



Steady-state solvers for stability analysis of vortex dominated flows

by

Bastien JORDI

Imperial College London

Department of Aeronautics

This thesis is submitted for the degree of

Doctor of Philosophy of Imperial College London

2015

DECLARATION

This is to certify that the work presented in this thesis has been carried out at Imperial College London, and has not been previously submitted to any other university or technical institution for a degree or award. The thesis comprises only my original work, except where due acknowledgement is made in the text.

Bastien Jordi

COPYRIGHT DECLARATION

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

ACKNOWLEDGEMENTS

To start this section I would like to gratefully thank Dr. Colin Cotter and Prof. Spencer Sherwin who have both been excellent supervisors. Thank you Colin for the constant time, attention and expertise you shared with me. Thank you Spencer for the valuable inputs you added into my work. I hope the steady-state solver of Nektar++ will be useful!

If I had the opportunity to do a PhD in the United-Kingdom, I owe it to Prof. Iraj Mortazavi and Prof. Charles-Henri Bruneau, my Master's project supervisors. They introduced me to Spencer and supported my application. Thank you very much for that. I am also thankful to all the teachers I had throughout all the steps of my education who supported me and motivated me.

I would also like to formally thank the Seventh Framework Programme of the European Commission for their support to the ANADE project (Advances in Numerical and Analytical tools for DEtached flow prediction) under grant contract PITN-GA-289428. I felt really privileged being part of this adventure full of opportunities. During this project I had the chance to spend twice two months abroad to collaborate with other institutions. I would like to thank Prof. Eusebio Valero and Dr. Esteban Ferrer for hosting me during the summer 2013 at the Universidad Politécnica de Madrid. Thank you to Dr. Vincent Couallier for making my stay possible at ONERA in 2014. Both secondments were really interesting and allowed

me to discover the reality of life outside of Imperial College.

All the ANADE fellows are of course worth mentioning here. We had the chance to meet several times a year. Most of the time in nice places. A couple of times in amazing places. Our meetings were really interesting and fruitful. Special thanks to Andrea, Andrea, Maria-Chiara, Christophe, Antonio, Oli, Moritz and Esteban for sharing this experience with me.

I would also like to mention all the people from the Aeronautics department at Imperial College London. In a serious way, I would like to thank the Nektar++ team and especially Chris and Dave who helped me to deal with many code related "emergencies". Thanks to administrative staff who helped me and pull out of me unexpected paper work skills.

In a less serious way, I would like to thank every body who shared an open space office with me. We really had a good time together during these years. Meeting new people. Going to a pub. Talking about football. Becoming friends. Helping each other at the beginning of the PhD. Receiving a new computer. Sharing the joy of this day, the best of a PhD student's life. Meeting a character from the Lion King. Closing the window. Plunging. Enduring smelly food eaten at strange hours. Being locked in the University after 11pm. Closing the window. Establishing a rule saying that a late arrival in the office was synonym of a two pounds fine. Suppressing this insane rule. Quoting Els' rule several times a week. Taping any glass before drinking because it had a funny meaning. Finding out later that it was not so funny actually. Building fortresses. Closing the window. Playing Fustsal. Steaming together. Looking for the secret footballer. Praying that some miracle cough medicine could miraculously solve some persons' issues. Using the office's beamer during the Olympics and the World Cup. Betting on Football games. Going to Wembley hospital visiting a biker whose front wheel broke. Having the perfect view for the premiers of Titanic and Skyfall. Going to Queen's day. Swearing that we will go to Munich this October. No, not this October but the next one. Going to the dune. Eating sea snails and climbing oysters. Counting the sunny days in London after coming back from California. Despairing that the counter stays blocked at one.

Invading any free desk. Visiting the Narnia closet. Closing the window (because let's face it: London is no tropical place!). Putting stickers and post cards in the office. Taking naps in the office. Establishing that Friday is pie day. Having lunch outside the union or in the Park. Frequently discussing the possibility of taking a beer for lunch. Dressing up as undead wolf. Ending up in Hammersmith with a top bun less burger. Moving from RODH 363 to E455B. Taking the good chairs with us. Missing having to close the window because our new office does not have any. Falling on the floor after attempting to stand-up with a dead leg. Being stressed-out together towards the end of the PhD. Swallowing a confetti. Racing up the stairs. By order of appearance, the main actors of this adventure are Thibault, Justin, Christian, Hubert, Dirk, Markus, Jean-Eloi, Andreas, Olga, Mike and Laurent.

I would also like to thank some people very important to me that I do not have the chance to know personally though. Marta, David and their colleagues really helped me improving my English. Daniel and Gilbert kept me interested and entertained with their discussions every days for more than three years. Tony, Tim, Manu and Boris shared with me a lot of nights full of emotions.

Thanks to all my friends who visited me, hosted me or went to trips with me during my PhD student life: Benoit, Olivier, Thomas, Marie-Astrid, Amélie, Julie, Matt, Suzie, Mélanie, all the Guillaumes (especially Pons and Dober, 26), Charlotte, Malory, Fiona, J.-B., Marine, Florent, Marie-Claire, Marion, Élisa, Dominique, Annick (and the palet players), Julien, Clara, Dorian, Suze, Renaud, Catherine, Nora, Tatiana, Tom, Virginie, Wendy, Lulu, Alex, Marion, Florent, Jay, Lori and Aurélia. Some of those week ends might have substantially shortened my life expectancy but it definitely worthed it!

This last section is for those who could have been mentioned many times above. I thank my parents, Dominique and Elisabeth, for everything. Literally everything, from the Saturday mornings as a kid to the support as a grown up. Thanks to my sister, Flora, for helping me improving my organization skills and sharing confidential contact details. Thanks to Lionel, Emmanuelle and Amaury for welcoming me into your family, everywhere on the planet, regardless my type of clothing. And

finally, thank you so much to Célia for sharing your life with me for the past seven years. I showed you the other side of the river, you showed the other side of the world. We still have so much to live and to scratch. Thanks again to you all. I love you.

ABSTRACT

To compute stability analysis numerically with high accuracy, it is crucial to carefully choose the base flow around which the governing equations will be linearised. The steady-state solution is mathematically appropriate because it is a genuine solution of the fluid motion equations. In this thesis we introduce an encapsulated formulation of the selective frequency damping (SFD) method. The SFD method is an alternative to Newton's method to obtain unstable equilibria of dynamical systems. In its encapsulated formulation, the SFD method makes use of splitting methods, which means that it can be wrapped around an existing time-stepping code as a "black box". This largely simplifies the implementation of a steady-state solver into an already existing unsteady code. However this method has two main limitations: it does not converge for arbitrary control parameters; and when it does, it may take a very long time to reach a steady-state solution. Hence we also present an adaptive algorithm to address these two issues. We show that by evaluating the dominant eigenvalue of a "partially converged" steady flow, we can select SFD parameters that ensure an optimum convergence of the method. We apply this adaptive method to several classical two-dimensional test cases of computational fluid dynamics and we show that a steady-state solution can be obtained with a very limited (or without any) *a priori* knowledge of the flow stability properties. Eventually, we study the three-dimensional behaviour of the interac-

tion between two identical co-rotating trailing vortices. We use the SFD method to obtain steady base flows and compute Tri-Global stability analysis. We show that there is a fundamental qualitative difference between the least stable eigenmode observed at $Re = 250$ and the most unstable eigenmode obtained at $Re = 600$.

CONTENTS

Introduction **12**

1 Numerical methods **15**

1.1 The spectral/*hp* element method 16

1.1.1 Weighted-residuals method 18

1.1.2 Galerkin formulation 20

1.1.3 Domain decomposition 22

1.2 Time discretisation of the incompressible Navier-Stokes equations . . 25

1.3 Steady-state solvers 29

1.3.1 Newton's method: the time-stepper approach 30

1.3.2 Selective frequency damping method 34

1.4 Global linear stability analysis 36

1.4.1 General presentation 37

1.4.2 Modified Arnoldi method 39

1.5 Stabilisation methods 40

1.5.1 De-aliasing technique 41

1.5.2 Spectral vanishing viscosity 42

1.5.3 Boundary condition for unbounded truncated domains 43

1.6 Nektar++ 45

2 Encapsulated selective frequency damping method **47**

2.1 Encapsulated formulation 49

2.1.1 Generalities on splitting methods 50

2.1.2 Encapsulated SFD method 52

2.2 One-dimensional model 57

2.2.1 Limiting behaviour 59

2.2.2 Stability regions 60

2.3 Numerical simulations 65

2.3.1 Unstable steady-state 66

2.3.2 Stability of the SFD method 70

2.4 Chapter summary 74

3 An adaptive selective frequency damping method **76**

3.1 Evaluation of optimum parameters 77

3.2 Adaptive algorithm 82

3.3 Numerical simulations 86

3.3.1 Incompressible flow past a cylinder at $Re = 100$ 87

3.3.2 Incompressible flow past a cylinder at $Re = 300$ 94

3.3.3 Incompressible flow past an ellipse at $Re = 150$ with an angle of attack 97

3.3.4 Incompressible flow past a rotating cylinder at $Re = 100$ . . . 101

3.4 Chapter summary 104

4 Tri-Global stability analysis of two co-rotating vortices **106**

4.1 Vortex detection 109

4.2 Generalities on trailing vortices 111

4.3 Numerical set-up 116

4.4 Numerical simulations 119

4.4.1 Two co-rotating trailing vortices at Re = 250 120

4.4.2 Two co-rotating trailing vortices at Re = 600 125

4.5 Chapter summary 131

Conclusion **132**

Bibliography **146**

INTRODUCTION

Flow stability analysis is a major topic in fluid dynamics. From Osborne Reynolds' experiments on the flow within a pipe in 1883 to the modern simulations executed on massively parallel super-computers, hydrodynamic stability has been vastly studied over more than a century. The work presented in this thesis does not focus on experiments but on techniques to perform flow stability analysis on a computer.

There exists a wide variety of numerical methods to evaluate the linear stability properties of a flow. Local and global stability analysis aim to obtain the dominant eigenmode and eigenvalue of the fluid system to determine if an infinitesimal perturbation grows when time goes to infinity. Transient growth analysis focuses on short term amplification of initial perturbation by taking into account the non-normality of the dominant eigenmodes. Sensitivity and receptivity analysis use adjoint methods to determine how to affect a flow to have a maximum impact on a given parameter (for example the drag coefficient).

The common feature of these methods is that they are all based on the linearisation of the Navier-Stokes system around a *base flow* [1]. The choice of this base flow is crucial to compute an accurate stability analysis, relevant with the physical features of the flow studied. The *mean flow* (*i.e.* an average in time) or the solution of the Reynolds Average Navier-Stokes (RANS) equations are often used as base flow. Even though both are not genuine solutions of the fluid motion equations, they may be useful to detect relevant features of the flow. However, considering a

[1] Note that the nonlinear dynamics of finite perturbations could be studied by integrating a nonlinear constraint within the adjoint problem formulation [1]. However the starting point of such an analysis is also the linearisation of the fluid system around a base flow.

strict mathematical approach, a base flow must be a genuine solution of the fluid motion equations to allow linearisation of the equations system. This is the case of a steady-state solution of the Navier-Stokes equations which makes it an appropriate candidate to be selected as base flow.

Throughout this thesis we use a steady-state solution of the Navier-Stokes equations as base flow to compute linear global stability analysis. When stable flows are considered, obtaining a steady-state is trivial, one only has to execute a flow solver and wait until the solution becomes constant in time. When unstable flows are studied though, obtaining a steady-state is a concrete challenge. The problem of finding an equilibrium of a nonlinear system has to be addressed.

Newton's methods are very commonly used to find the roots of a nonlinear system of equations. Such methods show a quadratic convergence towards the steady-solution when provided a good initial guess. However in fluid dynamics this approach may prove impractical. For challenging flow problems at high Reynolds number, it may be the case that many complicated bifurcations, at various Reynolds numbers, must be crossed before finding the required solution. Then the Newton's method must be coupled with a continuation method. Also, due to the size of fluid computations, evaluating the Jacobian operator may require impractically large storage capacities.

As an alternative to classical Newton's methods, the selective frequency damping method (SFD) was introduced by Åkervik *et al.* in 2006. This technique consists of using standard tools of feedback control theory to modify the Navier-Stokes equations. The augmented system converges towards a steady-state solution when iterated in time. The starting point of the work realised for this thesis was to develop a steady-state solver into a code that implements high-order discretisation methods. The SFD method was chosen to be implemented into the Nektar++ toolkit.

Nektar++ [2, 3] is an open source library currently being developed at Imperial College London. It is a tensor product based finite element package written in C++ that implements high-order discretisation methods (both in space and in time). High-order schemes provide a more accurate solution than low-order schemes at a

given number of degrees of freedom (provided that the solution is smooth). As the eigenmodes returned by a linear global stability analysis are expected to be smooth, we believe that high-order codes can show very good performances for studying flow stability.

This thesis is composed of four chapters. In Chapter 1 we review the numerical methods used to compute all the simulations presented in the other chapters. This chapter summarises existing and vastly studied techniques, no new notions are presented there. In Chapter 2 we introduce the *encapsulated* formulation of the selective frequency damping method. This technique allows a very easy implementation of a steady-state solver into an existing unsteady code using a splitting method and reformulating the classical SFD method. A correspondence between the ability of the SFD method to converge towards a steady-state and the convergence of a simplified one-dimensional model is also proposed in this chapter. An *adaptive* SFD method is presented in Chapter 3. This algorithm is based on the idea that having a coarse approximation of the dominant eigenvalue of a given flow is enough to tune the parameters of the SFD method and ensure convergence towards a steady-state. The adaptive SFD method has been applied to several classical (two-dimensional) examples of computational fluid dynamics to support our arguments. In Chapter 4 we apply the SFD method to obtain an unstable steady-state solution of a fully three-dimensional flow and compute a Tri-Global stability analysis. The flow considered is the interaction between two identical co-rotating Batchelor vortices because it presents an interest to the aeronautics and Formula 1 industries.

CHAPTER 1

NUMERICAL METHODS

In this chapter we review the numerical methods used throughout this thesis. Our guideline is to use *high-order methods* to compute flow stability analysis of vortex dominated flows. Commercial CFD solvers are generally based on low-order methods. High-order discretisation schemes provide a more accurate solution than low-order schemes at a given number of degrees of freedom (provided that the solution is smooth). Such schemes are suitable for achieving high performance, even on unstructured mesh, because they have a compact stencil. These aspects make high-order methods very attractive and they currently receive wide attention from the research community.

In Sec. 1.1 and Sec. 1.2 we present high-order schemes to discretise the incompressible Navier-Stokes equations in space and in time. In Sec. 1.3 we detail two approaches to obtain unstable steady-state solutions of the fluid motion equations. In Sec. 1.4 we highlight the main features of *global stability analysis* and present the algorithm used throughout this thesis to evaluate the stability of a steady flow. These *steady-state solvers* and *stability solver* present the similarity of taking advantage of an underlying high-order discretisation scheme to be implemented as wrappers around existing codes. In Sec. 1.5 we present three techniques to control

numerical errors that may arise using high-order methods on coarse meshes and truncated unbounded domains. Finally, in Sec. 1.6 we introduce the Nektar++ tool kit. The numerical methods presented in this chapter are implemented in this code. The simulations of this thesis have all been computed using Nektar++.

1.1 The spectral/hp element method

In this section we present the spectral/hp element method, used in this thesis to discretise the fluid motion equations in space. This method combines features of methods very commonly used in computational fluid dynamics that are the *finite element method* and *spectral methods*.

The finite element method was originally introduced in structural mechanics [4] and was adopted in computational fluid dynamics for its ability to handle complex geometries. The main aspect of this method is to divide a computational domain in a finite number of subdomains to generate a *mesh*. These subdomains are called *elements* and their characteristic size is denoted h. The solution of a differential equation system is approximated by a sequence of local approximation on each element. With the finite element method, the smaller the elements (with some caveats on element shape), the more accurate the approximated solution. Decreasing h to increase the quality of the solution refers to an *h-type* refinement. Traditionally, the solution is approximated through piecewise linear functions over the elements of the mesh. In this case, the finite element method benefits from good geometrical flexibility. Arbitrary complex geometries may be decomposed into a finite number of subdomains. However this method only allows a linear convergence of the error.

Spectral methods approximate the solution globally through the entire computational domain using high-order polynomial functions. As opposed to the finite element method, spectral methods are not based on the decomposition of the computational domain. The whole domain is composed of one single element. Fourier series, Chebyshev or Legendre polynomials are generally used to approximate the solution globally. The polynomial order is denoted p and the procedure of increasing

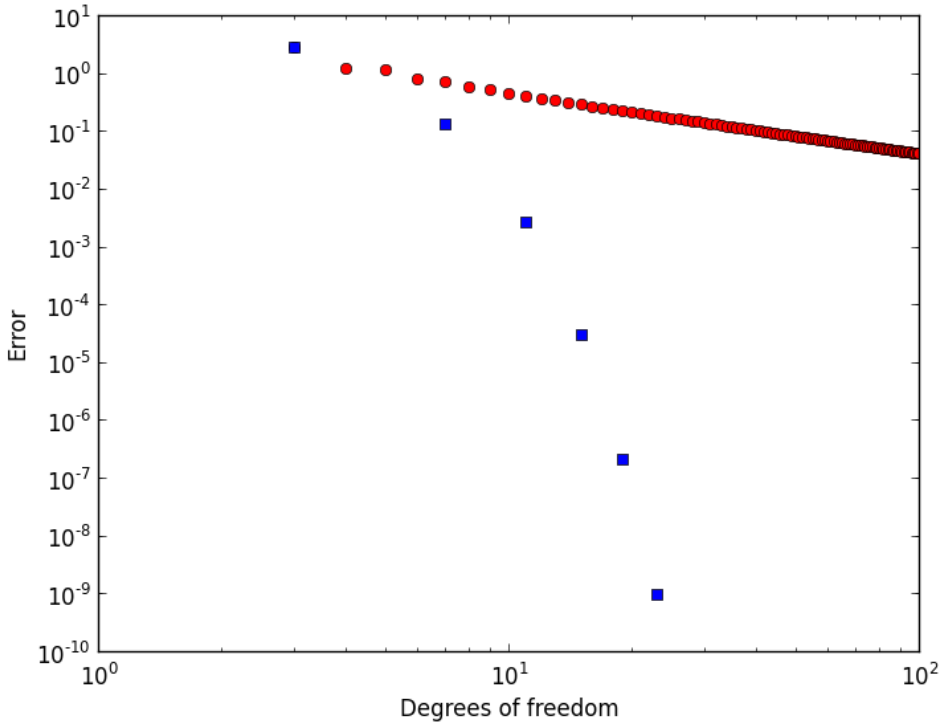


Figure 1.1: Illustration of the convergence properties for the *h-type* refinement (red circles) and *p-type* refinement (blue squares) strategies as function of the number of degrees of freedom.

p to increase the accuracy of the approximated solution is called *p-type* refinement. Due to the global definition of the solution, spectral methods can only handle very simple geometries. However, the *p-type* refinement strategy enables exponential convergence towards the exact solution.

The convergence properties of both *h-type* and *p-type* refinement strategies for the solution of $\nabla u(x) - u(x) = -(\pi^2 + 1)\sin(\pi x)$ on $x \in [-1, 1]$ are presented on Fig. 1.1. The error is obtained evaluating the difference between the numerical solution and the analytical one (which is $u(x) = \sin(\pi x)$). This semi-log graph highlights the linear convergence of the *h-type* refinement and compares with the exponential convergence of the *p-type* refinement. This feature suggests that, at a given number of degrees of freedom, increasing the polynomial order of the expansion

basis gives a more accurate solution than decreasing the size of the mesh elements at a low polynomial order. However such a behaviour can only be obtained when the solution sought for is infinitely differentiable (which is the case of $\sin(\pi x)$).

The spectral/*hp* element method, detailed by Karniadakis & Sherwin in [5], combines the geometrical flexibility of classical finite element methods and the high accuracy of spectral methods. Problems can then be solved on complex geometries using a high-order local expansion basis. Also, the spectral/*hp* element method enables efficient resolution through parallel domain decomposition because the data structure is sparser than for the spectral methods (which requires dense matrices). Note that *high-order* has different meanings for different communities. In the industrial world, where most of the commercial codes used are first or second-order, a method is considered high-order for a polynomial order $P \geqslant 3$. In the community of high-order methods, a numerical scheme is generally said to be high-order when $P \geqslant 7$. Also, note that the decomposition of the computational domain into a mesh has other utilities than handling complex geometries. For example, *h-type* refinement is necessary when the solution sought for presents a lack of continuity (*e.g.* a shock for compressible flows; sharp edges in the computational domain or boundary layers for incompressible flows). Roughly, *p-type* refinement is preferred when the solution is *smooth enough*, otherwise *h-type* refinement, with a lower polynomial order, should be considered. In Sec. 1.1.1, 1.1.2 and 1.1.3 we briefly outline the fundamental ideas of the spectral/*hp* element method.

1.1.1 Weighted-residuals method

The starting point of both the finite-element method and spectral methods is the establishment of a weak formulation of the governing equations. In this section we explain in a general case how such a formulation may be obtained.

In a domain Ω, we consider the linear differential equation

$$\mathbb{L}(u) = 0, \tag{1.1}$$

assuming that appropriate initial and boundary are defined.

To implement a method that solves (1.1) on a computer, we must seek a finite approximated solution. We assume that the solution $u(\boldsymbol{x}, t)$ can be accurately described by

$$u_h = \sum_{i=1}^{N_{\text{dof}}} \hat{u}_i(t)\Phi_i(\boldsymbol{x}), \tag{1.2}$$

where $\Phi_i(\boldsymbol{x})$ are analytical functions called *basis* (or *expension*) functions, $\hat{u}_i(t)$ are the N_{dof} unknown coefficients. Note that the higher N_{dof}, the more accurate the approximation of u.

As u_h is only an approximation of u, it does not satisfy exactly (1.1). We introduce the non-zero residual R such that

$$\mathbb{L}(u_h) = R(u_h). \tag{1.3}$$

We now introduce a set v_j of *weight* (or *test*) functions. The residual is restricted to satisfy

$$\int_\Omega v_j(\boldsymbol{x})R(u_h)\mathrm{d}\boldsymbol{x} = 0, \quad j = 1, \ \ldots\ , N_{\text{dof}}. \tag{1.4}$$

The technique of multiplying the weight function by the residual and integrating over the domain Ω is called *weighted-residual*. The method takes its name from the fact that the weighted-residual is constrained to be zero.

By substituting (1.3) into (1.4) we obtain

$$\int_\Omega v_j(\boldsymbol{x})\mathbb{L}(u_h)\mathrm{d}\boldsymbol{x} = 0, \quad j = 1, \ \ldots\ , N_{\text{dof}}, \tag{1.5}$$

which is the weak formulation of the linear problem (1.1). The solution of (1.5) is said to be weak because through mathematical operations (typically integration by parts) it is possible to shift some derivatives from u_h to v_j. Hence u_h has fewer (or weaker) continuity requirements that the exact solution u. The relaxation of some continuity constraints could make the solution of the weak problem (1.5) easier than the strong problem (1.1). Although it may happen that a weak solution presents some discontinuities and does not satisfy the strong equation. Note that throughout this thesis only incompressible flows, where there are no shocks, are considered. For such flows a weak solution is also solution of the strong problem.

The choice of the weight functions is very important because it determines the nature of the numerical scheme. A description of the most commonly used weight functions in computational fluid dynamics can be found in [5]. Throughout this thesis, the Galerkin method is used. This method is characterised by

$$v_j(\boldsymbol{x}) = \Phi_j(\boldsymbol{x}), \tag{1.6}$$

which means the basis functions and the test functions are chosen to be identical.

1.1.2 Galerkin formulation

In this section we review the steps of the solution of a linear differential equation with the Galerkin method. As the scheme presented in Sec. 1.2 that discretises in time the Navier-Stokes equations is composed from a succession of Helmholtz problems, here we present the fundamental aspects of the solution mechanism of the Helmholtz equation in one-dimension.

On the one-dimensional domain $\Omega = \{x \mid 0 < x < l\}$, the Helmholtz equation is defined by

$$\mathbb{L}(u) = \frac{\partial^2 u}{\partial x^2} - \lambda u + f = 0, \tag{1.7}$$

where λ is real, positive and constant. To this equation we associate a Dirichlet condition on the left boundary and a Neumann condition on the right boundary, such that

$$u(0) = c_{\mathcal{D}} \quad \text{and} \quad \frac{\partial u}{\partial x}(l) = c_{\mathcal{N}}. \tag{1.8}$$

To construct a weak Galerkin approximation of (1.7) we first multiply the problem by an arbitrary weight function v, which is zero on all Dirichlet boundaries, and integrate over the domain Ω,

$$\int_0^l v \frac{\partial^2 u}{\partial x^2} \, \mathrm{d}x - \int_0^l \lambda v u \, \mathrm{d}x + \int_0^l v f \, \mathrm{d}x = 0. \tag{1.9}$$

After integration by parts of the first term, this weak problem can be formulated as

$$a(v, u) = f(v), \tag{1.10}$$

where

$$a(v,u) = \int_0^l \left(\frac{\partial v}{\partial x}\frac{\partial u}{\partial x} + \lambda v u \right) \, \mathrm{d}x, \tag{1.11a}$$

$$\text{and} \quad f(v) = \int_0^l v f \, \mathrm{d}x + \left[v \frac{\partial u}{\partial x} \right]_0^l. \tag{1.11b}$$

Adopting similar notations as structural mechanics, we introduce $E(\Omega)$ the energy space such that $E(\Omega) = \{u \mid a(u,u) < \infty\}$ associated with the norm $||u||_E = \sqrt{a(u,u)}$. Functions belonging to the energy space are called H^1 functions. We only consider solutions of (1.10) that lie in the energy space and satisfy the Dirichlet boundary condition (1.8). This defines a *expansion* space $\mathcal{U}$ such that

$$\mathcal{U} = \{u \mid u \in H^1, \; u(0) = c_{\mathcal{D}}\}. \tag{1.12}$$

Similarly, the weight (or test) functions, which are by definition homogeneous (*i.e.* equal to zero) on all Dirichlet boundaries, belong the *test* space $\mathcal{V}$ defined by

$$\mathcal{V} = \{v \mid v \in H^1, \; v(0) = 0\}. \tag{1.13}$$

Hence the solution of (1.10) consists of finding $u \in \mathcal{U}$ for all $v \in \mathcal{V}$. Note that the Neumann boundary condition on the right of the domain is naturally handled through the formulation of $f(v)$ in (1.11b). Also note that the formulation of (1.10) is valid in infinite dimensions. To restrict it to a finite dimensional space, we introduce the subspaces $\mathcal{U}_h \subset \mathcal{U}$ and $\mathcal{V}_h \subset \mathcal{V}$ which contain a finite number of functions. Hence the approximation of (1.10) is formulated: find $u_h \in \mathcal{U}_h$ such that

$$a(v_h, u_h) = f(v_h), \quad \forall v_h \in \mathcal{V}_h. \tag{1.14}$$

To impose the Dirichlet boundary condition (1.8), the *lifting* process is used. Lifting consists in decomposing the approximated solution u_h into a known function $u^{\mathcal{D}} \in \mathcal{U}_h$, which satisfies the Dirichlet condition, and an unknown function $u^{\mathcal{H}} \in \mathcal{V}_h$ homogeneous on the Dirichlet boundary. Then u_h becomes

$$u_h = u^{\mathcal{H}} + u^{\mathcal{D}}, \quad \text{where } u^{\mathcal{H}}(0) = 0 \;\text{ and }\; u^{\mathcal{D}}(0) = c_{\mathcal{D}}. \tag{1.15}$$

We recall that in the Galerkin formulation, the same set of functions is used for the basis and test functions. Such an expansion is now possible because both $u^{\mathcal{H}}$ and v are chosen to be in the test space $\mathcal{V}_h$. The Galerkin formulation of (1.10) can then be written: find $u^{\mathcal{H}} \in \mathcal{V}_h$ such that

$$a(v_h, u^{\mathcal{H}}) = f(v_h) - a(v_h, u^{\mathcal{D}}), \quad \forall v_h \in \mathcal{V}_h \ \text{ and } \ u^{\mathcal{D}} \in \mathcal{U}_h. \tag{1.16}$$

It can be shown that the solution of this equation is unique [5].

