ on } \Gamma_{free} \quad (\text{A.57})$$

$$\mathbf{v} \frac{\partial\boldsymbol{\xi}^*}{\partial n} - \xi_p^* \hat{\mathbf{n}} + \langle \mathbf{u}_b, \hat{\mathbf{n}} \rangle \boldsymbol{\xi}^* - \langle \mathbf{u}_b, \boldsymbol{\xi}^* \rangle \hat{\mathbf{n}} = 0 \text{ on } \Gamma_- \quad (\text{A.58})$$

$\boldsymbol{\xi}$  is termed the adjoint global mode. When we eliminate the sensitivity of  $s$  we end up with a normalisation condition between the direct and adjoint global modes:

$$\int_{\Omega} \langle \hat{\mathbf{u}}, \boldsymbol{\xi}^* \rangle d\Omega = 1 \quad (\text{A.59})$$

As with the convected scalar problem, the only change that occurs to the governing equations of the adjoint base-flow are the source terms and the outlet boundary condition. These become:

$$-\mathbf{v}\nabla^2\lambda_i - \frac{\partial\lambda_p}{\partial x_i} + \sum_{j=1}^d \left\{ -u_{b,j} \frac{\partial\lambda_i}{\partial x_j} - u_{b,j} \frac{\partial\lambda_j}{\partial x_i} \right\} - \sum_{i,j}^d \left\{ \hat{u}_j \frac{\partial\xi_i^*}{\partial x_j} + \hat{u}_j \frac{\partial\xi_j^*}{\partial x_i} \right\} = 0 \text{ in } \Omega \quad (\text{A.60})$$

$$\lambda_p n_i + \frac{1}{Re} \frac{\partial\lambda_i}{\partial n} + \sum_{j=1}^d \{ \lambda_j u_{b,j} n_i + \lambda_i u_{b,j} n_j \} - \sum_j^d \{ \hat{u}_j n_j \xi_i^* + \hat{u}_j \xi_j^* n_i \} = 0 \text{ on } \Gamma_- \quad (\text{A.61})$$

Now that the sensitivities have been eliminated the final expression for the shape gradient can be written in Hadamard form.

$$d\mathcal{J}[\mathbf{V}] = \int_{\Gamma_0} \sum_i^2 \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \frac{\partial\hat{u}_i}{\partial n} \left\{ \mathbf{v} \frac{\partial\xi_i^*}{\partial n} \right\} d\Gamma + \int_{\Gamma_0} \sum_{i=1}^2 \langle \mathbf{V}, \hat{\mathbf{n}} \rangle \frac{\partial u_{b,i}}{\partial n} \left[ \mathbf{v} \frac{\partial\lambda_i}{\partial n} \right] d\Gamma \quad (\text{A.62})$$

The real part of the above gives the shape gradient of the global mode's growth rate.

# Appendix B

## Automatic generation of geometric properties

In order to apply shape optimisation to a shape parametrised by a set of design variables one needs to be able to compute the deformation field,  $V$ , associated with the change in any parameter.

Additionally, for objective functions that exist as surface integrals over the deformed shape, the curvature also needs to be calculated.

These expressions can be extremely complex and deriving them by hand can be slow and error-prone. An automated solution allows new parameterisations to be experimented very quickly.

A symbolic algebra library, *sympy*, was used and the quantities required were derived using a tensor formalism. The symbolic algebra package is able to generate an analytical expression for the required quantities based on the surface coordinate. The process is performed using a tensor formalism for surfaces which is summarised in [9].

### B.1 Derivation of shape quantities

An expression for the  $x$  and  $y$  coordinates in terms of surface coordinates,  $S^\alpha$ , is input by the user.

The position vector,  $\mathbf{R}$ , is then defined by:

$$\mathbf{R}(S^\alpha) = \begin{pmatrix} x(S^\alpha) \\ y(S^\alpha) \end{pmatrix} \quad (\text{B.1})$$

The covariant,  $\mathbf{S}_\alpha$ , and contravariant,  $\mathbf{S}^\alpha$  basis vectors are generated alongside the corresponding metrics  $S_{\alpha\beta}$  and  $S^{\alpha\beta}$ .

$$\mathbf{S}_\theta = \frac{\partial \mathbf{R}}{\partial S^\alpha} \quad (\text{B.2})$$

$$S_{\alpha\beta} = \mathbf{S}_\theta \cdot \mathbf{S}_\alpha \quad (\text{B.3})$$

$$S^{\alpha\beta} = (S_{\alpha\beta})^{-1} \quad (\text{B.4})$$

$$\mathbf{S}^\alpha = S^{\alpha\beta} \mathbf{S}_\beta \quad (\text{B.5})$$

A shift tensor,  $Z_\alpha^i$ , where the upper index denotes a coordinate in the ambient space in which the surface is embedded is defined. For this work the embeddings used here are a two dimensional space defined by cartesian coordinates. This causes the shift tensor to be identical to the covariant basis.

The surface Christoffel symbol for a single surface coordinate is defined as

$$\Gamma_{\beta\gamma}^\alpha = \mathbf{S}^\alpha \cdot \frac{d\mathbf{S}_\beta}{dS^\gamma} \quad (\text{B.6})$$

For ambient cartesian coordinates the upper Levi-Citta symbol is simply:

$$\varepsilon^{ij} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (\text{B.7})$$

We can now find the normal vector from the result

$$\hat{\mathbf{n}} = N^j = \varepsilon^{ij} \varepsilon_\alpha Z_j^\alpha \quad (\text{B.8})$$

In order to get the surface curvature we must first get the covariant surface derivative of the covariant basis:

$$\nabla_\beta \mathbf{S}_\alpha = \frac{\partial \mathbf{S}_\alpha}{\partial S^\beta} - \Gamma_{\beta\alpha}^\gamma \mathbf{S}_\gamma \quad (\text{B.9})$$

Taking the dot product of this with the normal vector gives the curvature tensor  $B_{\alpha\beta}$ . This is used to get the mean curvature  $\kappa$  by

$$\kappa = B_\alpha^\alpha = B_{\alpha\beta} S^{\alpha\beta} \quad (\text{B.10})$$

The last thing that remains is imply to get the surface deformation field,  $V$ , by taking the derivative of  $\mathbf{R}$  with respect to the relevant parameter:

$$V_k = \frac{\partial \mathbf{R}}{\partial a_k} \quad (\text{B.11})$$

Whilst this is perhaps excessive for two dimensional shapes embedded in a cartesian space it can be easily extended into more dimensions and different underlying coordinate systems.