In this section we highlighted the key features of the establishment of the Galerkin formulation of a one-dimensional Helmholtz problem. We have shown that Neumann boundary conditions are naturally imposed through the weak formulation and that Dirichlet boundary conditions are imposed strongly using the lifting process. Note that minor modifications of the problem formulation are necessary to impose Robin boundary conditions (which are a linear combination of Neumann and Dirichlet type conditions). The extension to higher space dimensions is straightforward.

1.1.3 Domain decomposition

The key feature of finite element methods is the decomposition of the computational domain Ω into a finite number N_{el} of subdomains Ω_e called elements such that

$$\Omega = \bigcup_{e=1}^{N_{el}} \Omega_e \ \text{ and } \ \bigcap_{e=1}^{N_{el}} \Omega_e = \emptyset. \tag{1.17}$$

When using a large number of elements, it would be extremely inconvenient to globally define the expansion basis $\Phi_i(x)$ (defined in (1.2)). Instead we introduce the local expansion basis $\phi_p(\xi)$ defined on the one-dimensional standard element $\Omega_{st} = \{\xi \mid \xi \in [-1, 1]\}$. We also introduce an analytical mapping function $\Psi_e(\xi)$ transforming the local coordinates into the corresponding global coordinate on the element Ω_e. An inverse mapping $\xi = \Psi_e^{-1}(x)$ is also needed. Hence the approximated solution u_h can be written

$$u_h = \sum_{i=1}^{N_{\text{dof}}} \hat{u}_i \Phi_i(x) = \sum_{e=1}^{N_{el}} \sum_{p=0}^{P} \hat{u}_p^e \phi_p^e(\xi), \tag{1.18}$$

where P is the polynomial order of the expansion and $\phi_p^e(\xi) = \phi_p(\Psi_e^{-1}(x))$.

To ensure the continuity of this expansion in one dimension, we enforce

$$\hat{u}_P^e \phi_P^e(1) = \hat{u}_0^{e+1} \phi_0^{e+1}(-1), \tag{1.19}$$

where the superscripts e and $e+1$ refer to two adjacent elements. This technique is called *boundary/interior* decomposition. The modes at a boundary of an element are linked to the modes of the neighbour element but the interior modes are independent and are zero at the boundaries. This implies that there are more local expansion modes $\hat{u}_p^e$ than global expansion modes $\hat{u}_i$.

We introduce vector notations such that $\hat{\boldsymbol{u}}^e = [\hat{u}_0^e, \ \dots, \hat{u}_P^e]^\top$ represents the coefficients of u_h on the element Ω_e, and $\boldsymbol{\phi}^e = [\phi_0^e, \ \dots, \phi_P^e]^\top$ represents the local expansion modes. Then (1.19) may be written

$$u_h = \sum_{e=1}^{N_{el}} \boldsymbol{\phi}^{e\top} \hat{\boldsymbol{u}}^e. \tag{1.20}$$

If we adopt similar notations for the test function v_h, in the absence of boundary conditions, the matrix formulation of Helmholtz problem (1.14) on the element Ω_e is

$$\hat{\boldsymbol{v}}^{e\top}[(\boldsymbol{L}^e + \lambda \boldsymbol{M}^e)\hat{\boldsymbol{u}}^e = \boldsymbol{f}^e] \quad \Rightarrow \quad (\boldsymbol{L}^e + \lambda \boldsymbol{M}^e)\hat{\boldsymbol{u}}^e = \boldsymbol{f}^e, \tag{1.21}$$

where $\boldsymbol{M}^e$ is the *elemental mass matrix*, $\boldsymbol{L}^e$ is the *elemental Laplacian operator* and $\boldsymbol{f}^e$ is the projection of the forcing term on the coefficient space. The matrices are defined such that

$$\boldsymbol{M}_{pq}^e = \int_{\Omega_e} \phi_p^e(\xi)\phi_q^e(\xi)\mathrm{d}\xi \quad \text{and} \quad \boldsymbol{L}_{pq}^e = \int_{\Omega_e} \frac{\mathrm{d}}{\mathrm{d}\xi}\phi_p^e(\xi)\frac{\mathrm{d}}{\mathrm{d}\xi}\phi_q^e(\xi)\mathrm{d}\xi. \tag{1.22}$$

Note that (1.21) is only defined at the element level. To obtain a global solution an *assembly matrix* must be defined. This matrix, which is very sparse, handles the connectivity between adjacent elements and allows us to obtain the global coefficients $\hat{u}_i$. Also, boundary conditions are incorporated within a global definition of the right-hand side term.

The treatment of u_h on the local element basis allows us to perform many operations (integration, differentiation...) at a low computational cost. This process can also be parallelised very efficiently.

The choice of the expansion basis $\phi_p(\xi)$ is crucial for the numerical behaviour of the solver. Two categories of expansion basis exist. One is called *nodal expansion*, it uses a set of $P+1$ Lagrange polynomials of order P. The solution is evaluated on a set of *nodal* points (or *quadrature* points) which have a physical meaning. With appropriate choice of quadrature points the nodal basis can be made orthogonal and the corresponding elemental mass matrix is diagonal. Orthogonal expansion basis are very well conditioned, which is advantageous for *p-type* refinements. To enforce continuity it is sufficient to ensure that the end points of the standard element are part of the set of quadrature points. For all choice of quadrature rules, the nodal expansion results in a full elemental Laplacian matrix.

The second type of expansion basis is the *modal expansion*. Modal basis are hierarchical because they require $P+1$ polynomials from order 0 to P and a polynomial of order $P-1$ is contained in a polynomial of order P. This expansion is based on Jacobi polynomials, which are orthogonal. Note that Legendre and Chebyshev polynomials are specific examples of Jacobi polynomials. However to impose the C^0-continuity between neighbour elements and the boundary/interior decomposition, the orthogonality must to be broken. Linear modes are required at the boundaries while Jacobi polynomials define the interior modes. This generates a penta-diagonal elemental mass matrix and a nearly diagonal Laplacian matrix (except for the linear boundary modes).

In this thesis we use a *modified* modal expansion basis, with linear modes at the boundaries. This choice was made because this basis is "nearly" orthogonal (then well conditioned) and computationally more efficient than a nodal basis. On the one-dimensional standard element, the modified modal expansion basis is defined by

$$\phi_p(\xi) = \begin{cases} \frac{1-\xi}{2} & p = 0, \\ \left(\frac{1-\xi}{2}\right)\left(\frac{1+\xi}{2}\right) J_{p-1}^{1,1}(\xi) & 0 < p < P, \\ \frac{1+\xi}{2} & p = P, \end{cases} \tag{1.23}$$

where $J_p^{1,1}$ are Jacobi polynomials of order p.

The concept of subdomain decomposition can of course be generalised to higher dimensions. For two-dimensional simulations, the computational domain is composed of quadrilateral elements or of triangles. When quadrilateral elements are used, the standard element is the square $[-1,1]^2$ and the expansion basis is a tensor product of two one-dimensional expansions (1.23). For triangles, generally used to mesh complex geometries, the extension is slightly more complicated. The standard element is a triangle and its local coordinates are not independent from one another. Then a direct expression of the tensor product is not possible. Karniadakis & Sherwin [5] introduced a *collapsed* coordinate system to map the standard square element into the standard triangle. In three-dimensions, with hexahedra the standard element is the cube $[-1,1]^3$ but collapsed coordinates must be used for prisms, pyramids or tetrahedra. For more details about two and three-dimensional expansion basis, the reader is invited to refer to [5].

Finally, we can add that, when a high polynomial order is used, the spectral/*hp* element method can accurately approximate the solution on a mesh that is much coarser than the type of mesh required for low order methods. Then curved elements may be needed to precisely define complex geometries. To handle curved elements, a non-linear mapping Ψ_e is used. Hence the inverse mapping may not have an analytical expression and in such a case, iterative methods are necessary to compute the transform from the physical space to the standard element.

1.2 Time discretisation of the incompressible Navier-Stokes equations

The spectral/*hp* element method highlighted in Sec. 1.1 can be applied to discretise in space any kind of equation system. Throughout this thesis though, we are interested in physical phenomena governed by the incompressible Navier-Stokes equations. In this section we review a time integration method specifically developed for these equations. The non-dimensional formulation of the incompressible

Navier-Stokes equations is

$$\begin{cases} \partial_t \mathbf{u} = -\left(\mathbf{u} \cdot \nabla\right) \mathbf{u} - \nabla p + Re^{-1} \nabla^2 \mathbf{u}, \\ \nabla \cdot \mathbf{u} = 0, \end{cases} \tag{1.24}$$

where $\mathbf{u}$ is the fluid velocity and p is the kinematic pressure. $Re = U_\infty L/\nu$ is the Reynolds number where U_∞ is the characteristic velocity, L is the characteristic length and ν is the kinematic viscosity. Appropriate initial and boundary conditions must also be defined. Note that in system (1.24) the pressure can be seen as a parameter whose role is to enforce the incompressibility constraint (*i.e.* the divergent-free velocity field).

The time-integration methods to solve the Navier-Stokes equations can be classified in two main families: coupled methods where velocity and pressure are solved together and splitting methods where the system is decomposed into several subproblems. Through a careful splitting of the Navier-Stokes equations, it is possible to separate the primary variables. In this section we present the *velocity-correction* scheme introduced by Karniadakis *et al.* [6], which is a high-order splitting scheme. Note that in the literature splitting methods can also be called fractional-step methods.

The main idea of a splitting method is to decompose a large problem into several smaller (simpler) subproblems such that the final solution of one subproblem is used as initial condition for the next one. Here, the solution of the advection (nonlinear) term, the diffusion (linear) term and the pressure term are separated. Also, the velocity-correction scheme uses an explicit/implicit time-integration mechanism. The advection term is expressed explicitly whereas the diffusion term is treated implicitly. The stiffness of the linear term would require the use of impractically small time steps if it were solved with an explicit method, that is why it is solved implicitly.

We introduce $\mathbf{u}^n$ and p^n the approximated solutions at time $t = n\Delta t$, where $n \in \mathbb{N}$ and Δt is the time step. The general time-discrete formulation of the velocity-

correction scheme is

$$\frac{\hat{\mathbf{u}} - \sum_{q=0}^{J-1} \alpha_q \mathbf{u}^{n-q}}{\Delta t} = -\sum_{q=0}^{J-1} \beta_q \left(\mathbf{u}^{n-q} \cdot \nabla\right) \mathbf{u}^{n-q}, \tag{1.25a}$$

$$\frac{\hat{\hat{\mathbf{u}}} - \hat{\mathbf{u}}}{\Delta t} = -\nabla p^{n+1}, \tag{1.25b}$$

$$\frac{\gamma_0 \mathbf{u}^{n+1} - \hat{\hat{\mathbf{u}}}}{\Delta t} = \nu \nabla^2 \mathbf{u}^{n+1}, \tag{1.25c}$$

where $\hat{\mathbf{u}}$ and $\hat{\hat{\mathbf{u}}}$ are intermediate velocity fields. The parameter J stands for the order of the method. The coefficients α_q, β_q and α_0 depend on the order J. A list of appropriate stiffly-stable coefficient values (up to third order) is given in [6]. Throughout this thesis we used a second order scheme in time, which corresponds to $\alpha_0 = 2$, $\alpha_1 = -1/2$, $\beta_0 = 2$, $\beta_1 = -1$ and $\gamma_0 = 3/2$.

Solving (1.25b) is the key step of the velocity correction scheme. First we assume that the second intermediate velocity satisfies the incompressibility constraint (*i.e.* $\nabla \cdot \hat{\hat{\mathbf{u}}} = 0$), which leads to the formulation of a Poisson equation for the pressure such that

$$\nabla^2 p^{n+1} = \nabla \cdot \left(\frac{\hat{\mathbf{u}}}{\Delta t}\right). \tag{1.26}$$

Then we can express the second intermediate velocity field as a function of $\hat{\mathbf{u}}$ and p^{n+1}. Note the incompressibility constraint is only guaranteed for $\hat{\hat{\mathbf{u}}}$. An error is introduced at the last step and $\mathbf{u}^{n+1}$ is not completely divergence free.

Splitting methods enable us to solve complex problems by decomposing them into smaller subproblems but this process introduces an error (called splitting error). Orszag *et al.* [7] showed that this splitting error reduces the order of accuracy of the whole time-integration scheme. To maintain the temporal accuracy of order J, the solution of (1.26) requires a specific treatment of the pressure boundary condition. Where homogeneous Dirichlet boundary conditions are present (*i.e.* solid walls), Karniadakis *et al.* [6] suggest to use high-order boundary conditions of the type

$$\frac{\partial p}{\partial n}^{n+1} = -\left[\frac{\partial \hat{\mathbf{u}}}{\partial t} + \sum_{q=0}^{J-1} \beta_q \left[\left(\mathbf{u}^{n-q} \cdot \nabla\right) \mathbf{u}^{n-q} + \nu \nabla \times \boldsymbol{\omega}^{n-q}\right]\right] \cdot \mathbf{n}, \tag{1.27}$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ represents the vorticity field and $\mathbf{n}$ is a unit vector normal to

the boundary. Note that when an homogeneous Dirichlet boundary condition is imposed on the velocity field, the condition on the pressure is of Neumann type.

The pressure boundary condition (1.27) requires a high-order spatial discretisation of the Navier-Stokes equations. Indeed, the term $\nabla \times \boldsymbol{\omega}$ requires a polynomial order $P > 2$ to be evaluated. Hence the velocity correction scheme presented here can be second or third order in time only if the spatial discretisation technique used is sufficiently high-order. That is why it is well suited to be coupled with the spectral/*hp* element method.

With this method, the time evolution of the incompressible Navier-Stokes equations is decomposed into *four* steps. First, the intermediate velocity field $\hat{\mathbf{u}}$ is explicitly evaluated as a linear combination of the advection term at previous time steps. Second, a Poisson equation (with high-order boundary conditions) is solved to obtain the pressure at the new time iteration. Then the second velocity field $\hat{\hat{\mathbf{u}}}$ is expressed as a function of the first intermediate velocity field and the pressure. Finally, an Helmholtz equation is solved to obtain the velocity field at the new time level.

By introducing intermediate velocity fields, the time evolution of the Navier-Stokes equations is transformed into successive solution of the elliptic problems (1.26) and (1.25c). Hence the implementation of the velocity-correction scheme essentially requires the implementation of an efficient solver of the Poisson and Helmholtz equations. Note that generally, an Helmholtz equation is better conditioned than a Poisson equation. This means that iterative methods will converge slower when solving a Poisson equation than an Helmholtz equation. Then in the implementation of the velocity-correction scheme, the *pressure solver* is the most demanding part of the algorithm in terms of computational time.

The velocity-correction scheme presented in this section offers better performances in terms of accuracy and stability that any Adams-family type methods [6] that are splitting methods commonly used in computational fluid dynamics. Ferrer *et al.* [8] showed that the limit of stability of the time-evolution of the incompressible Navier-Stokes equations using an high-order Galerkin method depends only on

the temporal splitting scheme and not on the the spatial discretisation. In [8], the authors also showed that the velocity-correction scheme is a stabilised-like scheme whose stability is entirely governed by the values of the time step and the kinematic viscosity.

Throughout this thesis, the Navier-Stokes solver is based on the spectral/hp element method and on the velocity correction scheme. Then the solution returned presents a high-order accuracy both in space and in time.

1.3 Steady-state solvers

The numerical method described in Sec. 1.2 enables an accurate representation of the time-evolution of unsteady flows. In this section we present two approaches to obtain steady-state solutions of the incompressible Navier-Stokes equations (*i.e.* time-independent solutions).

When the system considered is stable, obtaining a numerical steady-state solution is trivial: we simply have to execute the solver and wait long enough until the flow becomes constant in time. However for unstable flows, obtaining a steady-state solution is a concrete challenge. The problem of finding an unstable fixed point of a non-linear system has to be addressed. One possible approach is to use Newton's methods, because a quadratic convergence towards the steady-state is guaranteed if a good initial guess of the solution is provided. However continuation methods are usually required to study problems with increasing complexity (*e.g.* with high Reynolds number) because a good initial guess is generally not at hand.

Finding an unstable steady-state solution of the incompressible Navier-Stokes equations is the fundamental preliminary step before computing flow stability analysis. The development of an accurate, robust and efficient steady-state solver is the heart of the work presented in this thesis.

In Sec. 1.3.1 we review the Newton's method introduced by Tuckerman & Barkley [9] that takes advantage of an existing unsteady code's structure to easily implement a Newton's methods. In Sec. 1.3.2 we present an alternative to classical

Newton's method to obtain steady-state solutions of the Navier-Stokes equations. This approach, based on the filtering of unstable temporal frequencies, is the selective frequency damping method [10].

1.3.1 Newton's method: the time-stepper approach

The classical Newton's method (also called Newton–Raphson method) aims to solve

$$\mathbf{F}(\mathbf{q}) = 0, \tag{1.28}$$

where $\mathbf{q}$ represents the problem unknown(s) and $\mathbf{F}$ is a non-linear operator.

We introduce the index k of the Newton's iterative method and we write the Taylor's expansion of $\mathbf{F}$ such that

$$\mathbf{F}(\mathbf{q}^{k+1}) = \mathbf{F}(\mathbf{q}^k) + \mathbf{J}(\mathbf{q}^k)(\mathbf{q}^{k+1} - \mathbf{q}^k) + \text{ h. o. t.}, \tag{1.29}$$

where $\mathbf{J}$ represents the Jacobian of $\mathbf{F}$.

Neglecting the high order terms of (1.29) and introducing the unknown $\delta\mathbf{q} = \mathbf{q}^{k+1} - \mathbf{q}^k$, the Newton's method consists in iterating

$$\begin{cases} \mathbf{J}(\mathbf{q}^k)\delta\mathbf{q} = -\mathbf{F}(\mathbf{q}^k), \\ \mathbf{q}^{k+1} = \mathbf{q}^k + \delta\mathbf{q}, \end{cases} \tag{1.30}$$

until $||\delta\mathbf{q}||$ becomes smaller than a desired tolerance.

To solve (1.30) numerically, a procedure to obtain the Jacobian matrix and a linear solver are necessary. These steps may require a lot of work to be implemented efficiently and, if the system studied is large, the memory requirements may be prohibitive. The time-stepper approach introduced by Tuckerman & Barkley [9] exploits the structure an existing unsteady solver to minimize the programming efforts in order to implement a steady-state solver. In computational fluid dynamics the implementation of a Navier-Stokes solver is a tremendous task and it could be highly valuable to perform several kinds of analysis with a single code.

If the kinematic pressure is set to be solution of the Poisson equation $\nabla^2 p = \nabla \cdot [(\mathbf{u} \cdot \nabla)\,\mathbf{u}]$ (with appropriate boundary conditions), the incompressible Navier-Stokes equations can be written only as function of the velocity such that

$$\begin{aligned} \partial_t \mathbf{u} &= -\left[\mathbf{I} - \nabla\nabla^{-2}\nabla\cdot\right](\mathbf{u}\cdot\nabla)\,\mathbf{u} + Re^{-1}\nabla^2\mathbf{u}, \\ &= \mathcal{N}(\mathbf{u}) + \mathcal{L}\mathbf{u}, \end{aligned} \tag{1.31}$$

where $\mathcal{L}$ and $\mathcal{N}$ represent respectively the linear and non-linear parts of the Navier-Stokes equations (the method used to discretize in space these operators is not addressed here).

The time-stepper approach is based on the fact that there already exists a code to solve (1.31). This code must implement a first-order implicit/explicit time integration scheme. As the linear operator $\mathcal{L}$ is responsible for the fastest time-scales in the fluid system, this term is treated implicitly whereas the non-linear term is treated explicitly. Note that an explicit treatment of the linear term (*i.e.* a fully explicit code) would require to use impractically small time steps to guarantee the stability of the solver. Hence the solution of (1.31) on one time step implements

$$\begin{aligned} \mathbf{u}(t+\Delta t) &= \mathbf{u}(t) + \Delta t\left[\mathcal{N}(\mathbf{u}(t)) + \mathcal{L}\mathbf{u}(t+\Delta t)\right], \\ &= \left(\mathbf{I} - \Delta t\mathcal{L}\right)^{-1}\,\left(\mathbf{I} + \Delta t\mathcal{N}\right)\,\mathbf{u}(t). \end{aligned} \tag{1.32}$$

Note that in most Navier-Stokes solvers, the inverse operator $\left(\mathbf{I} - \Delta t\mathcal{L}\right)^{-1}$ is not constructed explicitly but some tricks are used to deal with it economically. The time-stepper approach exploits these tricks in order to transform an unsteady code into a steady-state solver.

We denote $\mathcal{N}_\mathbf{u} + \mathcal{L}$ the Jacobian operator of system (1.31). The method presented in this section is *matrix-free*, which means that the Jacobian operator is never formally constructed, only its action on the unknown vector $\delta\mathbf{u}$ is required. Then the Newton's method applied to the incompressible Navier-Stokes equations can be written

$$\begin{cases} \left(\mathcal{N}_\mathbf{u} + \mathcal{L}\right)\delta\mathbf{u} = -\left(\mathcal{N} + \mathcal{L}\right)\mathbf{u}, \\ \mathbf{u} := \mathbf{u} + \delta\mathbf{u}. \end{cases} \tag{1.33}$$

The issue with the solution of the linear system (1.33) is that $\mathcal{N}_\mathbf{u} + \mathcal{L}$ is poorly conditioned. It means that the ratio between this operator's largest and smallest singular values is large. As a consequence, iterative methods such as the conjugate gradient (or a variant for operators which are not symmetric positive definite) converge very slowly or may not converge at all. To address this issue a preconditioner must be used. This is a very standard trick in numerical analysis which consists in multiplying both sides of an equation by the same matrix. The perfect preconditioner is the inverse of the operator considered. In system (1.33) the linear operator $\mathcal{L}$ is the main source of difficulties because it models features with a large range of time-scales. Hence it is the source the poor preconditioning of $\mathcal{N}_\mathbf{u} + \mathcal{L}$. Then we would like to use a preconditioner containing $\mathcal{L}^{-1}$. The question arising now is: how can we access $\mathcal{L}^{-1}$ without having to explicitly form this matrix?

The answer is given by looking at the structure of the unsteady code defined by (1.32). The operator $(\mathbf{I} - \Delta t\mathcal{L})^{-1}$ is present in the formulation of (1.32) (but is never constructed explicitly in the code). Then it can be used as preconditioner because it contains $\mathcal{L}^{-1}$. This approach is called *Stokes preconditioning* [11]. We multiply both sides of (1.33) by $(\mathbf{I} - \Delta t\mathcal{L})^{-1} \Delta t$ and obtain

$$\begin{gathered}
(\mathbf{I} - \Delta t\mathcal{L})^{-1} \Delta t \left(\mathcal{N}_\mathbf{u} + \mathcal{L}\right) \delta\mathbf{u} = -(\mathbf{I} - \Delta t\mathcal{L})^{-1} \Delta t \left(\mathcal{N} + \mathcal{L}\right) \mathbf{u}, \\
(\mathbf{I} - \Delta t\mathcal{L})^{-1} \left[\mathbf{I} + \Delta t\mathcal{N}_\mathbf{u} - (\mathbf{I} - \Delta t\mathcal{L})\right] \delta\mathbf{u} = -(\mathbf{I} - \Delta t\mathcal{L})^{-1} \left[\mathbf{I} + \Delta t\mathcal{N} - (\mathbf{I} - \Delta t\mathcal{L})\right] \mathbf{u}, \\
\left[(\mathbf{I} - \Delta t\mathcal{L})^{-1} (\mathbf{I} + \Delta t\mathcal{N}_\mathbf{u}) - \mathbf{I}\right] \delta\mathbf{u} = -\left[(\mathbf{I} - \Delta t\mathcal{L})^{-1} (\mathbf{I} + \Delta t\mathcal{N}) - \mathbf{I}\right] \mathbf{u}.
\end{gathered} \tag{1.34}$$

We notice that the right hand side of (1.34) is actually the difference between two consecutive time steps. Then it can be evaluated very easily with the existing unsteady code. Similarly, the left hand side is the difference between two *linearised* time steps and can also be evaluated easily. The linear system (1.34) can be solved with the Bi-CGSTAB [12] method and iterated until convergence (*i.e.* until $||\delta\mathbf{u}||$ becomes smaller than a desired tolerance).

The method presented in this section requires minimum efforts to be implemented and its formulation is *matrix-free*. The features of classical Newton's meth-

ods are preserved and quadratic convergence is guaranteed if the initial guess is close enough to the final solution. However for high Reynolds number flows, many bifurcations may occur and must be tracked carefully with the help of a continuation method.

For incompressible flows, the idea behind a continuation method is to use the steady-state solution at a given Reynolds number as initial condition for executing the Newton's method at a higher-Reynolds number. This allows us to gradually increase the Reynolds number. Historically, the lid-driven cavity flow has been one of the first case studied by Schreiber & Keller with continuation methods [13]. Reviews about continuation techniques and bifurcation analysis have been presented by Cliffe *et al.* [14] and Dijkstra *et al.* [15]. These methods are widely used in computational fluid dynamics. For example, Wales *et al.* used a continuation method to obtain an unstable solution of the RANS equation to study the compressible turbulent flow past a two-dimensional aerofoil [16, 17, 18]. Cherubini *et al.* used a continuation method to find an unstable steady-state of the incompressible Navier-Stokes equations to study the effects of non-normal modes (*i.e.* transient growth analysis) on large laminar separation bubbles [19].

Continuation methods are also generally used to track down complicated bifurcation patterns. Chen *et al.* [20, 21] investigated the multiplicity of steady-state solutions of the two-sided lid-driven cavity flow. Mamun & Tuckerman [22] detailed the bifurcations of the cylindrical Couette flow. Tuckerman & Boronska [23] presented an extremely rich bifurcation diagram of the Rayleigh–Bénard convection within a vertical cylinder filled with water and heated from below. This analysis exhibits more different flow configurations than the ones that could be found experimentally [24]. Also, Sanchez *et al.* [25] used a continuation method to carry on a bifurcation analysis of the flow in a cylindrical container driven by differential rotation.

The literature on continuation methods is extremely rich and the list given above is far from being exhaustive.

In the following section we present the selective frequency damping (SFD) method,

for which convergence is less sensitive to the choice of the initial condition. Throughout this thesis the SFD method will be used to obtain unstable steady-state solutions of the Navier-Stokes equations. However the encapsulated formulation introduced in Chapter 2 is strongly inspired by the time-stepper approach with the notion of adapting an existing unsteady flow solver. The stability analysis solver presented in Sec. 1.4.2 also uses the time-stepper approach. Eventually, we add that the SFD method does not require continuation methods to converge at high Reynolds number.

1.3.2 Selective frequency damping method

In 2006 Åkervik *et al.* introduced the selective frequency damping (SFD) method [10]. It was proposed as an alternative to classical Newton's methods for finding unstable steady-state solutions of the Navier-Stokes equations. The SFD method is based on the damping of unstable temporal frequencies. In this section, we detail the steps leading to its formulation.

We consider the problem

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}), \tag{1.35}$$

where $\mathbf{q}$ represents the problem unknown(s), $\mathbf{F}$ is an operator (which can be non-linear) and the dot represents the time derivation. With appropriate initial and boundary conditions, (1.35) models the time evolution of any kind of dynamical system. We denote $\mathbf{q}_s$ the steady-state solution of system (1.35). When this solution is reached, $\dot{\mathbf{q}}_s = \mathbf{F}(\mathbf{q}_s) = 0$.

The first step of the establishment of the SFD method is to introduce a linear forcing term on the right-hand side of (1.35). This term contains a scalar control coefficient and a target towards which the solution will be driven to. We obtain the new problem formulation

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}) - \chi(\mathbf{q} - \mathbf{q}_s), \tag{1.36}$$

where χ is the control coefficient and $\mathbf{q}_s$ is the target steady-state to reach. This stabilisation technique is called proportional feedback control and is commonly used

in control theory [26].

There is no generation of an artificial steady-state since when $\mathbf{q} = \mathbf{q}_s$, the forcing term vanishes. Hence a steady solution of (1.36) is also a steady solution of (1.35). In practice, especially for real flow problems, the steady-state is generally not known *a priori*. Then $\mathbf{q}_s$ can not be chosen as the target to reach.

The second step of the establishment of the SFD method is to introduce a new target solution $\bar{\mathbf{q}}$ which is defined as a modification of $\mathbf{q}$ with reduced temporal fluctuations (*i.e.* a temporally low-pass filtered solution). By damping the most dangerous frequencies, the corresponding instabilities are extinguished. This idea was originally introduced by Pruett *et al.* [27, 28] in their attempt to develop a temporal filtered model for large-eddy simulations.

The filtered quantity $\bar{\mathbf{q}}$ is solution of the first order low-pass time filter

$$\dot{\bar{\mathbf{q}}} = \frac{\mathbf{q} - \bar{\mathbf{q}}}{\Delta}, \tag{1.37}$$

where Δ is the filter width.

System (1.37) is the differential form of the low-pass time filter. Such a filter cuts off the high frequencies but does not affect the lower ones. In control theory, filters are usually defined by their transfer function. The gain of filter (1.37) is 1 and its cut-off frequency is $1/\Delta$. Then the transfer function of this filter is

$$\frac{\bar{\mathbf{q}}}{\mathbf{q}} = \frac{1}{1 + \mathrm{i}\omega\Delta}. \tag{1.38}$$

The final step of the establishment of the SFD method is to replace $\mathbf{q}_s$ in (1.36) by the filtered quantity $\bar{\mathbf{q}}$. Then we obtain the augmented system

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}) - \chi(\mathbf{q} - \bar{\mathbf{q}}), \\ \dot{\bar{\mathbf{q}}} = \frac{\mathbf{q} - \bar{\mathbf{q}}}{\Delta}. \end{cases} \tag{1.39}$$

This system is the original time continuous formulation of the SFD method as it was first introduced [10]. The target solution $\bar{\mathbf{q}}$ changes all along the algorithm execution and the steady-state solution of (1.35) is reached when $\mathbf{q} = \bar{\mathbf{q}}$.

System (1.39) can be implemented into an existing code by adding a forcing term into the original time-marching scheme and advancing the linear equation

(1.37). However this approach may prove very complicated because modifying a Navier-Stokes solver is never a trivial task. In Chapter 2 we present an alternative formulation of the SFD method which allows us to implement it avoiding any modification of an already existing unsteady code.

Note that the convergence of this method depends on an "appropriate choice" of the control coefficient χ and the filter width Δ. These parameters play a key role in the ability of the SFD method to reach a steady-state solution. A procedure to define parameters which ensure convergence of the method is proposed in Chapter 3.

1.4 Global linear stability analysis

Flow stability theory is an essential topic in fluid dynamics. Controlling the evolution of a flow is crucial in most industries. For example in aeronautics, engineers are interested in designing wings that postpone as far as possible transition to turbulence. Formula 1 engineers aim to generate vortices with specific dimensions and orientate their evolution. An identification of the stability properties of such flows could help resolving these challenging problems.

Linear stability theory aims to determine the minimum critical parameter (typically the Reynolds number for incompressible flows) above which a specific initial condition of infinitesimal amplitude grows exponentially in time [29]. With this definition, a flow is said to be linearly stable if it returns to its original state when infinitesimal perturbations are introduced. On the other hand, the perturbation of an unstable flow will generate an irreversible change in the flow pattern.

Historically, until the generalisation of high performances computers, the standard approach to quantify flow instabilities was the local linear stability theory [30, 31]. This technique is base on geometrical assumptions to reduce the order of a three-dimensional eigenvalue problem into a set of one-dimensional problems. Typically, the flows studied are planar (*i.e.* invariant in the spanwise direction) or axisymmetric (*i.e.* invariant in the azimuthal direction) with very slow variation in

the streamwise direction. The simplified one-dimensional eigenvalue problems can then be solved very quickly numerically or even analytically. Local stability analysis works very well but reaches a limit for flows presenting rapid evolutions in the streamwise direction. Nowadays, this technique is mainly used as a diagnostic tool.

Throughout this thesis, when we discuss the stability properties of a flow, we refer to a global linear stability analysis. In opposition to local theory, global linear stability theory refers to the analysis of the eigenspectrum of an operator without constricting the linearised flow to evolve in any specific direction. In two-dimensions such analysis are generally called Bi-Global and Tri-Global in three-dimensions [31, 32].

1.4.1 General presentation

In this section we present an overview of the main aspects of global linear stability problems in fluid dynamics. The starting point is to define a base flow $\mathbf{U}$ which is a solution of the incompressible Navier-Stokes equations (1.24). This base flow can be time-dependent (*e.g.* Floquet stability analysis requires a time-periodic base flow) but it should be a genuine solution of the fluid motion equations to mathematically allow linearisation of the fluid system. For this study, we consider steady base flows. In this section, we assume that a steady-state solution of (1.24) is known.

Subject to appropriate initial and boundary conditions, the evolution of infinitesimal perturbations $\mathbf{u}'$ of the base flow $\mathbf{U}$ is governed by the linearised Navier-Stokes equations

$$\begin{cases} \partial_t \mathbf{u}' = -\left(\mathbf{U}\cdot\nabla\right)\mathbf{u}' - \left(\mathbf{u}'\cdot\nabla\right)\mathbf{U} - \nabla p' + Re^{-1}\nabla^2\mathbf{u}', \\ \nabla\cdot\mathbf{u}' = 0. \end{cases} \tag{1.40}$$

A linear evolution operator $\mathcal{A}$ can be defined from (1.40). Hence an initial perturbation $\mathbf{u}'(0)$ evolves forward in time such that

$$\mathbf{u}'(t) = \mathcal{A}(\mathbf{U}, t)\mathbf{u}'(0). \tag{1.41}$$

As the base flow $\mathbf{U}$ is steady, system (1.41) is autonomous (*i.e.* $\mathcal{A}$ is time-invariant) and the normal mode solutions are of the form

$$\mathbf{u}'(\mathbf{x},t) = \exp(\mu_j t)\hat{\mathbf{u}}_j(\mathbf{x}) + \text{c.c.}, \tag{1.42}$$

where $\hat{\mathbf{u}}_j$ are the eigenmodes and μ_j are the eigenvalues (both are, in general, complex). We can define the growth rate σ_j and the frequency f_j such that $\mu_j = \sigma_j + \mathrm{i}f_j$. As the base flow is steady, we can fix t to be constant and equal to a parameter T. This parameter T is usually chosen of order unity [33] but one has to be careful not to take a value greater than the period of the leading eigenvalue in order to avoid aliasing problems. In other words, T must satisfy the Nyquist criterion in order to accurately capture the leading eigenmode frequency. Generally, this frequency is not known before computing stability analysis, then the upper bound of T is hard to approximate *a priori*.

We can now write the formulation of an eigenvalue problem for the operator $\mathcal{A}(\mathbf{U},T)$ such that

$$\mathcal{A}(\mathbf{U},T)\hat{\mathbf{u}}_j = \lambda_j\hat{\mathbf{u}}_j, \quad \text{with} \quad \lambda_j = \exp(\mu_j T). \tag{1.43}$$

Linear stability (or instability) of the base flow $\mathbf{U}$ is determined by the dominant eigenvalue of $\mathcal{A}(\mathbf{U},T)$ (*i.e.* the one of largest modulus). If there exists at least one λ_j such that $|\lambda_j| > 1$, then infinitesimal perturbations of the base flow grow exponentially in time. Hence $\mathbf{U}$ is said to be linearly unstable. In opposition, if all the eigenvalues verify $|\lambda_j| < 1$, then infinitesimal perturbations decay in time and the base flow is linearly stable. Most of the time $|\lambda_j| = 1$ indicates a bifurcation point (*i.e.* a change in stability state). Note that instead of evaluating the modulus of the eigenvalues λ_j, one can also look at the sign of the corresponding growth rate σ_j. A negative growth rate corresponds to a linearly stable base flow, and a positive growth rate corresponds to a linearly unstable one.

Computing the eigenvalues of $\mathcal{A}(\mathbf{U},T)$ is not trivial and several kinds of algorithms exist. In this thesis we used a modified version of the Arnoldi iteration method introduced by Barkley *et al.* [33] and summarized in Sec. 1.4.2.

1.4.2 Modified Arnoldi method

To numerically evaluate the eigenvalues of the linear operator $\mathcal{A}(\mathbf{U}, T)$, the time-stepper approach [9] is used. Similarly as the steady-state solver presented in Sec. 1.3.1, the idea is to adapt an existing unsteady code (which has a linearised Navier-Stokes operator implemented) to perform stability analysis of a given base flow $\mathbf{U}$.

We summarise here the main aspects of the modified Arnoldi iteration method used throughout this thesis to compute flow stability analysis. For a complete description of the method, please refer to Barkley's work [33]. The key feature of this method is to project the operator $\mathcal{A}(\mathbf{U}, T)$ into a lower dimensional Krylov subspace. If we consider a non-zero initial vector $\mathbf{u}_0$ of unit norm, we can define a sequence $\mathbf{T}_{k+1}$ of normalized vectors such that

$$\mathbf{T}_{k+1} = [\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ \dots \ , \mathbf{u}_k] = \left[\mathbf{u}_0, \frac{\mathcal{A}(\mathbf{U}, T)\mathbf{u}_0}{\alpha_1}, \frac{\mathcal{A}(\mathbf{U}, T)\mathbf{u}_1}{\alpha_2}, \ \dots \ , \frac{\mathcal{A}(\mathbf{U}, T)\mathbf{u}_{k-1}}{\alpha_k}\right], \tag{1.44}$$

where the α_j are selected to satisfy $||\mathbf{u}_j|| = 1$.

The Krylov subspace onto which $\mathcal{A}(\mathbf{U}, T)$ is projected is spanned by the sequence $\mathbf{T}_k$. The generation of this sequence actually means executing the linear Navier-Stokes solver for T time units with $\mathbf{u}_j$ for initial condition. This does not require any modification of the existing code but it is the most demanding part of the algorithm in terms of computational resources. The following algebraic operations need to be implemented but their cost is negligible.

Once a sequence $\mathbf{T}_{k+1}$ is constructed, a QR decomposition is performed and the elements of the corresponding upper triangular matrix $\mathbf{R}_{k+1}$ are used to define an upper Hessenberg matrix $\mathbf{H}$. The dominant eigenvalue of $\mathbf{H}$ approximates the dominant eigenvalue of $\mathcal{A}(\mathbf{U}, T)$ and it is possible to evaluate the approximation error. The advantage is that the eigenvalues of $\mathbf{H}$ can be easily evaluated with the help of the LAPACK library [34].

To obtain an accurate evaluation of the dominant eigenvalue, this process is iterated by constructing a new sequence $\mathbf{T}_{k+1}$. To avoid excessive storage requirements, we define a maximum number k_{max} of vectors contained in this sequence. At each

iteration, the oldest vector of the sequence is replaced by the application of the operator $\mathcal{A}(\mathbf{U}, T)$ on the newest vector of the sequence. This feature explains why this method is called "modified" Arnoldi, only a limited number of vectors is stored not the whole sequence. Through this iterative process, the dominant eigenvalue of $\mathbf{H}$ converges towards the dominant eigenvalue of $\mathcal{A}(\mathbf{U}, T)$ in a relatively limited number of steps for low computational cost and memory requirements.

Eventually, note that to correctly set up this algorithm, the inflow boundary conditions of the non-linear problem have to be modified to be homogeneous Dirichlet when $\mathcal{A}(\mathbf{U}, T)$ is applied to a field $\mathbf{u}_j$.

This algorithm has been implemented into the Nektar++ spectral/*hp* element framework (see Sec. 1.6). As the eigenmode solutions are expected to be smooth, using high-order methods allows us to obtain very accurate results. Rocco [35] showed that this implementation agrees very well with ARPACK (ARnoldi PACKage [36]). The differences between the dominant eigenvalues obtained with the two methods were found to be smaller that 10^{-5}.

Note that, when an adjoint linearised system is implemented, this algorithm can be adapted to evaluate the influence of non-normal modes [29]. This approach was adopted by Cantwell *et al.* to compute transient growth analysis of a flow through a pipe with abrupt geometry changes [37]. Such investigations can prove very important because some finite perturbations could push the solution towards the basin of attraction of a stable oscillator even though a linear analysis might conclude that a fixed point is asymptotically stable.

1.5 Stabilisation methods

In this section we present techniques to stabilise the spectral/*hp* element method applied to the incompressible Navier-Stokes equations. In Sec. 1.5.1 we explain how we can lower numerical errors due to the evaluation of nonlinear combination of polynomial functions. A procedure to reduce oscillations that may appear in the higher modes of high-order expansion methods is presented in Sec. 1.5.2. Finally,

in Sec. 1.5.3 we present a boundary condition that maintains the accuracy of the spectral/*hp* element method on the outflow of truncated unbounded domains.

1.5.1 De-aliasing technique

Aliasing effects arise in many domains. It is characterised by the fact that several signals (*i.e.* sinusoidal functions) can fit a sample of discrete points. This is for example responsible of the *wagon-wheel effect* in western films, where carts' wheels seem to be turning backwards. This phenomenon also appears in computational fluid dynamics.

Most numerical solvers use a number of quadrature points necessary to exactly integrate linear combination of polynomial functions. However, when using the velocity-correction scheme to solve the incompressible Navier-Stokes equations, a quadratic combination of the solution must be computed to evaluate the convective contributions (see Eq. (1.25a)). This inner product is "under-integrated", which means that the number of quadrature points used is too low to ensure exact integration of the non-linear term. Hence an *aliasing error* is introduced [38].

For numerical simulations that are sufficiently resolved, the aliasing error does not impact the robustness of the computations. However when coarse meshes are used, this error is not negligible. In the worst case this error builds-up in time and the modal energy contained in the higher modes grow, generating instabilities that lead the simulation to abruptly diverge.

Several rules can be applied within the framework of spectral/*hp* element methods to eliminate the aliasing error [39]. De-aliasing techniques are necessary when studying very large cases, where increasing the resolution would be too prohibitively expensive. In this study, the de-aliasing technique used consists in adding quadrature points to all the elements of the computational domain whenever the non-linear term is evaluated. Details of the implementation may be found in [40]. This method increases the computational cost but it enables stabilisation of simulations that would diverge without a specific treatment of the non-linear term. It also enables to increase accuracy of the solution in regions where non-linear effects are important.

1.5.2 Spectral vanishing viscosity

The *spectral vanishing viscosity* (SVV) method, originally introduced by Tadmor in 1989 [41], is a stabilisation technique especially designed for high-order spatial discretisation schemes. It is based on the idea that spurious oscillations of the high-order modes could be damped away without harming the physics of the lower modes. To stabilise the simulation, extra viscosity is added to the higher modes of the expansion basis.

When applied to the Navier-Stokes equations, and specifically to the velocity-correction scheme, the SVV method consists in adding $SVV(\mathbf{u}^{n+1})$ to the right-hand side of (1.25c), where the SVV function is defined by

$$SVV(\mathbf{u}) = \nu_{SVV} \frac{\partial}{\partial x}\left(K * \frac{\partial \mathbf{u}}{\partial x}\right), \tag{1.45}$$

where ν_{SVV} is the constant extra viscosity, K is the SVV kernel and $*$ denotes the convolution operator.

The kernel K governs the quantity of viscosity which is added to each mode. We use the kernel definition given by Kirby & Sherwin [42] which is

$$K(m) = \begin{cases} \exp\left(-\frac{(M-m)^2}{(M_{cut}-m)^2}\right), & m > M_{cut}, \\ 0, & m \leqslant M_{cut}, \end{cases} \tag{1.46}$$

where $M = P + 1$ is the total number of modes of the expansion basis, m is the current mode and M_{cut} defined the cutoff mode number below which the viscosity remains unchanged. For example, Fig. 1.2 shows this kernel for a $M = 10$ and $M_{cut} = 3$. Note that with this kernel, the SVV method will nearly add no viscosity to the fourth and fifth modes.

To illustrate the influence of the SVV method, we apply it to a two-dimensional diffusion equation $\partial_t \mathbf{u} = \nu \nabla^2 \mathbf{u} + SVV(\mathbf{u})$ on $(x, y) \in [0, 2] \times [0, 2]$ with $\nu = 10^{-5}$. The simulation was computed with 16 expansion modes (*i.e.* $M = 16$ and $P = 15$) on a quadrilateral grid with 16 elements and the initial condition was chosen to be $u(x, y, t = 0) = \sin(\pi x)\sin(\pi y)$. A discontinuity is introduced at the initial step such that $u(0.65, 0.70, t = 0) = 2.0$. Fig. 1.3 shows the solution of this equation at

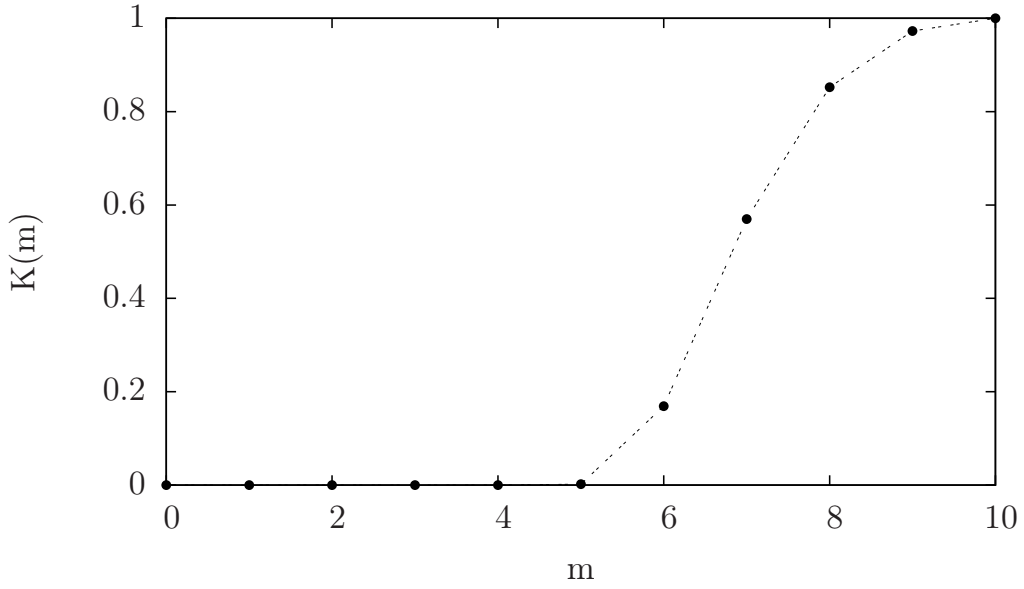


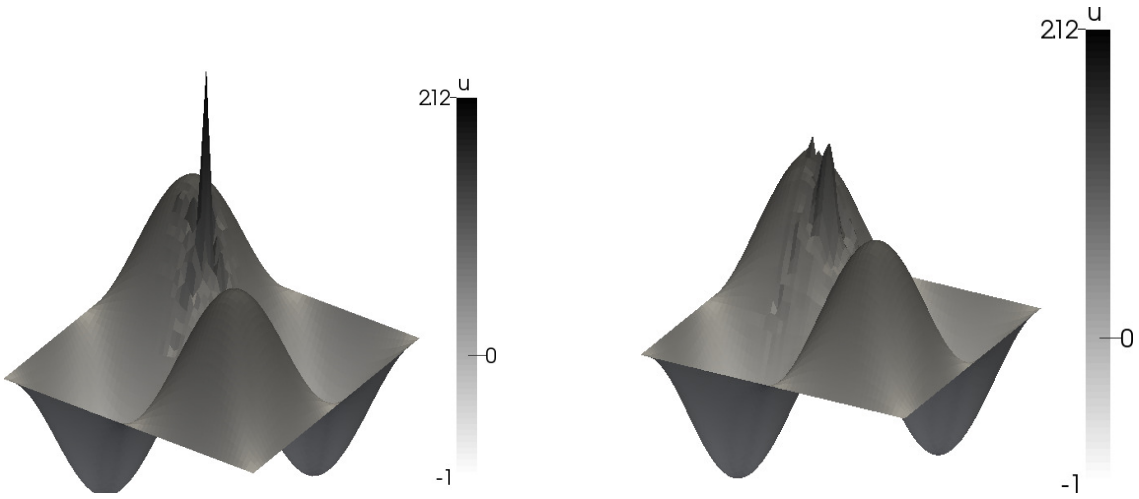
Figure 1.2: SVV kernel for $M = 10$ and $M_{cut} = 3$. This curve represents the percentage of viscosity added on each mode. On the tenth mode, ν_{SVV} is added.

$t = 0.1$ without SVV and with SVV for $M_{cut} = 10$ and $\nu_{SVV} = 1000 \times \nu$. We observe that the discontinuity is strongly damped when the SVV is switched on.

The computational cost of the SVV method for a Navier-Stokes solver is practically negligible because is only consists in evaluating $SVV(\mathbf{u}^{n+1})$ every time step. This method enables stabilisation of high Reynolds number flows. However it must be used carefully. If too much viscosity is added on too many modes, the physics of the problem studied may be drastically changed. As a rule of thumb, we can say that to efficiently use the SVV method, one should add the fewest viscosity as possible on the fewest modes as possible that enable computation stabilisation.

1.5.3 Boundary condition for unbounded truncated domains

The last stabilisation technique presented in this section consists in an alternative formulation of the outflow boundary condition. When unbounded incompressible flows are studied at moderate to high Reynolds numbers, the position of the outflow boundary is a question that needs to be carefully addressed. If strong vortices are



(a) (b)

Figure 1.3: Illustration of the influence of the SVV method for $M_{cut} = 10$ and $\nu_{SVV} = 0.1$. Evolution of the diffusion equation at $t = 0.1$ without SVV (a) and with SVV (b).

present within the flow, the domain must be large enough in order to give them space to be dissipated. Otherwise this may lead to erroneous results or, in some cases, to uncontrolled growth of the energy in the domain and to abrupt divergence of the simulation.

Increasing the size of the simulation is not always an option. As the Reynolds number increases, the computational domain should proportionally be increased too. This imposes a very strong limitation in terms of computational cost. A very long domain may waste computational resources on regions where the physics of the problem is of no interest.

Dong *et al.* [43] proposed an alternative outflow boundary condition that keeps the simulation accurate and stable even when strong vortices exit the computational domain. We denote $\partial\Omega_{out}$ the boundary where the outflow condition is defined. On $\partial\Omega_{out}$ we impose

$$-p\mathbf{n} + \nu\mathbf{n}\cdot\nabla\mathbf{u} - \left[\frac{1}{2}|\mathbf{u}|^2 S_0(\mathbf{n}\cdot\mathbf{u})\right]\mathbf{n} = 0, \tag{1.47}$$

where $\mathbf{n}$ is the outward-pointing unit vector normal to $\partial\Omega_{out}$. The scalar function

S_0 is a smoothed step function defined by

$$S_0(\mathbf{n} \cdot \mathbf{u}) = \frac{1}{2}\left[1 - \tanh\left(\frac{\mathbf{n} \cdot \mathbf{u}}{U_\infty \delta}\right)\right], \tag{1.48}$$

where U_∞ is the characteristic velocity and δ is a constant positive real number representing the smoothness of the step function. The smaller δ, the sharper S_0.

The authors who introduced this outflow boundary condition showed that it enabled to obtain satisfying results when considering the flow past a square cylinder at $Re = 100$. They showed that the lift and drag coefficients are really close to one another when the outflow condition is situated very close to the cylinder (at $4.5D$, D being the square sides length) or further away downstream (at $24.5D$). Dong *et al.* also showed that for the flow past a square cylinder at $Re = 10,000$, this type of outflow condition avoids abrupt divergence of the simulation.

1.6 Nektar++

Nektar++ [2, 3] is an open source library currently being developed at Imperial College London under the supervision of Prof. Spencer Sherwin (Imperial College, Department of Aeronautics) and Prof. Mike Kirby (University of Utah, School of Computing). It is a tensor product based finite element package written in C++. Nektar++ implements the spectral/*hp* element method and both continuous and discontinuous Galerkin operators can be used along with modal or nodal expansion basis. It can solve problems in one, two or three dimensions and several different equation systems can be solved, including the incompressible and compressible (viscous) Navier-Stokes equations. The code supports mixed curved meshes, hence computational domains composed of triangles and quadrilaterals could be used in two-dimensions or composed of tetrahedra, prisms and hexahedra in three-dimensions. Nektar++ also enables to execute massively parallel simulations. Some developers of the group are using ARCHER, which is the most powerful supercomputer in the United Kingdom (2015), to simulate the development of a wingtip vortex at $Re = 1.2 \cdot 10^6$ using 2000 CPUs [44] or the flow past a Formula 1 front wing at $Re = 2.2 \cdot 10^5$ using 4000 CPUs (see Fig. 1.6).

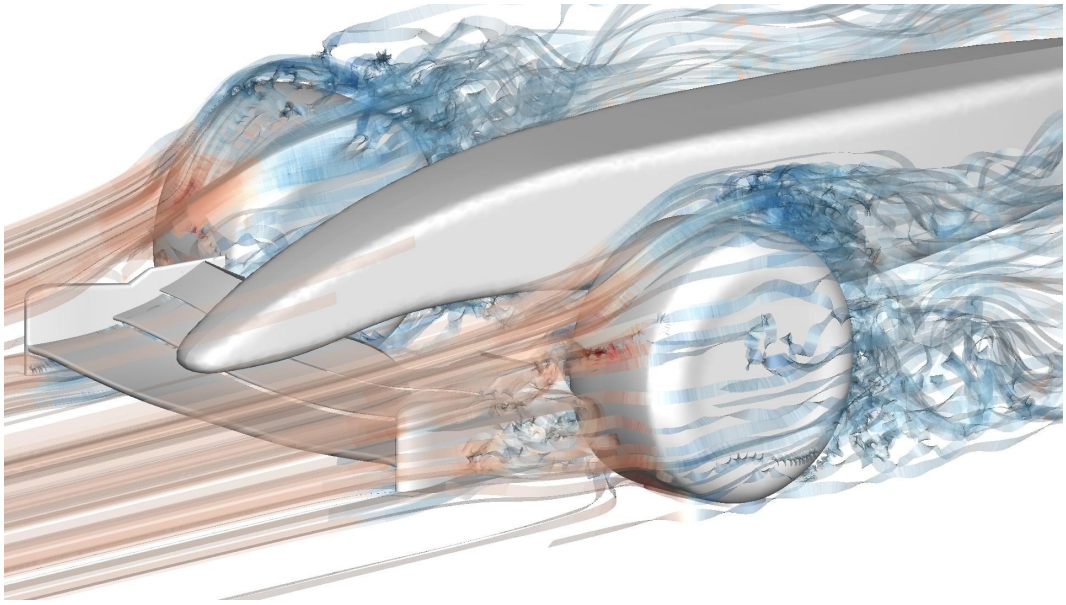


Figure 1.4: Incompressible flow past the front section of a Formula 1 racing car at $Re = 2.2 \cdot 10^5$ (based on the wing's main plan's chord). Streamlines show the flow trajectory and are coloured by pressure. The simulation has 18 million degrees of freedom at polynomial order $P = 3$ and uses the spectral vanishing vorticity technique. Figure courtesy to Jean-Eloi Lombard.

Throughout this thesis, we use Nektar++ to solve the incompressible Navier-Stokes equations. They are discretised in space using the continuous Galerkin method (*i.e.* with continuity enforced at the elements boundaries) highlighted in Sec. 1.1 with a modified modal hierarchical expansion basis. The time evolution of the flow problem is addressed using the second-order velocity correction scheme presented in Sec. 1.2. The steady-state solver present in Nektar++ implements the selective frequency damping method summarised in Sec. 1.3.2. However, unlike the authors who introduced this method [10], this version of the SFD method is not implemented by modifying the time-integration scheme. An alternative formulation that is presented in Chapter 2 is implemented in Nektar++. Finally, the modified Arnoldi iteration method described in Sec. 1.4.2 is used to realise Bi- and Tri-Global stability analysis.

CHAPTER 2

ENCAPSULATED SELECTIVE FREQUENCY DAMPING METHOD

To compute global linear stability analysis of a flow field, a base flow must be defined in advance. Several choices could be made to select this base flow. It is a common practice to use a time-averaged unsteady solution or a RANS solution for the base flow. Even though these choices may well predict the shedding frequency of a flow past a bluff body, they may also return growth rates which are not relevant to the physical solution [45]. More importantly, these base flows are not genuine solutions of the Navier-Stokes equations, hence the linearisation of the Navier-Stokes operator around such fields is not mathematically appropriate. As linearisation of the fluid motion equations around a base flow is the starting point of stability analysis, the base flow definition is crucial.

Throughout this thesis, a steady-state solution is used as base flow to compute stability analysis. This choice of base flow is fitting because linearisation of the Navier-Stokes operator is now performed around a genuine solution to the fluid system. When stable flows are studied, finding a numerical steady-state state is trivial: we only have to execute a flow solver and wait long enough until the flow

becomes constant in time. However stable flows generally present little interest and most of the time unstable flows are considered for industrial applications. Obtaining an unstable steady-state solution of the Navier-Stokes equations is a significant challenge though. The problem of finding an unstable fixed point of a non-linear system has to be addressed.

Newton's methods are commonly applied to find unstable steady-state solutions of the Navier-Stokes equations (see Sec. 1.3.1). The quadratic convergence of these methods is guaranteed if a good initial guess of the solution is used. For challenging flow problems at high Reynolds numbers, a continuation method is necessary to go through the many bifurcations that may occur before finding the required steady-state solution.

To obtain unstable equilibria of the Navier-Stokes equations, we use the selective frequency damping (SFD) method which is an efficient alternative to Newton's methods. Since it was introduced by Åkervik *et al.* [10] in 2006 (see Sec. 1.3.2), this method has been successfully applied for many different applications. For example, it was used to obtain unstable steady incompressible flow past a shallow cavity [46, 47], a square [48], a cylinder [49] and a sphere [50]. This method was also used to study the stability of laminar separation bubble [51] or the stability of roughness-induced boundary layer flows [52]. Furthermore the research group who introduced the SFD method extensively used it to study the global stability of an incompressible jet in a cross flow [53, 54, 55, 56, 57]. The SFD method was also applied in the study of supersonic mixers [58] to obtain unstable equilibria of the compressible Navier-Stokes equations. Combustion problems, such as the stability of round fuel jets [59] or flames [60, 61], were studied as well, using a base flow computed via the SFD method.

To our knowledge, users of the SFD method have always implemented it by modifying the existing time-marching code. For example, the group who introduced the SFD method in 2006, implemented it into the spectral-element code Nek5000 [62] by adding an extra term to the fourth-order Runge-Kutta time-integration scheme. A similar approach was adopted by other researchers who wanted to implement the

SFD method into their own codes. Most of the SFD users directly used Nek5000 though.

Implementing the SFD method by modifying a time-integration scheme requires to modify the existing Navier-Stokes solver at a very low level. The time-integration scheme is duplicated and modified to integrate the feedback control and the first order filter. If the existing code has several time-integration schemes implemented, they all have to be modified.

The method presented in this chapter has been developed with the intention of implementing a steady-state solver into Nektar++ in the least intrusive way possible. This goal was achieved through a mathematical reformulation of the SFD method. We called the resulting algorithm "encapsulated" SFD method. This chapter is based around the paper entitled "Encapsulated formulation of the selective frequency damping method", published in *Physics of Fluids* in March 2014 [63].

In Sec. 2.1 we detail the steps leading to this encapsulated formulation. In Sec. 2.2, we introduce a simple one-dimensional problem to analyse the stability properties of the method. Finally, the unstable steady-state obtained by SFD and a stability analysis of the incompressible flow past a cylinder at $Re = 100$ is presented in Sec. 2.3.

2.1 Encapsulated formulation

In this section we present the encapsulated formulation of the selective frequency damping (SFD) method [63]. The key feature of this procedure is to adapt an existing unsteady code in order to obtain a steady-state solver. This method was inspired by Tuckerman & Barkley's time-stepper approach (see in Sec. 1.3.1). The alternative formulation of the SFD method proposed here is obtained using a first-order splitting method. In Sec. 2.1.1 we highlight the main aspects of splitting methods and in Sec. 2.1.2 we detail the encapsulated formulation of the SFD method.

2.1.1 Generalities on splitting methods

Classical splitting methods are based on the idea that it is sometimes more efficient to solve several small problems than one large complex problem. In algebra, the partial fraction decomposition uses this approach. When it comes to solving equation systems, splitting methods consist in dividing the main problem into smaller (simpler) subproblems that can be solved separately using a variety of numerical methods (explicit or implicit). The subproblems are connected via initial conditions though. Typically, on one time step, the final solution of one subproblem is the initial condition for the next one. In this section we shortly present the theoretical aspects of the first order splitting method. Note that in Sec. 1.2 a splitting method is used to establish the velocity-correction scheme which discretises the incompressible Navier-Stokes equations in time.

Splitting methods can be applied to any initial value problem of the form

$$\dot{\mathbf{u}} = \mathbf{G}(\mathbf{u}), \tag{2.1}$$

that can be divided into two (or more) subproblems such as

$$\dot{\mathbf{u}} = \mathbf{G}_1(\mathbf{u}) + \mathbf{G}_2(\mathbf{u}). \tag{2.2}$$

Each subproblem is then considered separately and solved sequentially. Note that for simplicity, only the case where $\mathbf{G}$ is divided into two subproblems is considered here. The generalisation to more sub-processes is straightforward [64].

The sequential operator-splitting method of first order (also called A-B splitting method) is the simplest splitting method. The two subproblems are sequentially solved on the interval $[t^n, t^{n+1}]$ and the final solution of $\dot{\mathbf{u}} = \mathbf{G}_1(\mathbf{u})$ is used as initial condition for $\dot{\mathbf{u}} = \mathbf{G}_2(\mathbf{u})$. If we denote $\mathbf{u}^n_{sp}$ as the splitting solution of (2.2) at the time t^n, then the resolution of this problem is carried out by iterating the sequence:

$$\textbf{1 -} \text{ Solve } \dot{\mathbf{u}}^{(1)} = \mathbf{G}_1(\mathbf{u}^{(1)}) \text{ in } [t^n, t^{n+1}] \text{ with } \mathbf{u}^{(1)}(t^n) = \mathbf{u}^n_{sp}, \tag{2.3a}$$

$$\textbf{2 -} \text{ Solve } \dot{\mathbf{u}}^{(2)} = \mathbf{G}_2(\mathbf{u}^{(2)}) \text{ in } [t^n, t^{n+1}] \text{ with } \mathbf{u}^{(2)}(t^n) = \mathbf{u}^{(1)}(t^{n+1}). \tag{2.3b}$$

After each time step, the splitting solution is updated such that $\mathbf{u}_{sp}^{n+1} = \mathbf{u}^{(2)}(t^{n+1})$. System (2.3) is the general formulation of a first-order time splitting method. It appears clearly that the resolution of (2.3a) and (2.3b) are independent from one another. These subproblems are only linked through initial conditions.

We now present a special case where the operators $\mathbf{G}_1$ and $\mathbf{G}_2$ are both linear. Then (2.2) can be reformulated

$$\dot{\mathbf{u}} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{u}. \tag{2.4}$$

In the interval $[t^n, t^{n+1}]$, Eq. (2.4) can be solved exactly. Its solution is given by

$$\mathbf{u}(t^{n+1}) = e^{(\mathbf{A}+\mathbf{B})\Delta t}\mathbf{u}(t^n). \tag{2.5}$$

If we use the first order splitting method (2.3) to solve equation (2.4) in the interval $[t^n, t^{n+1}]$, the splitting solution is

$$\mathbf{u}_{sp}^{n+1} = e^{\mathbf{A}\Delta t}e^{\mathbf{B}\Delta t}\mathbf{u}_{sp}^{n}. \tag{2.6}$$

The difference between (2.5) and (2.6) is called the *local splitting error*. This error is zero if $e^{(\mathbf{A}+\mathbf{B})} = e^{\mathbf{A}}e^{\mathbf{B}}$, which is true if, and only if, $\mathbf{AB} = \mathbf{BA}$ [1]. In such a case, the first order splitting method is exact. Otherwise, if $\mathbf{A}$ and $\mathbf{B}$ do not commute, an error is introduced. This error is of order $\mathcal{O}(\Delta t^2)$ if the operators $\mathbf{A}$ and $\mathbf{B}$ are bounded. Note that if $\mathbf{G}_1$ and $\mathbf{G}_2$ are not linear, a splitting error is always introduced.

Higher order (of even number) splitting methods also exist [64, 65]. For example, the Strang splitting method [66] is a commonly used second order scheme. The

[1] If $\mathbf{A}$ and $\mathbf{B}$ commute then the binomial formula $(\mathbf{A}+\mathbf{B})^n \sum_{k=0}^{n} \binom{n}{k} \mathbf{A}^k\mathbf{B}^{n-k}$ is true. Hence we can write $e^{(\mathbf{A}+\mathbf{B})} = \sum_{n=0}^{\infty} \frac{1}{n!}(\mathbf{A}+\mathbf{B})^n = \sum_{n=0}^{\infty} \frac{1}{n!} \sum_{k=0}^{n} \binom{n}{k} \mathbf{A}^k\mathbf{B}^{n-k} = \sum_{0\le k\le n<\infty} \frac{1}{k!}\frac{1}{(n-k)!}\mathbf{A}^k\mathbf{B}^{n-k} = \sum_{k=0}^{\infty}\sum_{l=0}^{\infty} \frac{1}{k!}\frac{1}{l!}\mathbf{A}^k\mathbf{B}^l = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{A}^k \cdot \sum_{l=0}^{\infty} \frac{1}{l!}\mathbf{B}^l = e^{\mathbf{A}}e^{\mathbf{B}}$ (with $n-k=l$).
However if $\mathbf{A}$ and $\mathbf{B}$ do not commute, for example if $\mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ and $\mathbf{B} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, it is easy to show that $e^{(\mathbf{A}+\mathbf{B})} \neq e^{\mathbf{A}}e^{\mathbf{B}}$.

mechanism of this method consists of solving the first subproblem on one half of a time step, then solving the second subproblem on the whole time step and finally solving the first subrpoblem on the other half of the time step. This procedure essentially shifts the first order scheme outlined in (2.3) by half a Δt, thus decreasing the splitting error.

In the following section, only the first order splitting scheme is used to establish the encapsulated formulation of the SFD method. We show that a splitting scheme of higher order is not necessary for this method since we are only interested in steady-state solutions.

2.1.2 Encapsulated SFD method

The method proposed in this section is an alternative formulation of the SFD method introduced by Åkervik *et al.* [10] and summarised in Sec. 1.3.2. The aim of this method is to obtain unstable steady-state solution of the dynamical system

$$\dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}). \tag{2.7}$$

As for the time-stepper approach [9], we assume that a numerical solver for (2.7) already exist and we work around this solver, without modifying it, to implement a steady-state solver. To guide the reader through the different steps of how the encapsulated SFD method was developed, we start with the original formulation of the SFD method

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}) - \chi(\mathbf{q} - \bar{\mathbf{q}}), \\ \dot{\bar{\mathbf{q}}} = \frac{\mathbf{q}-\bar{\mathbf{q}}}{\Delta}. \end{cases} \tag{2.8}$$

System (2.8) is split into two parts and reformulated within the framework of the first order splitting method. The first subproblem is simply the dynamical system considered (which can be non-linear) with no condition imposed on $\bar{\mathbf{q}}$ such that

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}), \\ \dot{\bar{\mathbf{q}}} = \mathbf{0}, \end{cases} \quad \Longleftrightarrow \quad \dot{\mathbf{q}} = \mathbf{F}(\mathbf{q}). \tag{2.9}$$

Note that, to match the notations introduced in (2.3), every variable $\mathbf{q}$ and $\bar{\mathbf{q}}$ of system (2.9) should be written with the superscript $^{(1)}$ to highlight the fact that the first subproblem is considered. For clarity, this notation was not used here.

The second subproblem is linear and represents the actions of the feedback control and the first-order low-pass time filter. It is formulated as

$$\begin{cases} \dot{\mathbf{q}} = -\chi(\mathbf{q} - \bar{\mathbf{q}}), \\ \dot{\bar{\mathbf{q}}} = \frac{\mathbf{q}-\bar{\mathbf{q}}}{\Delta}, \end{cases} \iff \begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\bar{\mathbf{q}}} \end{pmatrix} = \begin{pmatrix} -\chi \boldsymbol{I}_n & \chi \boldsymbol{I}_n \\ \boldsymbol{I}_n/\Delta & -\boldsymbol{I}_n/\Delta \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \bar{\mathbf{q}} \end{pmatrix}, \tag{2.10}$$

where $\boldsymbol{I}_n$ is the identity matrix and the subscript n is equal to the dimension of $\mathbf{q}$ (not to be confused with the exponent n used in Sec. 2.1.1 which stands for the current time step).

A linear operator is defined by system (2.10). We denote $\boldsymbol{\mathcal{L}}_n$ this operator such that

$$\boldsymbol{\mathcal{L}}_n = \begin{pmatrix} -\chi \boldsymbol{I}_n & \chi \boldsymbol{I}_n \\ \boldsymbol{I}_n/\Delta & -\boldsymbol{I}_n/\Delta \end{pmatrix}. \tag{2.11}$$

Hence the original SFD method (2.8) can be reformulated

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\bar{\mathbf{q}}} \end{pmatrix} = \mathbf{F}(\mathbf{q}) + \boldsymbol{\mathcal{L}}_n \begin{pmatrix} \mathbf{q} \\ \bar{\mathbf{q}} \end{pmatrix}. \tag{2.12}$$

We now apply the first-order splitting method detailed in (2.3) to solve system (2.12). The two subproblems are then solved separately, using different numerical methods. We recall that the first subproblem is the dynamical system (2.7) for which we are interested in finding a steady-state solution. Therefore it can be solved with the existing code. This is the key feature of the encapsulated SFD method presented in this section. To solve (2.12), only a linear solver for (2.10) has to be implemented because a code to solve (2.7) already exists. Then the existing unsteady solver does not need to be modified and the solver for (2.10) can be plugged into this code.

We introduce the symbolic notation $\boldsymbol{\Phi}$ denoting this existing unsteady code. Hence at the step $(n+1)$, the numerical solution of (2.9) is given by

$$\mathbf{q}^{n+1} = \boldsymbol{\Phi}(\mathbf{q}^n). \tag{2.13}$$

As the solver $\mathbf{\Phi}$ has already been implemented, the resolution of the first subproblem is trivial: it only consists in calling $\mathbf{\Phi}$ on one time step. We now focus on the resolution of the second subproblem. Note that as (2.10) is linear, it can be solved exactly on the time interval $[t^n,\ t^{n+1}]$. Its solution is given by

$$\begin{pmatrix} \mathbf{q}(t^{n+1}) \\ \bar{\mathbf{q}}(t^{n+1}) \end{pmatrix} = e^{\boldsymbol{\mathcal{L}}_n \Delta t} \begin{pmatrix} \mathbf{q}(t^{n}) \\ \bar{\mathbf{q}}(t^{n}) \end{pmatrix}. \tag{2.14}$$

In (2.14), the only difficulty lays in expressing the exponential of a matrix. As the elements of $\boldsymbol{\mathcal{L}}_n$ belong $\mathbb{C}$, which is an algebraically closed field (*c.f.* fundamental theorem of algebra), this linear operator can be decomposed into $\boldsymbol{\mathcal{L}}_n = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$. The matrices $\mathbf{P}$ and $\mathbf{D}$ are defined as

$$\begin{aligned} \boldsymbol{P} &= \begin{pmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \chi \Delta \\ \boldsymbol{I}_n & \boldsymbol{I}_n \end{pmatrix}, \\ \text{and} \quad \boldsymbol{D} &= \begin{pmatrix} \mathbf{0}_n & \mathbf{0}_n \\ \mathbf{0}_n & -\boldsymbol{I}_n(\chi + \frac{1}{\Delta}) \end{pmatrix}, \end{aligned} \tag{2.15}$$

where $\mathbf{0}_n$ is an all-zero square matrix of $n \times n$ dimension.

Then the exponential of the matrix $\boldsymbol{\mathcal{L}}_n$ multiplied by the time step Δt can be written

$$e^{\boldsymbol{\mathcal{L}}_n \Delta t} = \boldsymbol{P} e^{\boldsymbol{D}\Delta t} \boldsymbol{P}^{-1}. \tag{2.16}$$

The expanded expression of (2.16) is

$$e^{\boldsymbol{\mathcal{L}}_n \Delta t} = \frac{1}{1+\chi\Delta} \times \begin{pmatrix} \boldsymbol{I}_n + \chi\Delta \boldsymbol{I}_n e^{-(\chi+\frac{1}{\Delta})\Delta t} & \chi\Delta \boldsymbol{I}_n [1 - e^{-(\chi+\frac{1}{\Delta})\Delta t}] \\ \boldsymbol{I}_n - \boldsymbol{I}_n e^{-(\chi+\frac{1}{\Delta})\Delta t} & \chi\Delta \boldsymbol{I}_n + \boldsymbol{I}_n e^{-(\chi+\frac{1}{\Delta})\Delta t} \end{pmatrix}. \tag{2.17}$$

According to splitting method theory, the initial conditions of the second subproblem must be the final solutions of the first subproblem. Therefore, at each time step, the output values from the first subproblem (2.13) are used as initial values for the second subproblem (2.14). Since the first subproblem does not include the filtered quantity $\bar{\mathbf{q}}$, the time discrete formulation of the encapsulated SFD method using a first-order splitting method and an exact resolution of the second subproblem

is given by the sequence

1 - Call unsteady solver: $\mathbf{q}^* = \mathbf{\Phi}(\mathbf{q}^n)$, (2.18a)

2 - Apply linear operator $e^{\mathcal{L}_n \Delta t}$ to $\mathbf{q}^*$ and $\bar{\mathbf{q}}$: $\begin{pmatrix} \mathbf{q}^{n+1} \\ \bar{\mathbf{q}}^{n+1} \end{pmatrix} = e^{\mathcal{L}_n \Delta t} \begin{pmatrix} \mathbf{q}^* \\ \bar{\mathbf{q}}^n \end{pmatrix}$. (2.18b)

This method is said to be *encapsulated* because the existing solver $\mathbf{\Phi}$ remains unchanged and only its output is used in the linear expression (2.18b). Then the encapsulated SFD method can be implemented as a wrapper function around the solver $\mathbf{\Phi}$ which can then act as a "black box". Any type of temporal and spatial discretisation can be used in $\mathbf{\Phi}$. Note that this method can be applied to any kind of dynamical system, not only within the framework of computational fluid dynamics.

For the implementation, $e^{\mathcal{L}_n \Delta t}$ does not need to be constructed. It is a $2n \times 2n$ matrix consisting of top-left, top-right, bottom-left and bottom right diagonal $n \times n$ blocks. We introduce the scalar coefficients a, b, c and d respectively equal to the only non-zero element of the top-left, top-right, bottom-left and bottom-right blocks of $e^{\mathcal{L}_n \Delta t}$ such that

$$\begin{aligned} a &= \frac{1 + \chi\Delta e^{-(\chi + \frac{1}{\Delta})\Delta t}}{1 + \chi\Delta}, \\ b &= \frac{\chi\Delta[1 - e^{-(\chi + \frac{1}{\Delta})\Delta t}]}{1 + \chi\Delta}, \\ c &= \frac{1 - e^{-(\chi + \frac{1}{\Delta})\Delta t}}{1 + \chi\Delta}, \\ d &= \frac{\chi\Delta + e^{-(\chi + \frac{1}{\Delta})\Delta t}}{1 + \chi\Delta}. \end{aligned} \tag{2.19}$$

The step by step implementation of the encapsulated SFD method is detailed in Algorithm 1, with the subscript i denoting the spatial discretisation of $\mathbf{\Phi}$.

We recall that the steady-state solution of the SFD method is reached when $\mathbf{q} = \bar{\mathbf{q}}$. Therefore Algorithm 1 is iterated until $||\mathbf{q} = \bar{\mathbf{q}}||$ becomes smaller than a desired tolerance. Note that in most codes used in computational fluid dynamics, scalar-vector multiplication and vector-vector addition are computed very efficiently. Furthermore the second subproblem is a loop over the degrees of freedom, which is

Algorithm 1 Implementation of the encapsulated SFD method

Choose a control coefficient χ

Choose a filter width Δ

Evaluate the scalar values a, b, c and d

while Steady-state solution not reached **do**

Compute $\mathbf{q}^* := \mathbf{\Phi}(\mathbf{q}^n) \Longleftrightarrow$ First subproblem, existing unsteady solver

Compute $\forall i \quad \begin{cases} \mathbf{q}_i^{n+1} := a\mathbf{q}_i^* + b\bar{\mathbf{q}}_i^n \\ \bar{\mathbf{q}}_i^{n+1} := c\mathbf{q}_i^* + d\bar{\mathbf{q}}_i^n \end{cases} \Longleftrightarrow$ Second linear subproblem

Update variables: $(\mathbf{q}^n,\ \bar{\mathbf{q}}^n) := (\mathbf{q}^{n+1},\ \bar{\mathbf{q}}^{n+1})$

end while

far faster than looping over the elements (*i.e.* finding information about neighbour elements is usually slow) and extremely parallelisable. Hence evaluating the resolution of the second subproblem is negligible in terms of computational cost. The execution of the solver $\mathbf{\Phi}$ is by far the most demanding part of Algorithm 1. Hence we can say that this steady-state solver has nearly identical computational performance to the unsteady solver $\mathbf{\Phi}$.

The encapsulated formulation of the SFD method introduces a splitting error. However, this does not affect the accuracy of the steady-state solution. Indeed, before convergence is reached, the solution of the SFD system (2.8) is not a proper solution of the original equation (2.7) but a controlled version with reduced temporal frequencies. Only at the steady-state, when the forcing term $\chi(\mathbf{q}-\bar{\mathbf{q}})$ vanishes, both systems have identical solutions. Therefore any numerical error occurring before the convergence is reached does not affect the accuracy of the final steady-state solution. In other words, by using a splitting method we introduce an error to the augmented system (2.8) at each time step. Meanwhile we want a steady-state of (2.7) which is obtained when (2.8) reaches convergence. Therefore transient numerical errors of (2.8) are no longer relevant. Hence the accuracy of the final steady-state solution obtained by using the encapsulated SFD method depends only on the accuracy of the existing solver $\mathbf{\Phi}$.

The encapsulated SFD method presented in this section is mainly a mathematical reformulation which allows us to implement the method in a non-intrusive way in an existing code. All the features of the original formulation are preserved. Note that the selection of the control coefficient χ and the filter width Δ has not been addressed yet. The choice of these parameters is crucial because a steady-state solution can not always be obtained for arbitrary values of χ and Δ. In the following section we apply the encapsulated SFD method to a very simple one-dimensional problem in order to give a better understanding the stability properties of this method.

2.2 One-dimensional model

In this section we study the stability of the SFD method. It means that we evaluate its ability to converge towards a steady-state solution. In this context, one should not confuse the *stability of a numerical algorithm* (*i.e.* when it converges or not) and the *stability of a flow field* (*i.e.* its response to small perturbations).

Specific care has to be taken with the selection of the control coefficient χ and filter width Δ. For small problems, typically two-dimensional flows at relatively low Reynolds numbers, it is possible to run several simulations with different parameters for the SFD method hoping that one will reach a steady-state. However if one wants to study challenging flows, with higher requirements for computational resources, this approach is not possible. In this section, we aim to understand the roles of χ and Δ in the convergence (or not) of the SFD method.

A simple one-dimensional problem is studied in order to analyse the influence of the control coefficient and the filter width on the stability of the SFD method. A clear understanding of their role should help users of the SFD method to choose parameters which ensure convergence towards a steady-state. The scalar problem considered is

$$\dot{u} = \gamma u, \tag{2.20}$$

where $\gamma \in \mathbb{C}$. This equation is solved exactly on the interval $[t^n,\ t^n + \Delta t]$ and if we

denote $\alpha = e^{\gamma \Delta t}$ then the solution is

$$u^{n+1} = \alpha u^n. \tag{2.21}$$

In this formulation, the time step Δt is an artificial parameter. Then for the remaining of this section we set $\Delta t = 1$.

Convergence towards the steady-state of (2.21) (*i.e.* $u^{n+1} = u^n = 0$) is guaranteed if and only if $|\alpha| < 1$. The main use of the SFD method is to stabilize unstable modes. Hence this method enables to reach the steady-state of (2.21) when $|\alpha| > 1$. Analysing this simple case allows us to highlight the roles of the parameters χ and Δ in the convergence (or not) of the SFD method.

To apply the encapsulated formulation of the SFD method to (2.21) we define the function $\boldsymbol{\Phi}_1$ such that $\boldsymbol{\Phi}_1(u^n) = \alpha u^n$. Then we introduce the operator $\boldsymbol{\mathcal{L}}_1$ which has the same form as (2.10) with $\boldsymbol{I}_1 = 1$. Considering these definitions, the application of the encapsulated formulation of the SFD method to the linear problem (2.21) is written

$$\begin{aligned}
\begin{pmatrix} u^{n+1} \\ \bar{u}^{n+1} \end{pmatrix} &= e^{\boldsymbol{\mathcal{L}}_1} \begin{pmatrix} \boldsymbol{\Phi}_1(u^n) \\ \bar{u}^n \end{pmatrix}, \\
&= e^{\boldsymbol{\mathcal{L}}_1} \begin{pmatrix} \alpha u^n \\ \bar{u}^n \end{pmatrix}, \\
&= e^{\boldsymbol{\mathcal{L}}_1} \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u^n \\ \bar{u}^n \end{pmatrix}, \\
&= \frac{1}{1+\chi\Delta} \times \begin{pmatrix} \alpha\left(1 + \chi\Delta e^{-(\chi+\frac{1}{\Delta})\Delta t}\right) & \alpha\chi\Delta\left[1 - e^{-(\chi+\frac{1}{\Delta})\Delta t}\right] \\ 1 - e^{-(\chi+\frac{1}{\Delta})\Delta t} & \chi\Delta + e^{-(\chi+\frac{1}{\Delta})\Delta t} \end{pmatrix} \begin{pmatrix} u^n \\ \bar{u}^n \end{pmatrix}, \\
&= \mathcal{M} \begin{pmatrix} u^n \\ \bar{u}^n \end{pmatrix}.
\end{aligned} \tag{2.22}$$

The eigenvalues of $\mathcal{M}$ (denoted λ_1 and λ_2) can be easily evaluated. These eigenvalues are functions of the control coefficient χ, the filter width Δ and the complex number α. To ensure stability of (2.22), and then convergence towards the steady-state of (2.21), the maximum of the eigenvalue magnitudes must be strictly smaller

than one. As α is a problem parameter, we want to be able to choose χ and Δ such that $\max(|\lambda_1|, |\lambda_2|) < 1$.

2.2.1 Limiting behaviour

In this section the limiting behaviour of the encapsulated SFD method applied to (2.21) is studied. The eigenvalues of (2.22) are evaluated when the parameters χ and Δ tend alternatively to zero or to infinity. The expression of the eigenvalue obtained will allow us to analyse the limiting stability of (2.22).

Limit when $\chi \to 0$

$$\begin{aligned} \lim_{\chi \to 0} \lambda_1 &= \alpha, \\ \lim_{\chi \to 0} \lambda_2 &= e^{-1/\Delta}. \end{aligned} \tag{2.23}$$

In this case, both $|\lambda_1|$ and $|\lambda_2|$ are smaller than one if and only if the magnitude of α is smaller than one. Let us recall that (2.21) converges if $|\alpha| < 1$. Hence if the original problem is not converging, applying the SFD method and choosing small control coefficient will not drive the solution towards its steady-state. This result was predictable because when χ tends to zero, the term $\chi(\mathbf{q} - \bar{\mathbf{q}})$ vanishes in (2.8). Then the stability of (2.8) is only governed by the stability of the original system (2.7).

Limit when $\chi \to +\infty$

$$\begin{aligned} \lim_{\chi \to +\infty} \lambda_1 &= 1, \\ \lim_{\chi \to +\infty} \lambda_2 &= 0. \end{aligned} \tag{2.24}$$

We can interpret this result by saying that if the control coefficient χ is chosen to be large, the encapsulated formulation of the SFD method is marginally stable. The steady-state can not be reached but the solution does not abruptly diverge.

Limit when $\Delta \to 0$

$$\begin{aligned} \lim_{\Delta \to 0} \lambda_1 &= \alpha, \\ \lim_{\Delta \to 0} \lambda_2 &= 0. \end{aligned} \tag{2.25}$$

The interpretation of this case is the same as the interpretation of the case $\chi \to 0$. This result was also predictable because a first order filter with a filter width Δ arbitrarily small (*i.e.* a cut-off frequency $1/\Delta$ tending to infinity) does not damp any frequency. Hence the "filtered" quantity $\bar{\mathbf{q}}$ is identical to $\mathbf{q}$ and the forcing term $\chi(\mathbf{q} - \bar{\mathbf{q}})$ vanishes. Note that this specific case is the only one were $\bar{\mathbf{q}} = \mathbf{q}$ does not indicate the presence of a steady-state.

Limit when $\Delta \to +\infty$

$$\begin{aligned} \lim_{\Delta \to +\infty} \lambda_1 &= \alpha e^{-\chi}, \\ \lim_{\Delta \to +\infty} \lambda_2 &= 1. \end{aligned} \tag{2.26}$$

Similar to the case $\chi \to +\infty$, if the filter width Δ is large, the encapsulated formulation of the SFD method is not stable.

Åkervik *et al.* [10] stated that choosing a large χ or a large Δ would make the system evolution very slow, but the SFD method would eventually converge to a steady-state. The results presented here suggest that when these parameters tend to an infinite value, the method is not stable and the steady-state cannot be found.

2.2.2 Stability regions

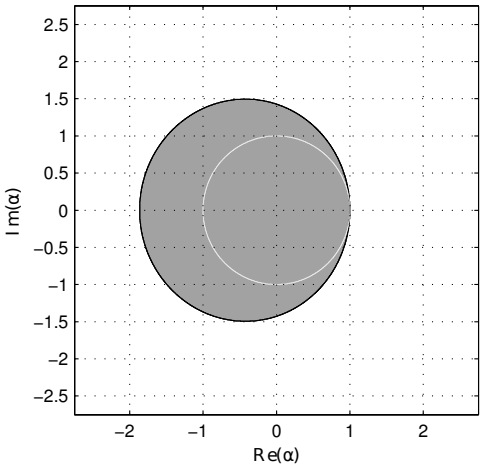
In this section we examine the stability regions of the encapsulated formulations of the SFD method. The goal, for a given control coefficient and filter width, is to identify for which α the one-dimensional problem will converge towards its steady-state. This section focuses only on the influence of χ and Δ. Note that the speed on convergence will be discussed in Sec. 3.1.

Fig. 2.1 consists of four graphs representing complex planes of α at fixed χ for four different values of Δ. For each α the eigenvalues of $\mathcal{M}$ are evaluated and if both eigenvalue magnitudes are smaller than one, the corresponding point on the complex plane is coloured in grey. Therefore the grey areas correspond to the stability regions of (2.22). For example, Fig. 2.1(a) shows that a control coefficient $\chi = 1$ and a filter width $\Delta = 0.5$ will drive (2.22) towards its steady-state for every α chosen within the grey area.

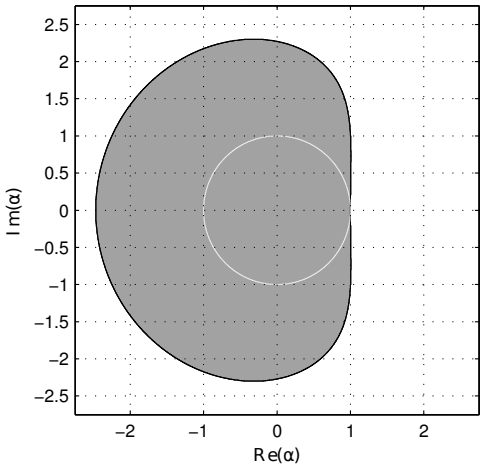
We recall that the one-dimensional system (2.21) is stable if and only if $|\alpha| < 1$ (*i.e.* only if α was situated within the unit disc, delimited by the white circle on Fig. 2.1). The most interesting feature observable in Fig. 2.1 is that the stability region of (2.22) expands beyond the unit disc. These graphs allow us to visualize the fact that the SFD method can stabilize unstable modes. This is achieved without introducing a loss in stability of the original system. Indeed the stability region of the original problem (*i.e.* the unit disc) is inside each stability regions of the SFD method.

Fig. 2.2 also indicates that, for a fixed control coefficient, the stability region of a given filter width includes all the stability regions of lower Δ. Then increasing the filter width does not introduce any loss of stability. It means that if (2.22) is converging towards a steady-state solution for a pair (χ, Δ), then choosing a higher filter width will also enable convergence. The stability region at this higher Δ is also closer to the real horizontal axis. This observation agrees with Massa [58] who stated that increasing Δ at constant χ yields a marginal improvement of the low-frequency stability region. However, this feature is true only until a certain value of Δ. On system (2.26), we showed that when the filter width tends to infinity, the SFD method is not stable.

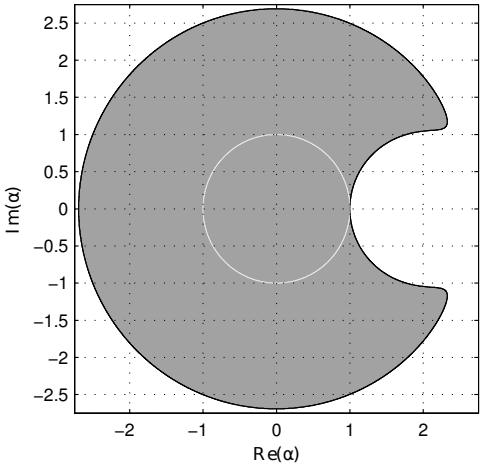
Stability regions for a fixed filter width and various control coefficients are presented on Fig. 2.3 and Fig. 2.4. These figures indicate that for a given filter width, increasing the control coefficient does not change the shape of the region, but only the area covered. Therefore if a pair (χ, Δ) is chosen such that the system (2.22) can reach the steady-state, increasing the control coefficient further can result in



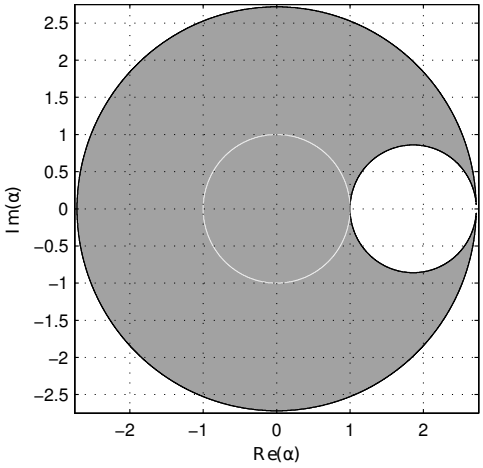
(a) $\Delta = 0.5$



(b) $\Delta = 2$



(c) $\Delta = 20$



(d) $\Delta = 1000$

Figure 2.1: Stability regions for $\chi = 1$ and various Δ. If α is inside the grey area then (2.22) converges towards the steady-state of (2.21). The unit circle (*i.e.* the region where $|\alpha| = 1$) is displayed in white.

destabilising the system (2.22) altogether. Hence, when using the SFD method, a large χ is generally not an appropriate choice. Masa reached a similar conclusion, in [58] it is explained that increasing χ at constant Δ leads to the stabilization of the high-frequency modes, but the destabilization of the low-frequency counterparts.

On every stability region presented in this section, we can notice that if α is real

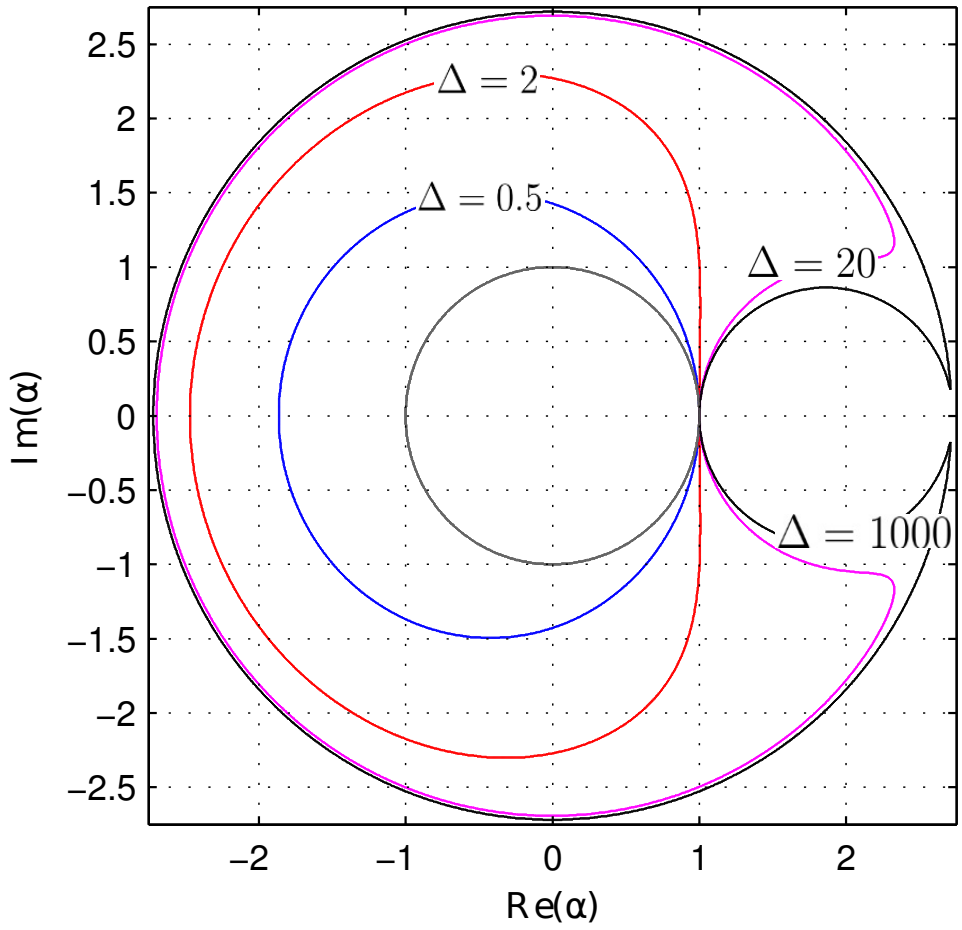
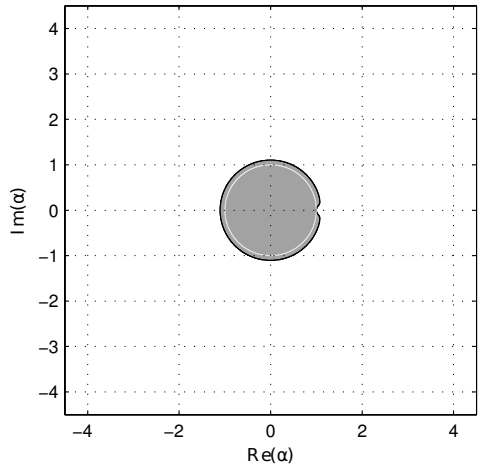


Figure 2.2: Superimposition of the stability regions shown on Fig. 2.1.

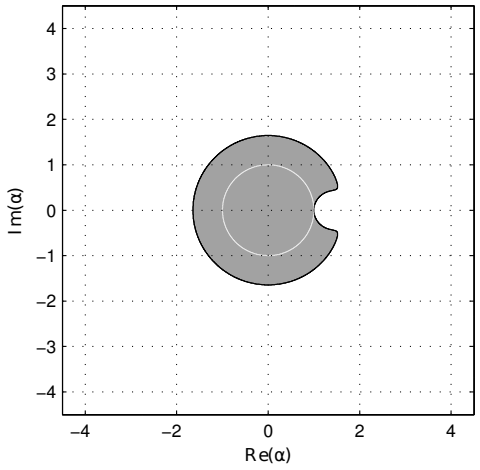
and greater than one it is not possible to find a pair χ and Δ for which (2.22) is stable. These α values correspond to a problem with pure exponential growth of the instability. Hence we can conclude that the stability of the SFD method relies on the oscillatory growth of the problem studied. This behaviour agrees with the result presented by Vyazmina [67] who said that if an unstable eigenvalue is real and positive, there is no frequency to be damped by the SFD method.

Note that when $\chi = 0$ and when Δ tends to zero, the stability region of (2.22) fits exactly the unit circle. This behaviour corresponds to the outcome of the limiting behaviour analysis presented in (2.23) and (2.25).

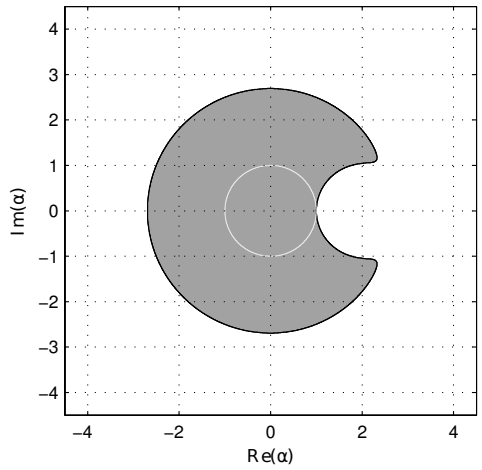
In the following section we apply the encapsulated SFD method to the incom-



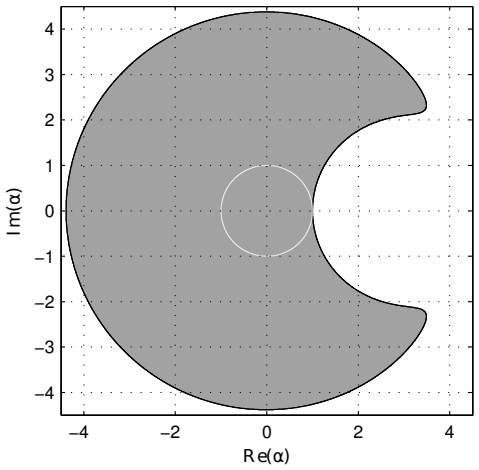
(a) $\chi = 0.1$



(b) $\chi = 0.5$



(c) $\chi = 1$



(d) $\chi = 1.5$

Figure 2.3: Stability regions for $\Delta = 20$ and various χ. If α is inside the grey area then (2.22) converges towards the steady-state of (2.21). The unit circle (*i.e.* the region where $|\alpha| = 1$) is displayed in white.

pressible flow past a cylinder. We show that the stability properties of the one-dimensional model presented here may be used to predict the effectiveness of the SFD method for a more complex problem.

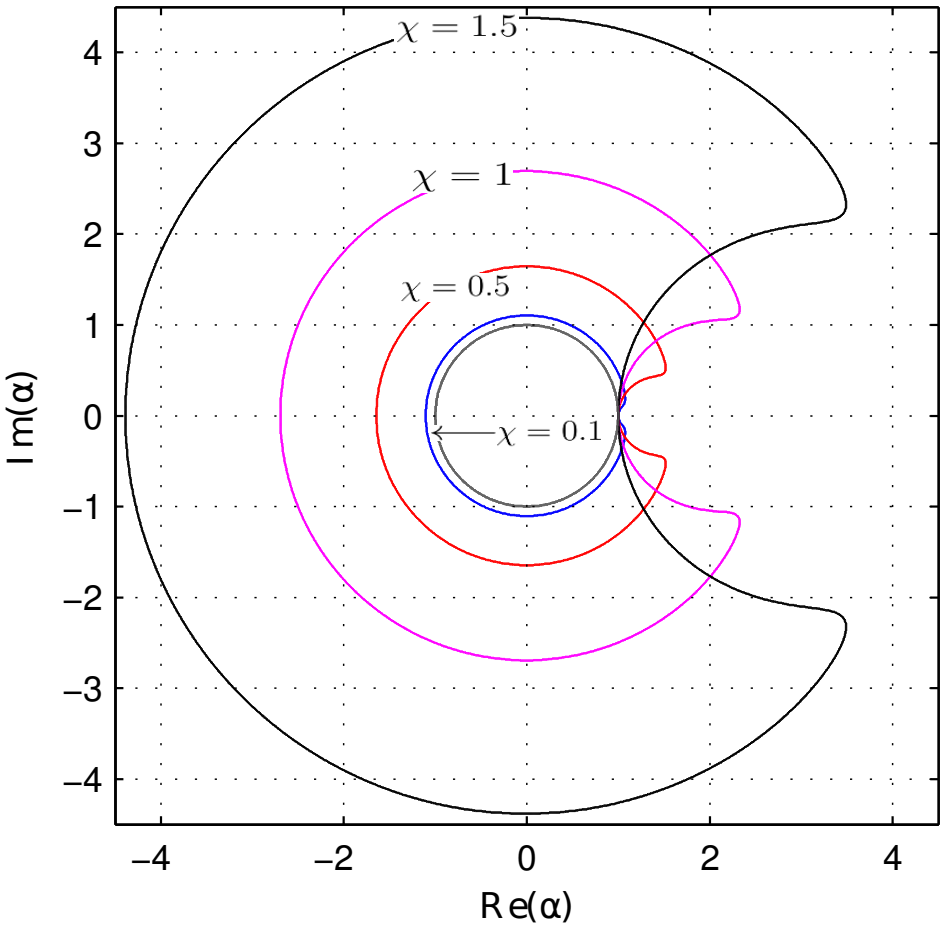


Figure 2.4: Superimposition of the stability regions shown on Fig. 2.3.

2.3 Numerical simulations

The encapsulated SFD method (2.18) has been implemented into the Nektar++ spectral/*hp* element framework as a wrapper function. In order to find the steady solution of the incompressible Navier-Stokes equations, a solver which implements the *velocity-correction* scheme (symbolised by the functions $\mathbf{\Phi}$ in (2.18)) is called at each time step. In this section we present the numerical steady-state of the two-dimensional incompressible flow past a circular cylinder above its critical Reynolds number Re_c (*i.e.* when the Reynolds number is high enough such that the viscous forces within the flow are not dominant). For this flow, the Reynolds number is

defined as $Re = U_\infty L/\nu$, where U_∞ is the inflow (or far field) velocity, L is the cylinder diameter and ν is the kinematic viscosity.

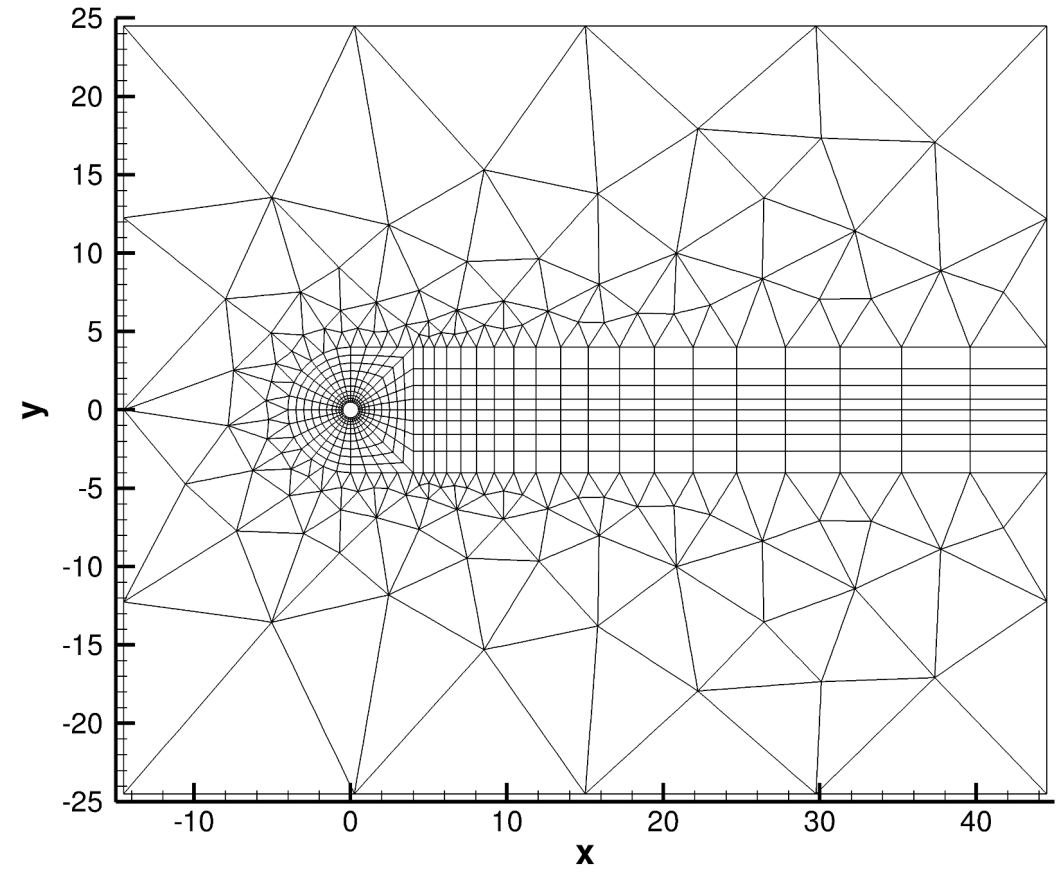
The numerical simulations presented in this section have been computed with the Nektar++ code using the second-order velocity-correction scheme to discretise the incompressible Navier-Stokes equations in time and the spectral/hp element method using a continuous Galerkin method to discretise the problem in space.

2.3.1 Unstable steady-state

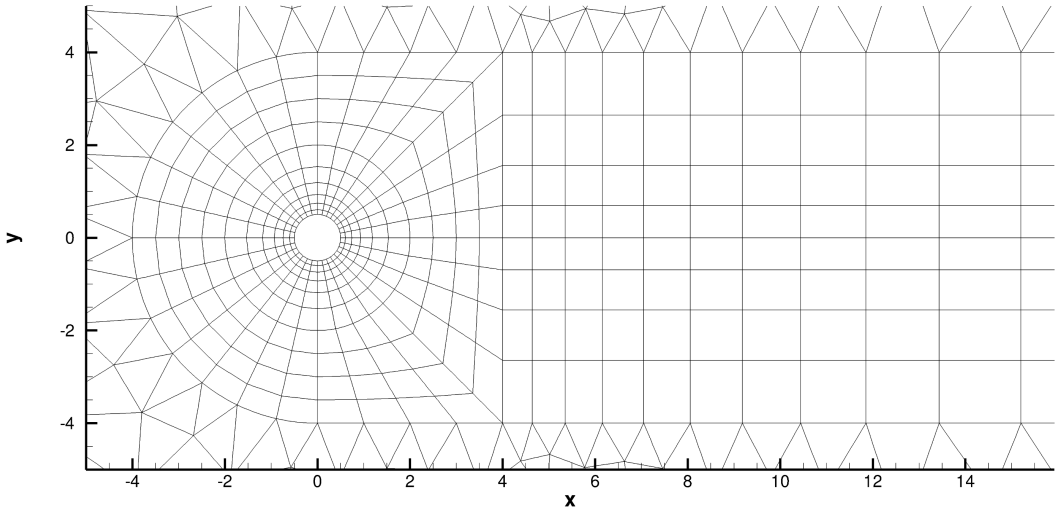
At the critical Reynolds number $Re_c \simeq 47$, the incompressible flow past a cylinder follows the path of a Hopf bifurcation [68, 69]. Below Re_c the flow is steady and stable and above Re_c, the flow becomes periodic, which indicates the presence of a limit cycle. This periodic orbit is stable because when a perturbation is introduced, the flow eventually returns to its periodic configuration. This behaviour remains up to the end of the subcritical laminar regime (*i.e.* $Re \simeq 2 \times 10^5$ [70]). A steady-state solution co-exists but is unstable. Hence the role of the SFD method is to control the evolution of the unstable frequencies and drive the solution towards its steady-state. In this section we are interested in the case of a Reynolds number $Re = 100$.

The computational domain used for this test case is presented on Fig. 2.5. Its dimensions are $-15 \leq x \leq 45$ and $-25 \leq y \leq 25$ and it is composed of 746 elements. The cylinder is centred at the origin and its diameter is one unit length. Near the cylinder boundary and in the wake behind the cylinder, the mesh is composed of structured quadrilateral elements. At the cylinder boundary, quadratic curved elements are used. The rest of the mesh is filled with triangular elements. On the cylinder surface, no-slip boundary conditions (*i.e.* homogeneous Dirichlet on the velocity and high-order Neumann on the pressure) are imposed. Dirichlet boundary conditions such as $(u,\ v) = (1,\ 0)$ are imposed on the left, top and bottom edges. This represents the far field characteristic velocity used to calculate the Reynolds number. Finally an outflow boundary condition is set on the right edge.

As the solution is expected to be smooth (*i.e.* the geometry studied does not have any "sharp" angle) a high polynomial order of 11 is chosen. Note that, the



(a) The entire computational domain.



(b) Close view on the sructured mesh close to the cylinder boundary.

Figure 2.5: Computational domain to simulated the incompressible flow past a cylinder at $Re = 100$.

computational domain shown in Fig. 2.5 is very coarse, but a high polynomial order leads to an accurate solution.

To highlight the fact that, unlike in Newton's methods, the SFD method does not need a good initial guess to converge, initial conditions such as $(u_0,\ v_0) = (0,\ 0)$ are chosen. The SFD parameters are chosen such that the control coefficient $\chi = 1$ and the filter width $\Delta = 2$. The problem is considered to be converged when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$. The steady-state has been obtained after the computation of about 1000 time units and the decay of $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ is exponential. The unstable steady-state solution obtained is shown on Fig. 2.6(b) and is identical to the one presented by Barkley [45]. Fig. 2.6(a) is simply a reminder of the behaviour of the incompressible cylinder flow at $Re = 100$ when the SFD is switched off.

A stability analysis, using the modified Arnoldi iteration method presented in Sec. 1.4.2, is performed with this steady-state as base flow. The growth rate σ and the frequency f of this flow configuration are

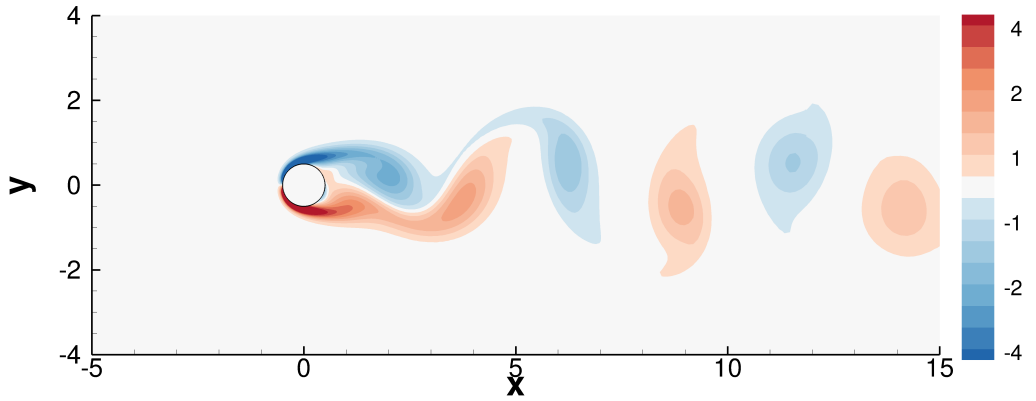
$$\sigma = 0.12708 \quad \text{and} \quad f = 2\pi \times 0.11788, \tag{2.27}$$

which correspond to the values presented by Barkley [45]. Note that, a velocity-correction scheme of second order was used to execute the modified Arnoldi iteration method. Jordi *et al.* [63] used a first order velocity-correction scheme to compute a similar stability analysis, which explains why the eigenvalue obtained in [63] is different from the one presented here.

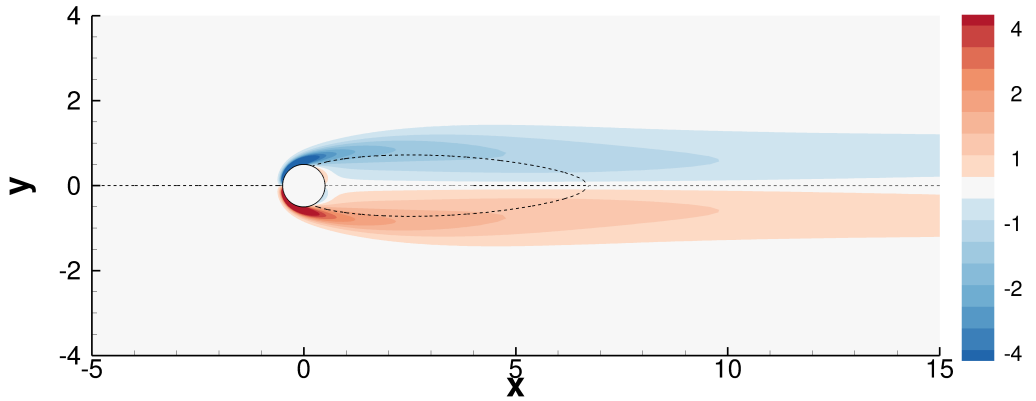
If the time length of each Arnoldi iteration (denoted T) is defined as being equal to one time unit, this growth rate and this frequency correspond to the two dominant (conjugate) eigenvalues

$$\lambda_{1,2} = e^{(\sigma \pm \mathbf{i} f)T} = 1.13551 e^{\pm 0.74066\ \mathbf{i}}. \tag{2.28}$$

As $|\lambda_{1,2}| > 1$, the flow is unstable and the corresponding instabilities exponentially grow through time. However the SFD method was able to control and suppress these instabilities and the solution converged towards a steady-state. The dominant eigenmode corresponding to the most unstable eigenvalue is presented in Fig. 2.7.



(a) Snapshot of the uncontrolled flow (unsteady and periodic). This behaviour is the well known von Kármán shedding.



(b) Unstable steady-state obtained by SFD. The dashed lines represent the separating streamlines.

Figure 2.6: Vorticity ($\omega = \partial_x v - \partial_y u$) of the incompressible flow past a two-dimensional cylinder at $Re = 100$. Note that only a small part of the whole computational domain is displayed.

This configuration corresponds to the results presented by Barkley [45] and also by Tu & Rowley [49].

Note that the control coefficient and the filter width used to obtain the unstable steady-state solution presented in this section have been selected arbitrarily. On a desktop (HP Z820 workstation equipped with 12 Intel Xeon processors), the execu-

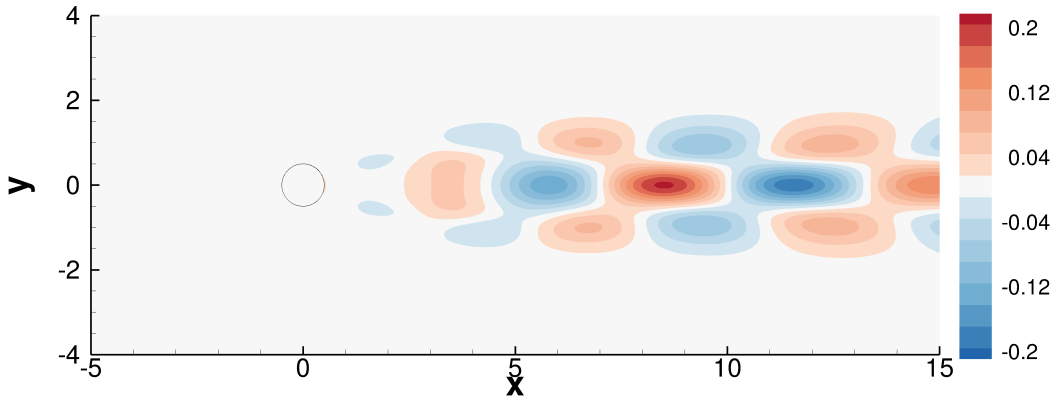


Figure 2.7: Vorticity of the dominant eigenmode of the incompressible flow past a two-dimensional cylinder at $Re = 100$.

tion of this test case takes three to twelve hours on one CPU with a direct solver for the Helmholtz equations of the velocity correction scheme. Then several different parameter pairs were tested and $\chi = 1$ and $\Delta = 2$ were found to be appropriate. With these control parameters, the SFD method is converging reasonably quickly. In the following section we propose an explanation as to why a given pair of parameters (χ, Δ) is able (or not) to force the system to evolve towards a steady-state solution.

2.3.2 Stability of the SFD method

In this section we propose a model to analyse the stability of the SFD method applied to the Navier-Stokes equations. The idea is to focus the analysis on how the SFD method influences the least stable eigenmode. In other words, we want to know if controlling the instabilities corresponding to the least stable eigenvalue enables the SFD method to converge towards a steady-state solution.

We denote the (conjugate) dominant unstable eigenvalues of a given flow problem as λ_D. We also introduce the control parameters χ_D and Δ_D such that the application of the SFD method with these parameters to the one-dimensional problem $u^{n+1} = \lambda_D u^n$ converges towards its steady-state solution. This one-dimensional

problem models the isolated evolution of the least stable eigenvalue of the flow system. We want to verify if χ_D and Δ_D are also capable of damping the instabilities of the flow whose dominant eigenvalue is λ_D and to force the system to evolve towards a steady flow.

The case studied is the incompressible flow past a circular cylinder at $Re = 100$. The dominant eigenvalues λ_D of this flow are known and given by (2.28). We follow the procedure described in Sec. 2.2.2 to obtain the stability regions of the SFD method applied to $u^{n+1} = \lambda_D u^n$ for $(\chi = 1,\ \Delta = 0.5)$, $(\chi = 1,\ \Delta = 1)$ and $(\chi = 1,\ \Delta = 2)$. The contours of these stability regions are shown on Fig. 2.8. The dominant eigenvalues λ_D are also shown on this figure in order to determine their location regarding a given stability region.

On Fig. 2.8 we observe that λ_D is situated inside the red and the green stability regions. It indicates that $(\chi = 1,\ \Delta = 1)$ and $(\chi = 1,\ \Delta = 2)$ ensure stabilisation of the unstable problem $u^{n+1} = \lambda_D u^n$, and that the SFD method converges towards a steady-state solution. However it is not the case for $(\chi = 1,\ \Delta = 0.5)$ because λ_D is outside the blue stability region.

We have evaluated how these control parameters perform when the SFD method is applied to the one-dimensional problem $u^{n+1} = \lambda_D u^n$. We want to verify if the same parameters act the same way on the flow problem. Three different numerical simulations were done. All are defined in the exact same way as described in Sec. 2.3.1, only the control coefficient and the filter width were modified. The convergence history for the parameters $(\chi = 1,\ \Delta = 0.5)$, $(\chi = 1,\ \Delta = 1)$ and $(\chi = 1,\ \Delta = 2)$ are shown on Fig. 2.9. We notice that when λ_D is situated inside the stability region of the one-dimensional case, the SFD method, with the same control parameters, applied to the flow problem controls the instabilities and converges towards a steady-state solution. Meanwhile if λ_D is outside of the stability region of the one-dimensional case, using the same parameters for the flow problem will not yield to a steady flow. In such a case, the SFD method is not able to completely suppress the unstable temporal frequencies. Fig. 2.9(b) shows that when the SFD method is not converging towards a steady-state, $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ does not

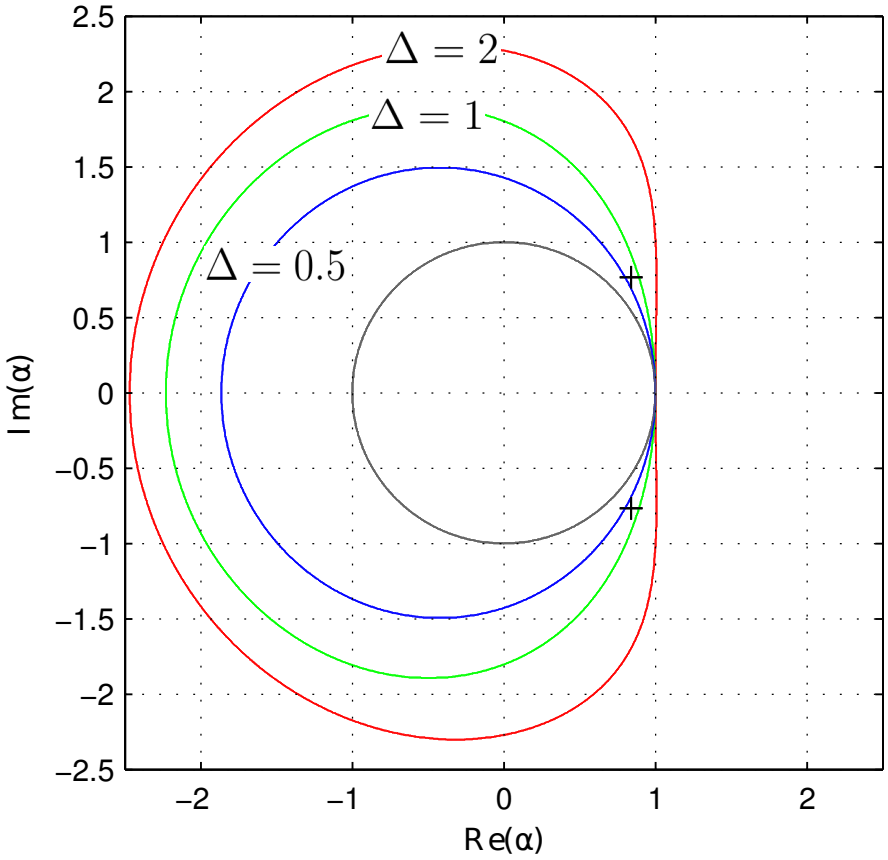
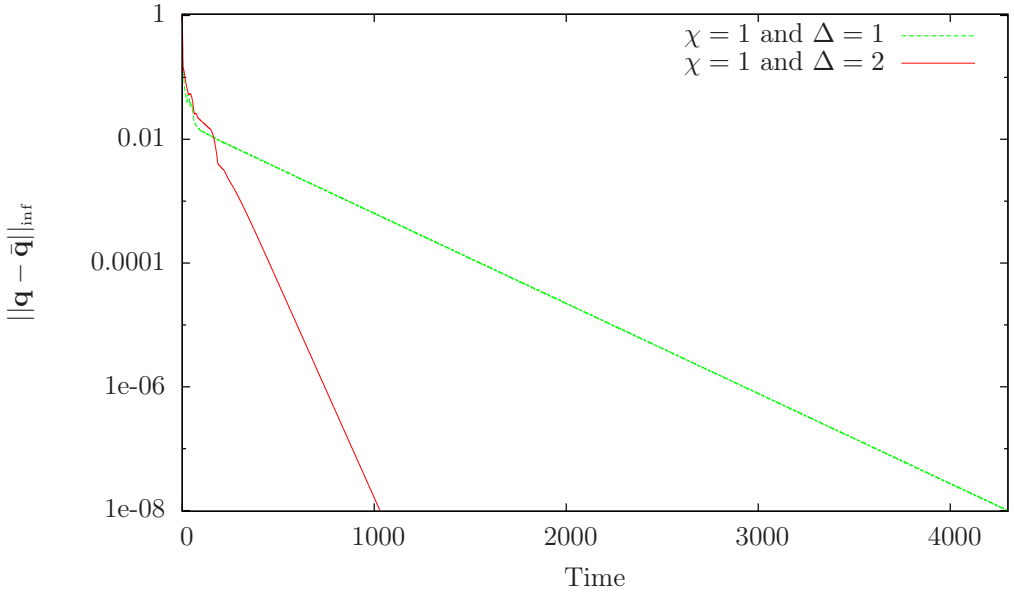


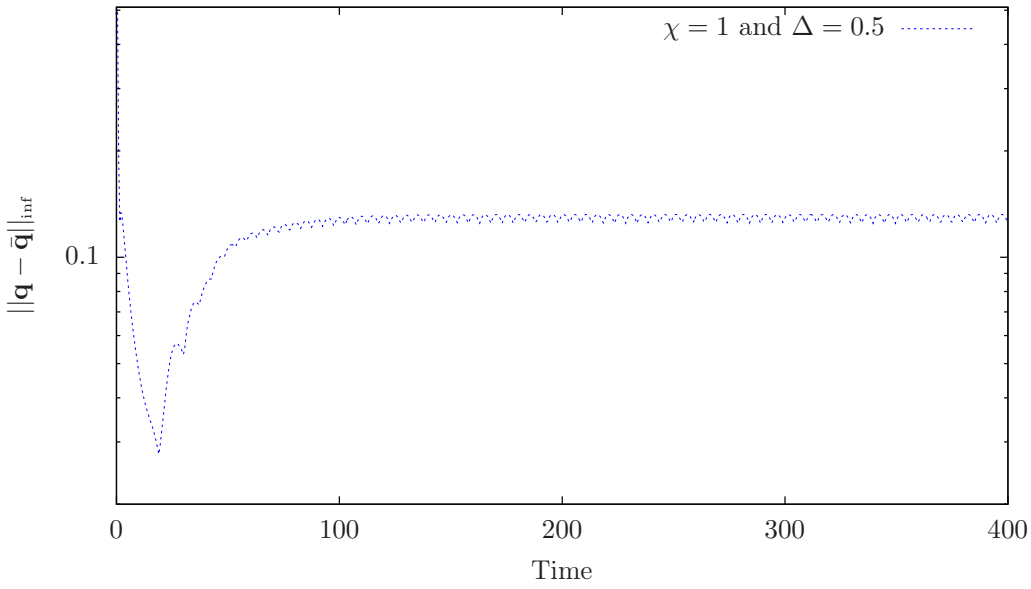
Figure 2.8: Contours of the stability regions of (2.22) for $\chi = 1$ and various Δ. The central grey circle represents the boundary of the unit disc. The two black crosses indicate the position of the dominant unstable eigenvalues λ_D (2.28) of the flow past a cylinder at $Re = 100$.

decrease but oscillates around a fixed value. Fig. 2.10 presents the outcome of the SFD method with ($\chi = 1,\ \Delta = 0.5$). The flow is partially controlled but the most dangerous instabilities cannot be completely damped by the SFD method with this choice of parameters.

Note that for parameters that enable convergence of the SFD method towards a steady flow, the computational time required to reach convergence strongly depends χ and Δ. Fig. 2.9(a) compares the number of iterations computed by the Navier-Stokes solver before obtaining the steady-state. With $\chi = 1$ and $\Delta = 1$ the SFD method needs about 4 times more iterations to converge than with $\chi = 1$ and $\Delta = 2$.



(a) Cases where the dominant unstable eigenvalues λ_D of the flow past a cylinder at $Re = 100$ are situated *inside* the stability region of the SFD method applied to $u^{n+1} = \lambda_D u^n$.



(b) Case where the dominant unstable eigenvalues λ_D of the flow past a cylinder at $Re = 100$ are situated *outside* the stability region of the SFD method applied to $u^{n+1} = \lambda_D u^n$.

Figure 2.9: Evolution of $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ in time for parameters which allow the encapsulated SFD method to converge towards the steady-state (a), and for parameters which do not (b).

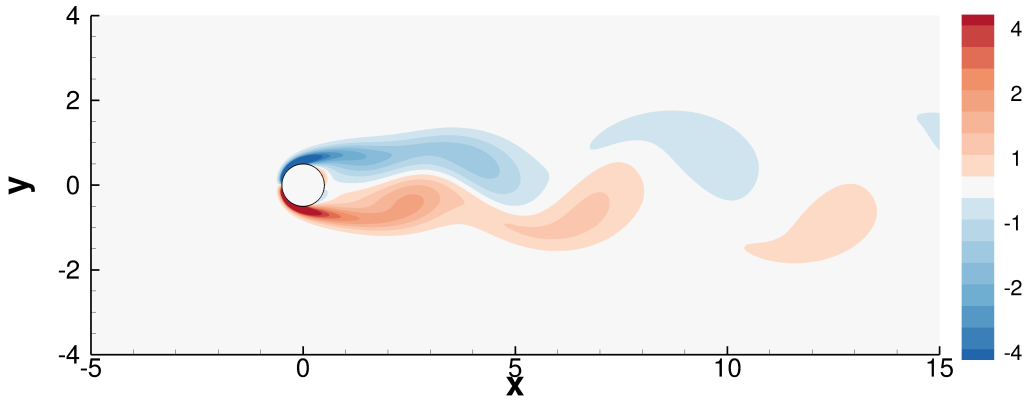


Figure 2.10: Snapshot obtained with SFD parameters ($\chi = 1,\ \Delta = 0.5$) which do not allow convergence towards a steady-state solution.

We recall that increasing the filter width does not always decrease the computation time of the method. Indeed when Δ becomes too large, the asymptotic behaviour presented in (2.26) appears.

In this section we proposed a procedure, based on the analysis of the SFD method applied to $u^{n+1} = \lambda_D u^n$, to predict if a given pair of control parameters ($\chi,\ \Delta$) can guarantee convergence of the SFD method applied to a flow whose least stable eigenvalue is λ_D. However this does not give any information on the convergence rate towards the steady-state solution. More importantly, this procedure is based on the assumption that the dominant eigenvalue is known *a priori*. Most of the time, especially for flows of industrial relevance, this is not the case. In the following chapter we introduce an *adaptive* SFD method to get around this issue.

2.4 Chapter summary

In this chapter we presented an alternative encapsulated formulation of the SFD method, which is used to obtain unstable steady-state solutions of the Navier-Stokes equations. This method enables to implement a steady-state state solver in a non-intrusive way by creating a wrapper function around an existing unsteady code. The

computational cost added by this algorithm is negligible. Hence the performance of the steady-state solver is equivalent to the one of the unsteady solver.

We also applied the encapsulated SFD method to a simple one-dimensional model. We observed that a relationship can be drawn between the ability of the SFD method to stabilise the unstable one-dimensional problem $u^n = \lambda_D u^{n+1}$ and the stabilisation of a flow whose dominant unstable eigenvalue is λ_D. This property was verified for the incompressible flow past a two-dimensional circular cylinder at $Re = 100$.

Finally, we have seen that the SFD method is not able to stabilise dynamical systems with real positive unstable eigenvalues. These problems are characterised by a pure exponential growth of the instabilities (*i.e.* with no oscillations). For example, wall confined jets have real positive unstable eigenvalues [71]. This feature suggests that the stability of the SFD method relies on the oscillatory growth of the instabilities of the problem studied.

CHAPTER 3

AN ADAPTIVE SELECTIVE FREQUENCY DAMPING METHOD

The convergence of the SFD method (in any formulation) is governed by two parameters, which are the control coefficient χ and the filter width Δ. For arbitrary choice of these parameters, the method may not be able to control the evolution of the instabilities within the flow. Hence the method does not always converge. Even when a steady-state solution can be found, convergence may be very slow. The selection of the parameters χ and Δ is central for users of the SFD method. We intend to address this issue in this chapter.

We present an adaptive procedure that couples the SFD method and a global stability analysis method. The idea is to approximate the dominant eigenvalue of the flow studied during the execution of the solver implementing the SFD method. This approximation is used to tune χ and Δ by using a simple one-dimensional optimisation model. The strength of this procedure is that it requires very little (or not any) knowledge of the flow behaviour before executing the code. This chapter is based around the paper entitled "An adaptive selective frequency damping method", published in *Physics of Fluids* in September 2015 [72].

In Sec. 3.1 we present how optimum parameters of the SFD method can be selected when the dominant eigenvalue of the system studied is known. In Sec. 3.2 we propose an adaptive procedure to automatically select parameters that ensure an optimum convergence of the SFD method. Finally in Sec. 3.3 we show that this procedure can be successfully applied to obtain unstable steady-state solutions of several classical test cases of computational fluid dynamics.

3.1 Evaluation of optimum parameters

We have seen in Sec. 2.3.2 that a link can be established between the convergence of the SFD method applied to the one-dimensional model $u^{n+1} = \lambda_D u^n$ and to a flow problem whose dominant eigenvalue is λ_D. We observed that if λ_D is inside the stability region of (2.22) for a given control coefficient and filter width, then these parameters also enable the SFD method to converge towards a steady flow. However this analysis does not provide any information about the convergence rate. In this section we propose a procedure to select parameters χ and Δ enabling optimum convergence of the SFD method.

We recall that the application of the SFD method applied to $u^{n+1} = \lambda_D u^n$ is written

$$\begin{aligned} \begin{pmatrix} u^{n+1} \\ \bar{u}^{n+1} \end{pmatrix} &= \begin{pmatrix} a\lambda_D & b\lambda_D \\ c & d \end{pmatrix} \begin{pmatrix} u^n \\ \bar{u}^n \end{pmatrix} \\ &= \mathcal{M} \begin{pmatrix} u^n \\ \bar{u}^n \end{pmatrix}, \end{aligned} \tag{3.1}$$

where a, b, c and d are defined in (2.19).

In Sec. 2.2.2 the stability regions presented allowed us to visualise the range of eigenvalues that can be controlled by a given pair (χ, Δ). On Fig. 2.1 and 2.3 the whole stability regions were coloured in grey. However through a different visualisation process it is possible to have a deeper insight on the stability behaviour. On Fig. 3.1 we present the stability region for $\chi = 1$ and $\Delta = 2$ including the contours of the dominant eigenvalue values.

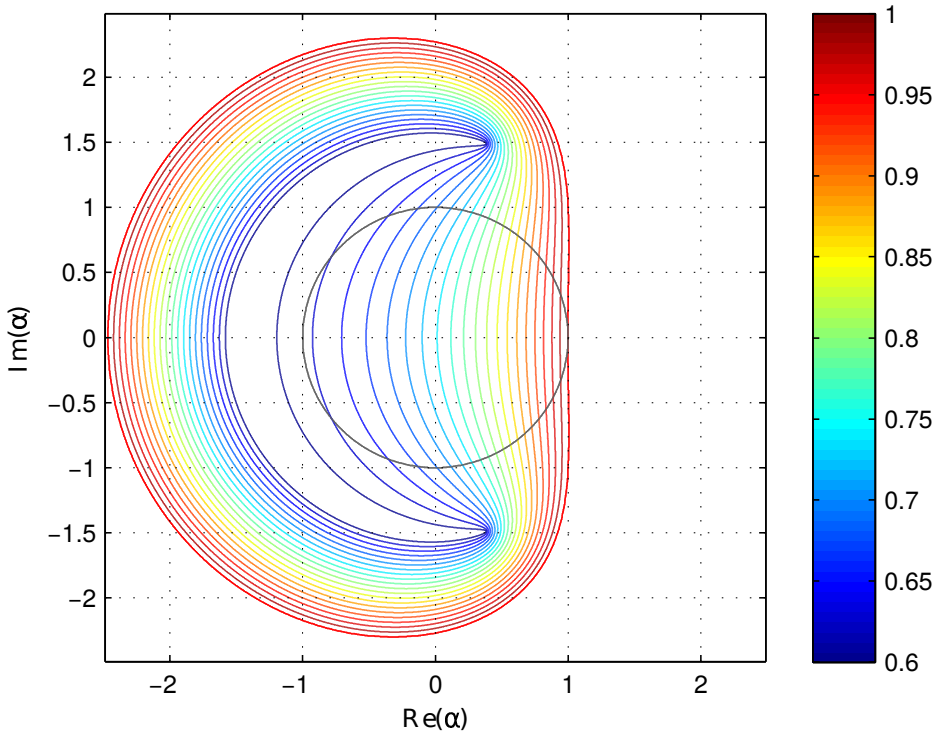


Figure 3.1: Stability region of (3.1) for $\chi = 1$ and $\Delta = 2$ coloured by values of the dominant eigenvalue. The lower the eigenvalue, the faster the convergence of (3.1). The central grey circle represents the limit of the unit disc.

On this figure we see that the dominant eigenvalue tends to one close to the border of the stability region. This means that system (3.1) converges very slowly if λ_D is close to the border (but inside the stability region). However the convergence of (3.1) is fast for a λ_D inside the middle crescent shape because this is where the dominant eigenvalue of $\mathcal{M}$ is small. The minimum dominant eigenvalues of $\mathcal{M}$ are conjugate and situated at the extremities of this crescent.

This analysis indicates the value of λ_D for which the application of the SFD method to $u^{n+1} = \lambda_D u^n$ converges the fastest for $\chi = 1$ and $\Delta = 2$. However we are interested in a slightly different problem. We assume that λ_D is known and fixed (*i.e.* it is the dominant eigenvalue of the flow studied) and we want to find the control coefficient and filter width guaranteeing the fastest convergence of (3.1)

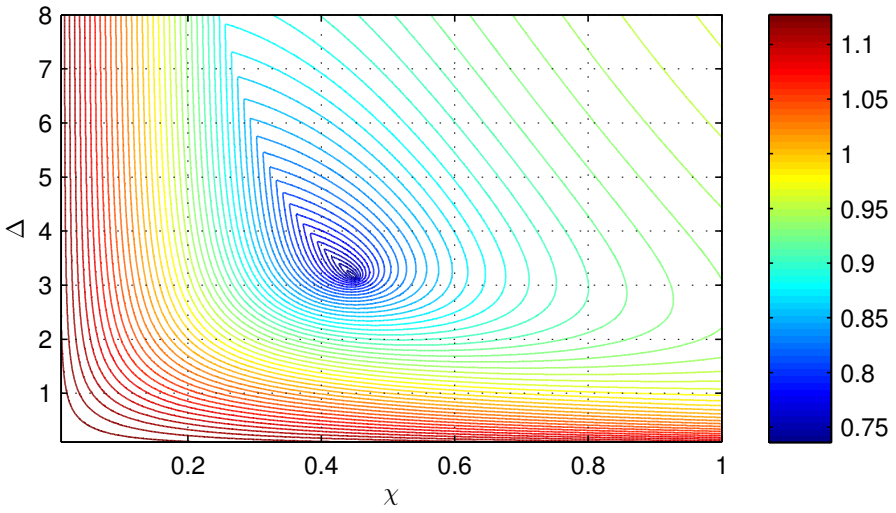


Figure 3.2: Contours of the dominant eigenvalue of $\mathcal{M}$ for $\lambda_D = 1.13551e^{\pm 0.74066\mathbf{i}}$ (*i.e.* $\sigma = 0.12708$ and $f = 2\pi \times 0.11788$). This eigenvalue is considered because it is the dominant one of the two-dimensional flow past a cylinder at $Re = 100$.

towards the steady-state $u = \bar{u} = 0$. Not the other way around. This problem converges the fastest when the dominant eigenvalue of $\mathcal{M}$ is minimum. Hence a two-dimensional optimisation problem has to be addressed [73].

We consider the two-dimensional space with χ on the horizontal axis and Δ on the vertical one (for $\chi \geqslant 0$ and $\Delta > 0$). For every point of this domain we evaluate the dominant eigenvalue of $\mathcal{M}$ in a similar way as for obtaining the stability regions. We obtained Fig. 3.2 with this procedure.

It appears on this picture that dominant eigenvalue of $\mathcal{M}$ presents one global minimum. Hence this minimum corresponds to the values of the control coefficient and filter width that enable the fastest convergence of (3.1) for a fixed λ_D. Note that graphs were computed for a very wide variety of parameters (*i.e.* $(\sigma, f, \chi, \Delta) \in]0, 5000]$) and this minimum, when it exists, has always been found to be unique. However when the frequency f of the eigenvalue chosen is to close to zero, there are no SFD parameters the enable convergence of (3.1). This comes from the fact that when there is no frequency to be damped, the SFD method is not an appropriate tool to find a steady-state solution.

As the matrix $\mathcal{M}$ is only 2×2 and its dominant eigenvalue has a unique minimum (when λ_D is fixed), a very simple line search method is used to find the parameters χ_{opt} and Δ_{opt} that guarantee the fastest convergence of (3.1). Such a method consists in reducing a two-dimensional optimisation problem into a series of one-dimensional problems. The idea is to evaluate the steepest descent direction and follow it until the dominant eigenvalue of $\mathcal{M}$ stops decreasing. The line search method used in this thesis is detailed in Algorithm 2. We define $MaxEig$ to be a routine that evaluates both eigenvalues of $\mathcal{M}$ and returns the magnitude of one with the largest magnitude. The constants $\delta\chi$ and $\delta\Delta$ are infinitesimal parameters to evaluate the gradient of $MaxEig$ and κ is the fixed step length of the line search method.

Note that using a varying step length κ would make this method more robust and would enable a faster convergence. However the simplicity of this optimisation problem is such that a fixed κ is acceptable because the code converges in a few seconds. As convergence is not guaranteed, a restart procedure has also been implemented. If the optimum parameters are not found in 10000 steps, we restart the algorithm with a greater initial value for the filter width. With this restart method, we were able to guarantee convergence of the line search algorithm.

We have presented a procedure to select the parameters that ensure the fastest convergence of the SFD method applied to $u^{n+1} = \lambda_D u^n$. When $\lambda_D = 1.13551e^{\pm 0.74066\mathbf{i}}$ (which is the dominant eigenvalue of the two-dimensional flow past a cylinder at $Re = 100$), we found that $\chi_{\text{opt}} = 0.451$ and $\Delta_{\text{opt}} = 3.144$. We also verified that these parameters guarantee the optimum convergence of the SFD method applied to the two-dimensional flow past a cylinder at $Re = 100$.

On Fig. 3.3 we show the convergence history for three different set of SFD parameters. We observe that the greatest convergence rate is reached when χ_{opt} and Δ_{opt} are used. For all the simulations executed on this test case, we were not able to obtain a greater convergence rate than the one of χ_{opt} and Δ_{opt}. Hence this behaviour suggests that the parameters that are optimum for the one-dimensional model are also optimum for the flow problem.

The issue with this feature is that to be able to adjust the parameters of the SFD

Algorithm 2 Line search method

Choose initial control coefficient χ and filter width Δ

Choose the parameters $\delta\chi$, $\delta\Delta$ and κ

while Optimum parameters are not found **do**

Set Descending := true

Compute $\lambda_0 := MaxEig(\mathcal{M}(\chi, \Delta))$

Set $\lambda_{\text{min}} := \lambda_0$

Compute $\lambda_\chi := MaxEig(\mathcal{M}(\chi + \delta\chi, \Delta))$

Compute $\lambda_\Delta := MaxEig(\mathcal{M}(\chi, \Delta + \delta\Delta))$

Evaluate descent direction $\boldsymbol{\beta} := \begin{cases} \beta_\chi := \lambda_0 - \lambda_\chi \\ \beta_\Delta := \lambda_0 - \lambda_\Delta \end{cases}$

while Descending **do**

Update $\chi := \chi + \kappa\beta_\chi$ and $\Delta := \Delta + \kappa\beta_\Delta$

Compute $\lambda_1 := MaxEig(\mathcal{M}(\chi, \Delta))$

if $\lambda_1 > \lambda_{\text{min}}$ **then**

Set Descending := false

else

Set $\lambda_{\text{min}} := \lambda_1$

end if

end while

if $|\lambda_1 - \lambda_0| < \varepsilon$ **then**

Optimum parameters are found

Return $\chi_{\text{opt}} := \chi - \kappa\beta_\chi$ and $\Delta_{\text{opt}} := \Delta - \kappa\beta_\Delta$

end if

end while

method, the dominant eigenvalue λ_D of the system is required. But this eigenvalue is obtained only after computing stability analysis; and stability analysis requires to linearise the governing equation around a steady-state solution. And a steady-state is what we intend to reach with the SFD method. In other words, to ensure

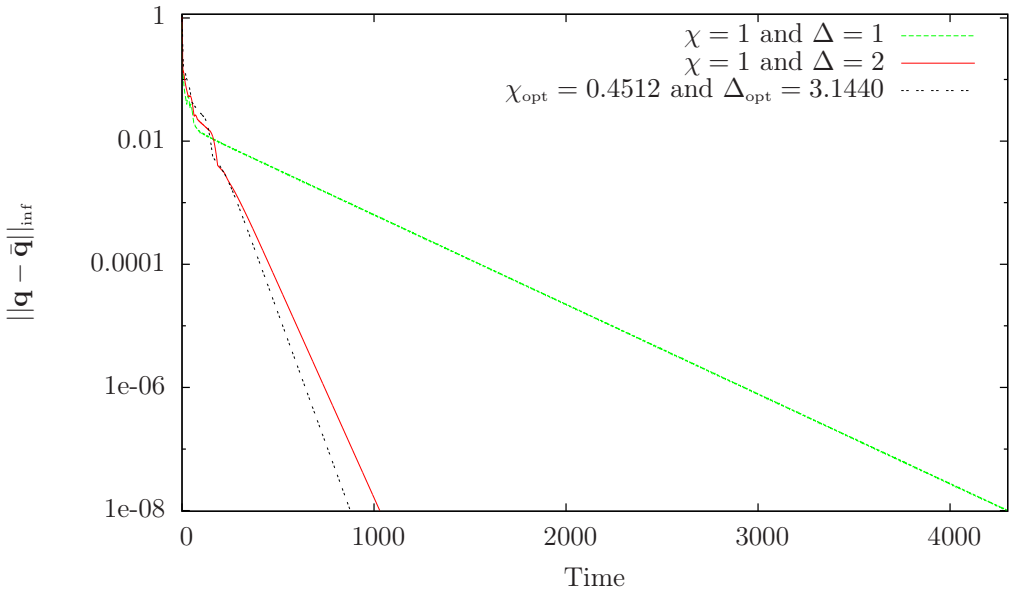


Figure 3.3: Evolution of $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ in time for parameters which allow the encapsulated SFD method to converge towards the steady-state solution of the incompressible flow past a cylinder at $Re = 100$.

convergence of the SFD method towards a steady-state solution, we need to know this steady-state solution. Fig. 3.4 illustrates this implicit problem. The closed loop shows that analysing the stability of the model (3.1) is useful only if the dominant eigenvalue λ_D is known *a priori*.

In the following section we present an adaptive algorithm to get around this issue.

3.2 Adaptive algorithm

In this section, we propose a procedure to address the issue of selecting appropriate parameters for the SFD method. Our algorithm links together the encapsulated SFD method presented in Chapter 2, the stability analysis method summarised in Sec. 1.4 and the one-dimensional optimisation model introduced in Sec. 3.1. The idea is to use a "partially converged" steady base flow for computing stability anal-

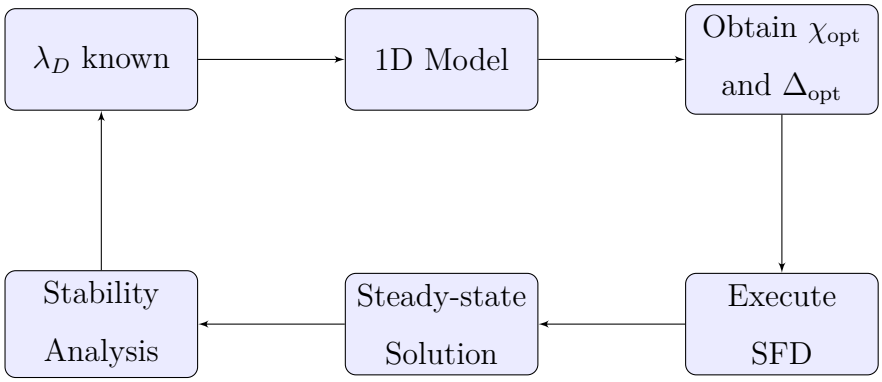


Figure 3.4: Illustration of the implicit problem of the unadapted SFD method. λ_D is the dominant eigenvalue of the flow; χ_{opt} and Δ_{opt} are the parameters that ensure an optimum convergence of (3.1).

ysis and obtain an approximation of the dominant eigenvalue (denoted $\tilde{\lambda}$). Then we use the procedure described in Sec. 3.1 to obtain the control coefficient $\tilde{\chi}$ and the filter width $\tilde{\Delta}$ that ensure the fastest convergence of the SFD method applied to $u^{n+1} = \tilde{\lambda} u^n$. These parameters are an approximation of the optimum parameters χ_{opt} and Δ_{opt}. The SFD method is then executed another time using the parameters $\tilde{\chi}$ and $\tilde{\Delta}$, so the instabilities can be damped more efficiently.

The main aspect of the procedure proposed here is the selection of the "partially converged" steady-state. To define this "partially converged" steady-state, we take a snapshot of the flow after executing the SFD method for T time units. The user has to define T considering that it has to be large enough to allow the flow to evolve but not too large to avoid wasting computational time executing the SFD method with badly suited parameters.

After defining the "partially converged" steady-state, the stability analysis method has to be computed. Note that here, the base flow used is not exactly a steady-state solution of the governing equations. The role of the SFD method is to reduce the temporal frequencies within the flow. Hence as the problem is not converged yet, the base flow selected is only an approximation of the steady-state solution. However our experiments suggested that computing stability analysis on an approx-

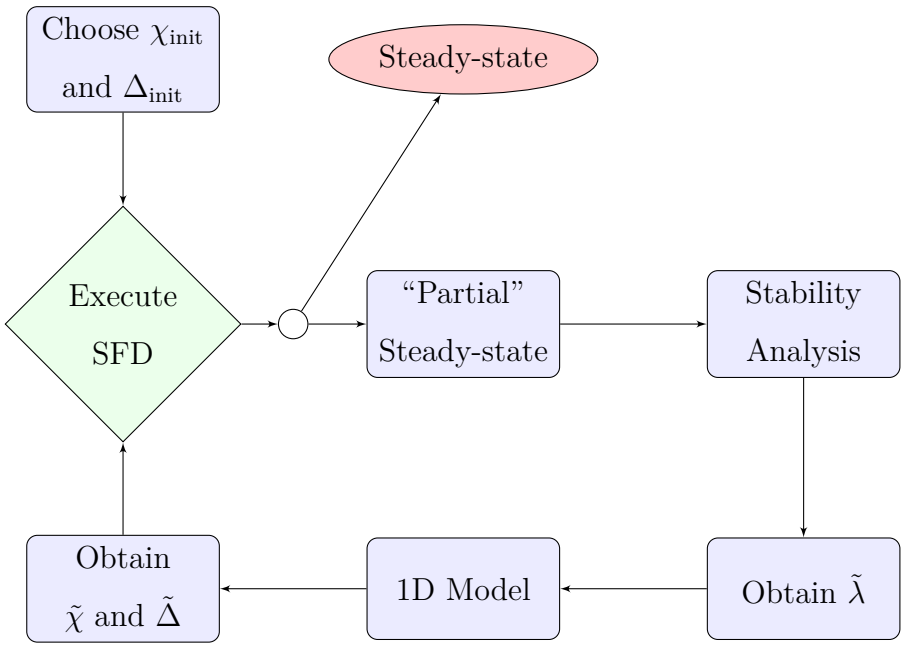


Figure 3.5: Illustration of the adaptive SFD method. χ_{init} and Δ_{init} are the initial parameters of the SFD method; $\tilde{\lambda}$ is an approximation of the dominant eigenvalue of the flow; $\tilde{\chi}$ and $\tilde{\Delta}$ are the parameters that ensure an optimum convergence of the SFD method applied to $u^{n+1} = \tilde{\lambda} u^n$. Note that the circle (right of "Execute SFD") represents a group of criteria to determine if the stability analysis method has to be computed or if the SFD parameters are fixed until convergence is reached.

imated steady-state gives a good approximation of the dominant eigenvalue of the flow studied. The user has to define a tolerance to determine when the modified Arnoldi iteration method is converged. As we only seek an approximation of the dominant eigenvalue, this tolerance does not need to be very small.

At this stage we have $\tilde{\lambda}$ which is an approximation of the dominant eigenvalue of the flow studied and we want to know $\tilde{\chi}$ and $\tilde{\Delta}$ which are the parameters ensuring the fastest convergence of the SFD method applied to $u^{n+1} = \tilde{\lambda} u^n$. The one-dimensional model presented in Sec. 3.1 is used to find $\tilde{\chi}$ and $\tilde{\Delta}$. After finding an approximation of the optimum parameters, the control coefficient and filter width are updated and the SFD method is executed again for T time units.

This adaptive process is iterated until the norm $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ becomes smaller than

a desired tolerance. When this becomes true, we consider that the approximation of the steady base flow is good enough, hence the control coefficient and filter width are fixed until convergence. If this tolerance is too large, then the flow will not have enough time to evolve, and the "partially converged" flow will not be a good approximation of the steady-state. If this tolerance is too small, stability analysis will be computed a lot of times using base flows close to each other which will waste computational resources. Our experiments suggested that a tolerance of 10^{-2} seems to be a good compromise.

This adaptive method is illustrated on Fig. 3.5. Note that when stability analysis has to be computed more than once (*i.e.* the loop of Fig. 3.5 is executed several times), the initial condition of the modified Arnoldi iteration method is the final solution of the previous computation. This allows us to speed up convergence of the stability analysis solver.

As we only seek an approximation of the dominant eigenvalue, the flow stability analysis method does not need to be very well converged. We fixed the tolerance of the modified Arnoldi iteration method to be 10^{-3}. A more accurate evaluation of the eigenvalue of an "approximated" steady-state solution would result in a waste of computational resources.

Also, we recall that the SFD method is based on the damping of unstable temporal frequencies. Then it can only control unstable problems with oscillatory growth of the instabilities (*i.e.* the imaginary part of the dominant eigenvalue is non-zero). Otherwise there is no frequency to be damped by the method. Hence all the $\tilde{\lambda}$ successively computed for this algorithm must have a non-zero imaginary part.

All users of the SFD method (in any formulation) must define an initial control coefficient and filter width. To execute the adaptive SFD method, the time T between two consecutive execution of the stability analysis method is an additional parameter that has to be defined.

3.3 Numerical simulations

In this section we present the application of the adaptive algorithm detailed in Sec. 3.2 to some classical examples of computational fluid dynamics. The aim is to show that the method is able to reach unstable steady-state solutions even if very little (or not any) information about the flow properties is available.

The algorithm has been implemented into the Nektar++ spectral/*hp* element framework. It couples two methods that act as wrappers around a flow solver that are the encapsulated SFD method [63] and a modified Arnoldi iteration method. The incompressible Navier-Stokes solver used throughout this section implements an unstabilised continuous Galerkin method to discretise the problem in space and a second order velocity-correction scheme for time-integration.

In our implementation, the user has to define the initial control coefficient χ_{init}, the initial filter width Δ_{init} and the time T for which the SFD method is executed before computing stability analysis. We define the length of one Arnoldi iteration to be equal to one time unit. Hence if the stability analysis solver converges in n iterations, the computational time will be approximately the same as executing the Navier-Stokes solver for n time units.

The adaptive algorithm is executed until $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-2}$. Once this mark is reached, we consider that the SFD parameters are a good approximation of the optimum ones. Then the control coefficient and the filter width are fixed until convergence (*i.e.* when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$).

In Sec. 3.3.1 the behaviour of the adaptive SFD method is detailed for the incompressible flow past a cylinder at $Re = 100$. Then in Sec. 3.3.2 we show that this case can easily be extended to higher Reynolds numbers and present the case $Re = 300$. In Sec. 3.3.3, the incompressible flow past an ellipse at $Re = 150$ is studied in order to show that the adaptive SFD method works well on geometries that do not contain axial symmetries. Finally, in Sec. 3.3.4 we consider the incompressible flow past a rotating cylinder at $Re = 100$ with a rotation rate $\alpha = 5$ and show that the steady-state solution of the unstable mode II can be reached without requiring the

use of continuation methods.

3.3.1 Incompressible flow past a cylinder at $Re = 100$

The first test case presented is the two-dimensional incompressible flow past a circular cylinder at $Re = 100$ (where the diameter of the cylinder is the characteristic length). At this Reynolds number, the flow is unstable and, if no control method is used, vortex shedding occurs. This flow was studied in Sec. 2.3.1 and we showed the SFD method is able to suppress the unsteady oscillations in the cylinder wake and force the system to evolve towards an unstable steady-state solution. In this section we are not interested in finding a steady-state *per se*, we want to evaluate the behaviour the adaptive algorithm presented in Sec. 3.2. We show that the adaptive SFD method is able to automatically select parameters that ensure convergence at an optimum rate. We use two different pairs of initial parameters: one for which the unadapted SFD method is converging slowly and the other for which the unadapted SFD method is not converging.

Both cases presented in this section use the same computational domain. It is composed of 746 elements and its dimensions are $-15 \leq x \leq 45$ and $-25 \leq y \leq 25$. The mesh is made of structured curved quadrilaterals close to the cylinder boundary and in the wake and of triangles elsewhere. No-slip boundary conditions are imposed at the cylinder surface and Dirichlet boundary conditions $(u,\ v) = (1,\ 0)$ are set at the left, top and bottom edges. An outflow boundary condition is defined at the right edge of the domain. The initial conditions are such that $(u_0,\ v_0) = (0,\ 0)$. As the solution is expected to be smooth, a polynomial order of 7 is used and the time-step is $\Delta t = 0.01$.

First, we execute the adaptive SFD method with initial parameters that ensure a slow convergence of the unadapted SFD method. These parameters are a control coefficient $\chi_{\text{init}} = 1$ and a filter width $\Delta_{\text{init}} = 1$. As we showed in Sec. 2.3.2, the unadapted method converges very slowly with these values (in about 4100 time units, see Fig 2.9(a)). Then here, we aim to improve the convergence rate of the SFD method by updating the control coefficient and the filter width during the

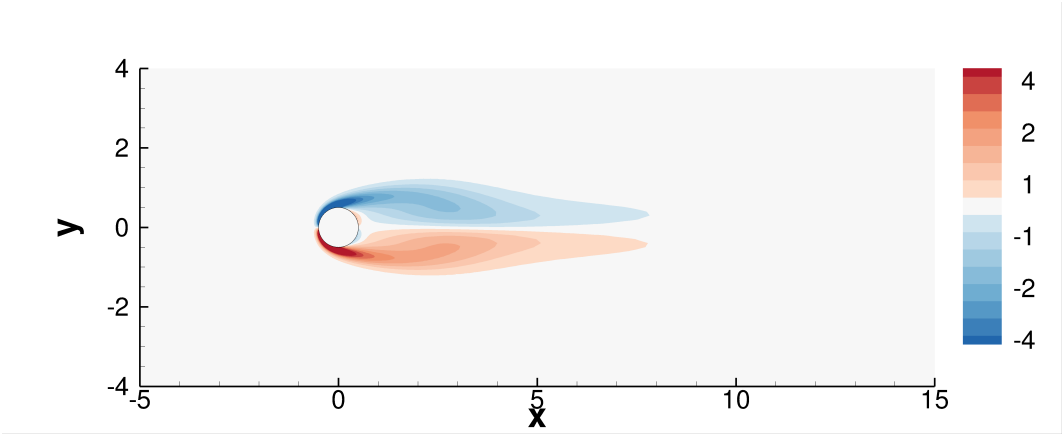
Time	Arnoldi method	Dominant eigenvalue	SFD parameters	$\|\|\mathbf{q}-\bar{\mathbf{q}}\|\|_{\text{inf}}$
$t=25$	Conv. in 59 steps	$\tilde{\sigma}_1=0.135$; $\tilde{f}_1=0.908$	$\tilde{\chi}_1=0.548$; $\tilde{\Delta}_1=2.482$	0.0399
$t=109$	Conv. in 39 steps	$\tilde{\sigma}_2=0.143$; $\tilde{f}_2=0.823$	$\tilde{\chi}_2=0.506$; $\tilde{\Delta}_2=2.821$	0.0355
$t=173$	Conv. in 38 steps	$\tilde{\sigma}_3=0.136$; $\tilde{f}_3=0.785$	$\tilde{\chi}_3=0.481$; $\tilde{\Delta}_3=2.967$	0.0180
$t=236$	Conv. in 34 steps	$\tilde{\sigma}_4=0.133$; $\tilde{f}_4=0.766$	$\tilde{\chi}_4=0.468$; $\tilde{\Delta}_4=3.042$	0.0158
$t=295$	Conv. in 33 steps	$\tilde{\sigma}_5=0.130$; $\tilde{f}_5=0.755$	$\tilde{\chi}_5=0.461$; $\tilde{\Delta}_5=3.084$	0.0122
$t=335$	Conv. in 22 steps	$\tilde{\sigma}_6=0.129$; $\tilde{f}_6=0.753$	$\tilde{\chi}_6=0.459$; $\tilde{\Delta}_6=3.193$	0.0099

Table 3.1: Successive execution of the stability analysis method for the adaptive SFD method when $T=25$ and $\chi_{\text{init}}=1$ and $\Delta_{\text{init}}=1$. The time at which the modified Arnoldi iteration method is called and the number of steps required to converge (*i.e.* when the residual becomes smaller than 10^{-3}) are reported in the first and second columns. The third column lists the successive approximations of the dominant eigenvalue, the corresponding optimum parameters are reported in the fourth column and the norm $||\mathbf{q}-\bar{\mathbf{q}}||_{\text{inf}}$ at each execution of the modified Arnoldi iteration method in the fifth column.

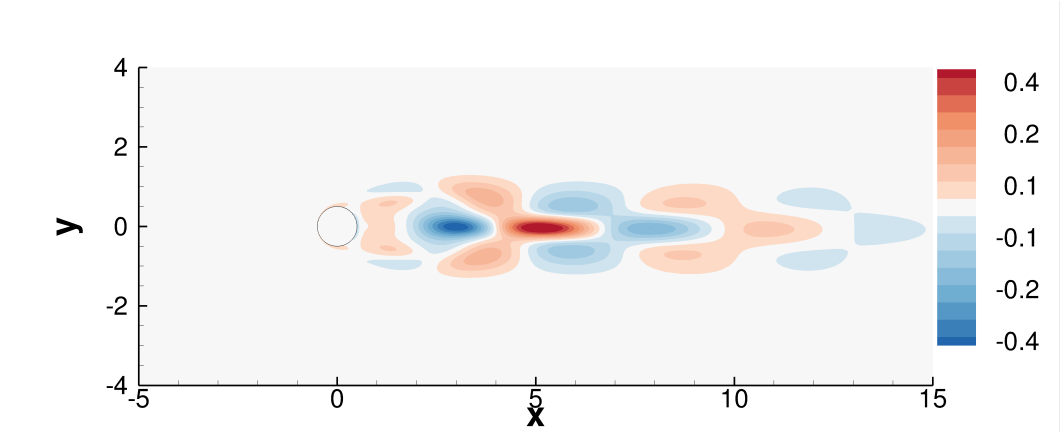
solver execution.

We choose $T=25$ to execute our adaptive algorithm, which means that the stability analysis method is computed every 25 time units until $||\mathbf{q}-\bar{\mathbf{q}}||_{\text{inf}}$ becomes smaller than 10^{-2}. The steady-state solution is reached after a total of 1093.5 time units. The stability analysis solver is executed 6 times for a total of 225 Arnoldi steps. We recall that for the modified Arnoldi iteration method, the computational cost of one step is approximately the same as the execution of the non-linear solver. Hence about 20% of the computational resources necessary to obtain a steady-state with the adaptive SFD method are spent evaluating the dominant eigenvalue of several "partially converged" base flows. This is eventually a good investment because the total number of steps required to converge with the adaptive method is nearly four times smaller than when the unadapted SFD method is executed.

To illustrate the algorithm behaviour in more details, the results of the successive execution of the stability analysis method are reported in Table 3.1. Note that



(a) Snapshot of the SFD method at $t = 25$.

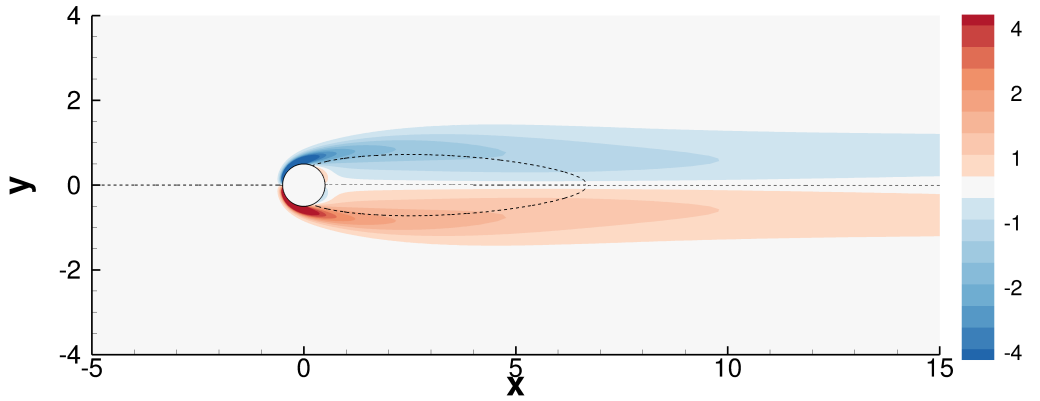


(b) Dominant eigenmode returned by the Arnoldi method using (a) as base flow.

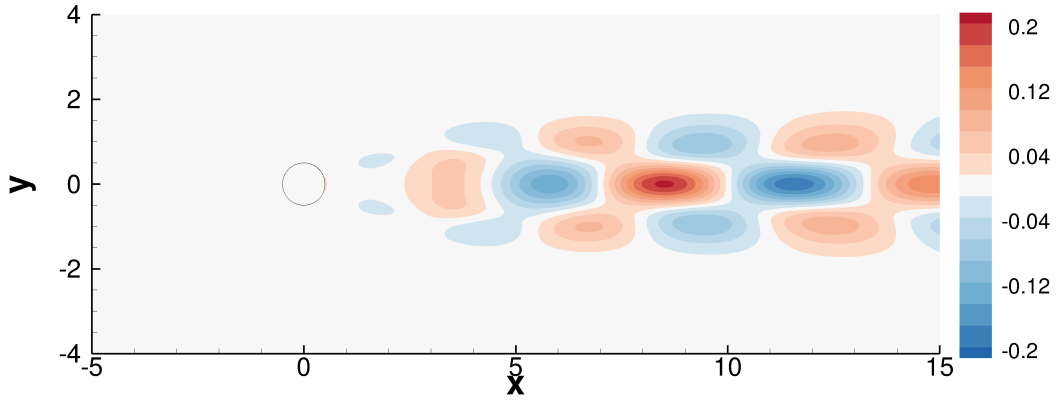
Figure 3.6: Vorticity of the first "partially converged" steady-state. Obtained for the parameters $\chi_{\text{init}} = 1$, $\Delta_{\text{init}} = 1$ at $t = 25$.

the time t reports the total of time units computed by the algorithm by adding the execution of the non-linear solver and the stability analysis solver. The first "partially converged" base flow (obtained at $t = 25$) is shown on Fig. 3.6. This flow field is nearly symmetric but the wake behind the cylinder is shorter than the one of the converged steady-state (see Fig. 3.7(a)). The first line of Table 3.1 reports the features to this flow.

In Table 3.1, we observe that after the third iteration, the value of the approximated eigenvalue and the corresponding optimum SFD parameters evolve very little.



(a) Unstable steady-state obtained by SFD. The dashed lines represent the separating streamlines.



(b) Corresponding dominant eigenmode.

Figure 3.7: Vorticity of the steady-state of the incompressible flow past a two-dimensional cylinder at $Re = 100$.

This suggests that, for this problem, a coarse approximation of the steady-state solution provides a quite good evaluation of the dominant eigenvalue of the flow. The steady-state obtained (when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$) is shown on Fig. 3.7. This flow is identical to the one presented on Sec. 2.3.1 which confirms that both the adaptive and unadapted SFD method converge towards the same steady-state solution.

We use the data presented in Sec. 2.3 and Sec. 3.1 to evaluate the advantages of the adaptive procedure. We recall the growth rate and the frequency of the

Time	Arnoldi method	Dominant eigenvalue	SFD parameters	$\|\|\mathbf{q}-\bar{\mathbf{q}}\|\|_{\text{inf}}$
$t=50$	Conv. in 59 steps	$\tilde{\sigma}_1=0.142$; $\tilde{f}_1=0.813$	$\tilde{\chi}_1=0.499$; $\tilde{\Delta}_1=2.859$	0.0225
$t=159$	Conv. in 41 steps	$\tilde{\sigma}_2=0.132$; $\tilde{f}_2=0.763$	$\tilde{\chi}_2=0.466$; $\tilde{\Delta}_2=3.054$	0.0158
$t=228$	Conv. in 34 steps	$\tilde{\sigma}_3=0.130$; $\tilde{f}_3=0.753$	$\tilde{\chi}_3=0.459$; $\tilde{\Delta}_3=3.095$	0.0099

Table 3.2: Successive execution of the stability analysis method for the adaptive SFD method when $T=50$ and $\chi_{\text{init}}=1$ and $\Delta_{\text{init}}=1$. The time at which the modified Arnoldi iteration method is called and the number of steps required to converge (*i.e.* when the residual becomes smaller than 10^{-3}) are reported in the first and second columns. The third column lists the successive approximations of the dominant eigenvalue, the corresponding optimum parameters are reported in the fourth column and the norm $||\mathbf{q}-\bar{\mathbf{q}}||_{\text{inf}}$ at each execution of the modified Arnoldi iteration method in the fifth column.

converged steady-state solution are $\sigma=0.127$ and $f=0.741$ respectively. The optimisation Algorithm 2 returned that the corresponding optimum parameters are $\chi_{\text{opt}}=0.451$ and $\Delta_{\text{opt}}=3.144$. When the unadapted SFD method is executed using these optimum parameters, it converges in 878 time units (see Fig. 3.3), which is only 20% faster than using the adaptive algorithm. Hence the adaptive algorithm used here can significantly decrease the number of time units necessary before reaching convergence of the SFD method without requiring any *a priori* knowledge of the flow.

We executed other simulations only changing the time T between two consecutive executions of the modified Arnoldi iteration method. For $T=50$, convergence is reached in 996 time units (including 134 Arnoldi steps) and details of the three successive executions of the stability analysis method are reported in Table 3.2. We also observe for this case that the approximations of the dominant eigenvalue are rapidly accurate and that the final SFD parameters are very close to the optimum ones.

The convergence history of the cases where $T=25$ and $T=50$ are reported on Fig. 3.8 along with the cases of the unadapted SFD method for $\chi=1$ and $\Delta=1$

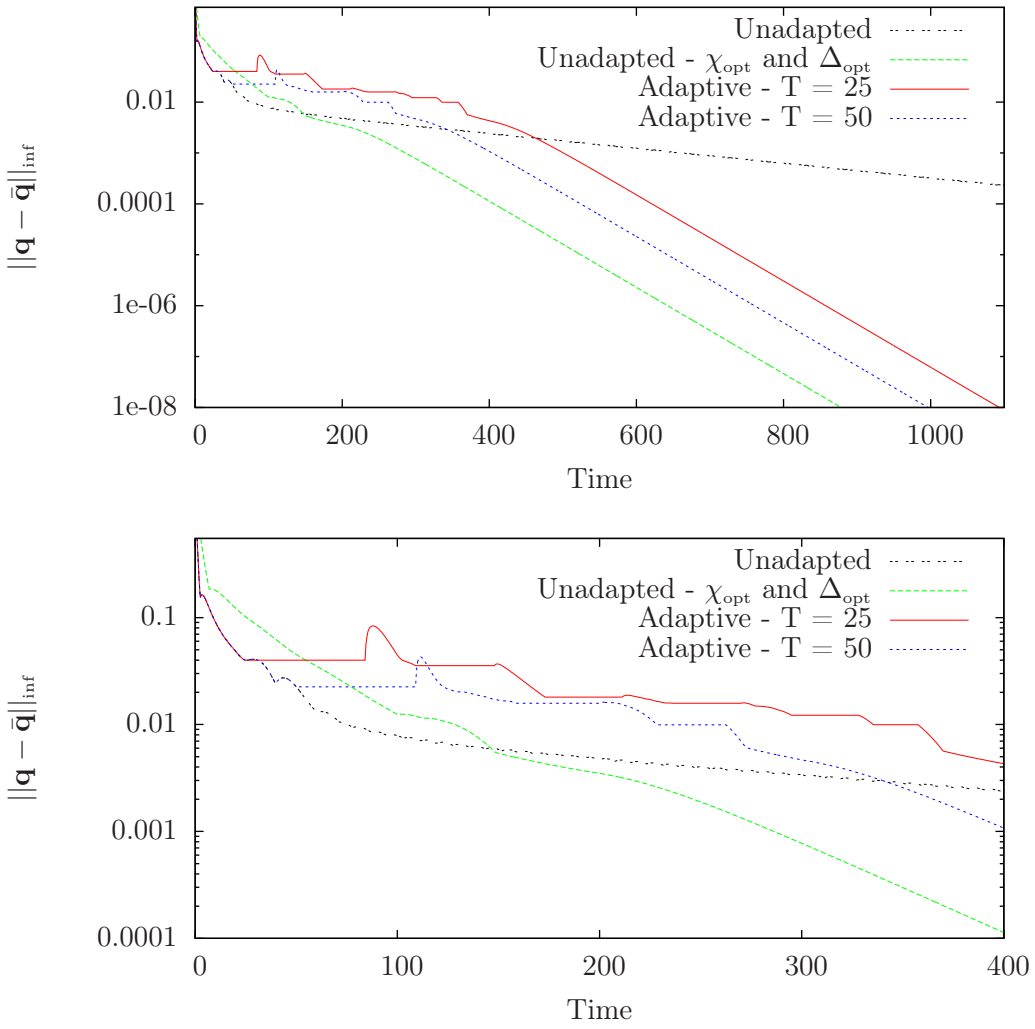


Figure 3.8: Convergence history of the adaptive SFD method for $T = 25$ and $T = 50$ with $\chi_{\text{init}} = 1$ and $\Delta_{\text{init}} = 1$. The cases of the unadapted SFD method for $\chi = 1$, $\Delta = 1$ (curve called "Unadapted") and for the optimum parameters $\chi_{\text{opt}} = 0.451$, $\Delta_{\text{opt}} = 3.144$ are also reported.

and for the optimum parameters $\chi_{\text{opt}} = 0.451$ and $\Delta_{\text{opt}} = 3.144$. The horizontal plateaux illustrate the time spent executing the stability analysis method. It is noticeable that in our adaptive method, after the control coefficient and filter width are fixed, all the cases have the same exponential convergence rate. This rate is

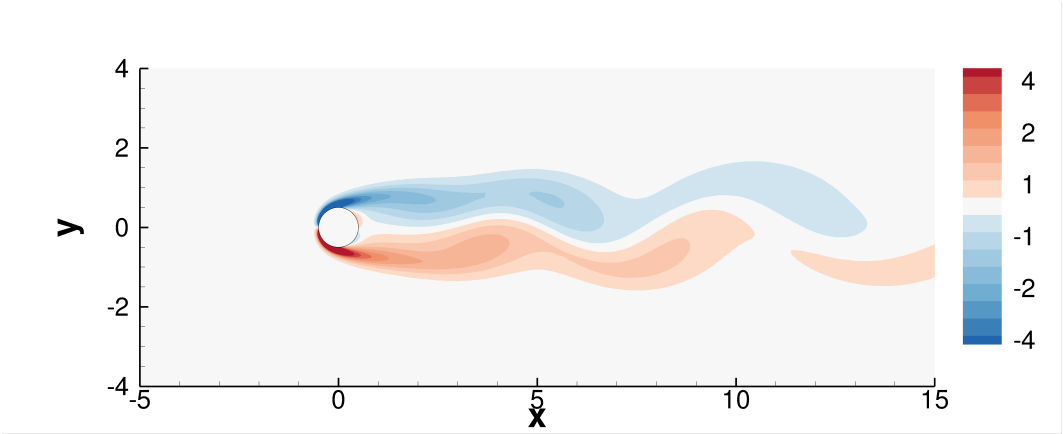
greater than the unadapted SFD method for $\chi = 1$ and $\Delta = 1$ and similar to the one of the unadapted SFD method executed with optimum parameters.

We now execute exactly the same set of simulations, only changing the initial parameters. We select $\chi_{\text{init}} = 1$ and $\Delta_{\text{init}} = 0.5$. In Sec. 2.3.2 we have shown that these initial parameters do not enable the SFD method to converge (see Sec. 2.3.2). The time evolution of $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ is such that it decreases for a certain time, then increases abruptly and eventually oscillates around a fixed value (see Fig. 2.9(b)). Hence in this case, we aim to use the adaptive algorithm presented in Sec. 3.2 to adjust the parameters such that a steady-state solution can be reached.

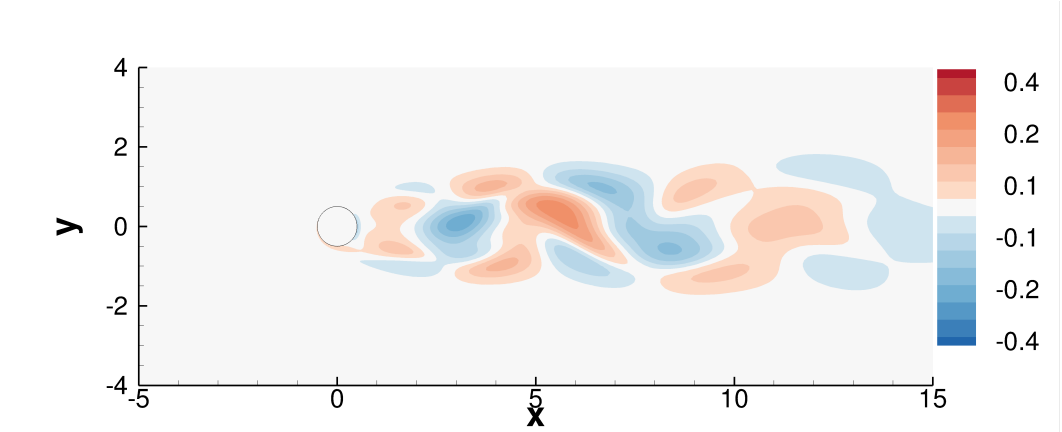
The first "partially converged" steady flow of the case $T = 50$ is shown on Fig. 3.9. Note that this flow configuration is not symmetric but the oscillation are reduced in comparison with the vortex shedding of the uncontrolled case (presented on Fig. 2.6(a)). The corresponding dominant eigenmode oscillates around the horizontal axis but its structure is comparable to the dominant eigenmode of the converged steady-state presented on Fig. 3.7(b). However the stability analysis executed using this "partially converged" base flow approximates well the dominant eigenvalue of the flow. Indeed the first approximated growth rate and frequency returned by the Arnoldi method are $\tilde{\sigma}_1 = 0.104$ and $\tilde{f}_1 = 0.821$ respectively. This is an other argument to support the idea that a coarse approximation of the steady flow can lead to a good approximation of the dominant eigenvalue.

The convergence history of the cases where $T = 25$, $T = 50$ and $T = 100$ are reported on Fig. 3.10 along with the cases of the unadapted SFD method for $\chi = 1$ and $\Delta = 0.5$ and for the optimum parameters $\chi_{\text{opt}} = 0.451$ and $\Delta_{\text{opt}} = 3.144$. The horizontal plateaux illustrate the time spent executing the stability analysis method. It is noticeable that in our adaptive method, after the control coefficient and filter width are fixed, all the cases have the same exponential convergence rate. This rate is similar to those of the unadapted SFD method executed with the optimum parameters. For conciseness, a detailed table is not presented for this case.

We conclude this section by observing that the adaptive SFD method has successfully (and automatically) selected parameters that ensure an optimum convergence



(a) Snapshot of the SFD method at $t = 50$.



(b) Dominant eigenmode returned by the Arnoldi method using (a) as base flow.

Figure 3.9: Vorticity of the first "partially converged" steady-state. Obtained for the parameters $\chi_{\text{init}} = 1$, $\Delta_{\text{init}} = 0.5$ at $t = 50$.

rate towards a steady-state solution even if little care is taken when choosing the initial control coefficient and filter width.

3.3.2 Incompressible flow past a cylinder at $Re = 300$

In this section we show that for the incompressible flow past a cylinder, the extension to higher Reynolds number test cases is straightforward. We execute the adaptive algorithm presented in Sec. 3.2 for $Re = 300$.

The computational domain is composed of 1330 elements and its dimensions

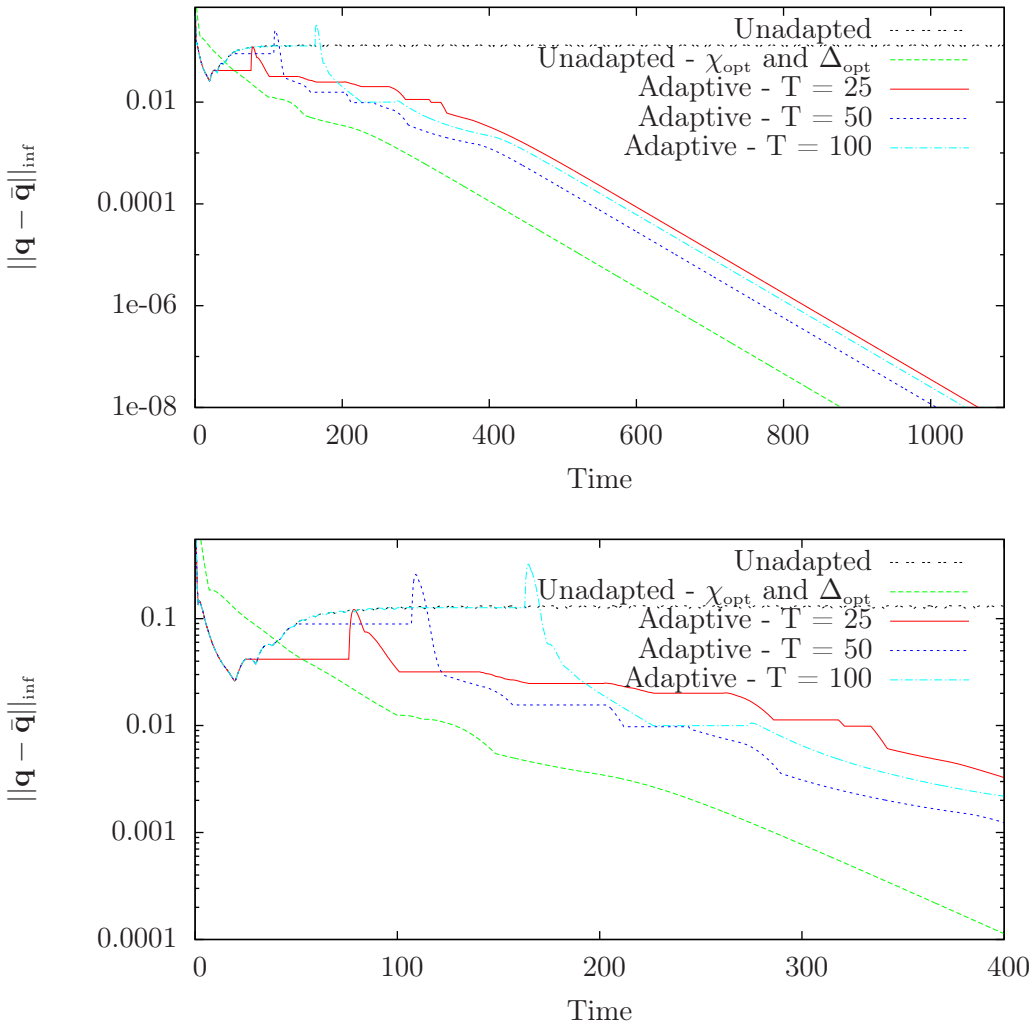
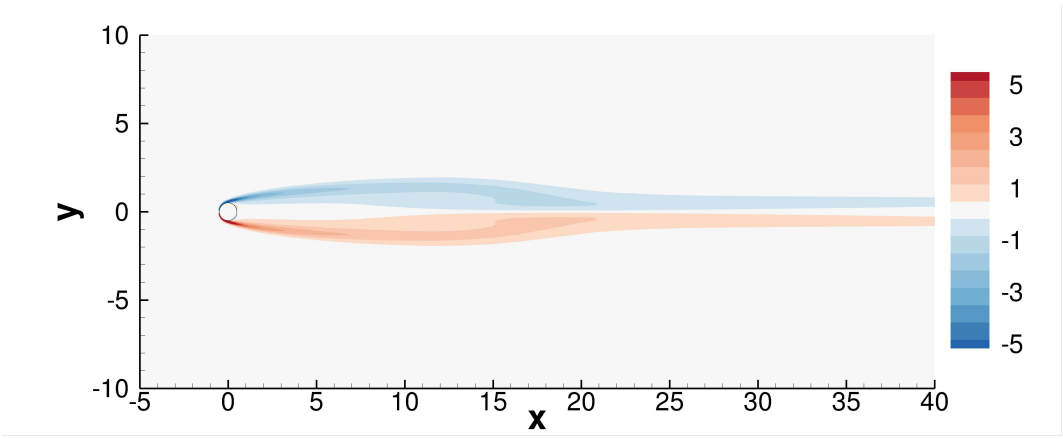
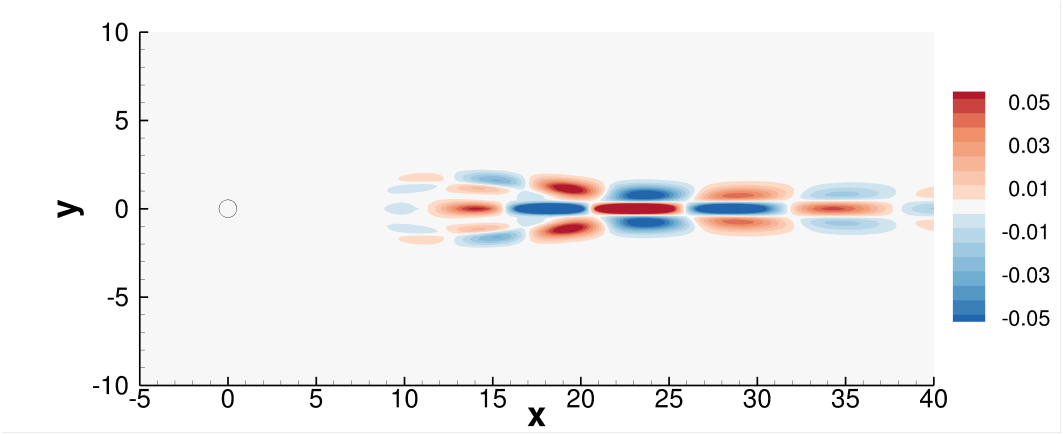


Figure 3.10: Convergence history of the adaptive SFD method for $T = 25$, $T = 50$ and $T = 100$ with $\chi_{\text{init}} = 1$ and $\Delta_{\text{init}} = 0.5$. The cases of the unadapted SFD method for $\chi = 1$, $\Delta = 0.5$ (curve called "Unadapted") and for the optimum parameters $\chi_{\text{opt}} = 0.451$, $\Delta_{\text{opt}} = 3.144$ are also reported.

are $-15 \leq x \leq 100$ and $-30 \leq y \leq 30$. The mesh is made of structured curved quadrilaterals close to the cylinder boundary and in the wake and of triangles elsewhere. No-slip boundary conditions are imposed at the cylinder surface and Dirichlet boundary conditions $(u,\ v) = (1,\ 0)$ are set at the left, top and bottom edges. An



(a) Unstable steady-state obtained by SFD.



(b) Corresponding dominant eigenmode.

Figure 3.11: Vorticity of the steady-state of the incompressible flow past a two-dimensional cylinder at $Re = 300$.

outflow boundary condition is defined at the right edge of the domain. The initial conditions are such that $(u_0,\ v_0) = (0,\ 0)$. As the solution is expected to be smooth, a polynomial order of 7 is used and the time-step is $\Delta t = 0.001$.

The initial parameters are chosen to be $\chi_{\text{init}} = 1$ and $\Delta_{\text{init}} = 2$ and the modified Arnoldi iteration method is executed every $T = 200$ time units. With these settings, the adaptive SFD method calls the stability analysis method five times and converges in a total of 5622 time units (including 377 Arnoldi steps). The steady-state solution and the corresponding dominant eigenmode obtained are shown on Fig. 3.11.

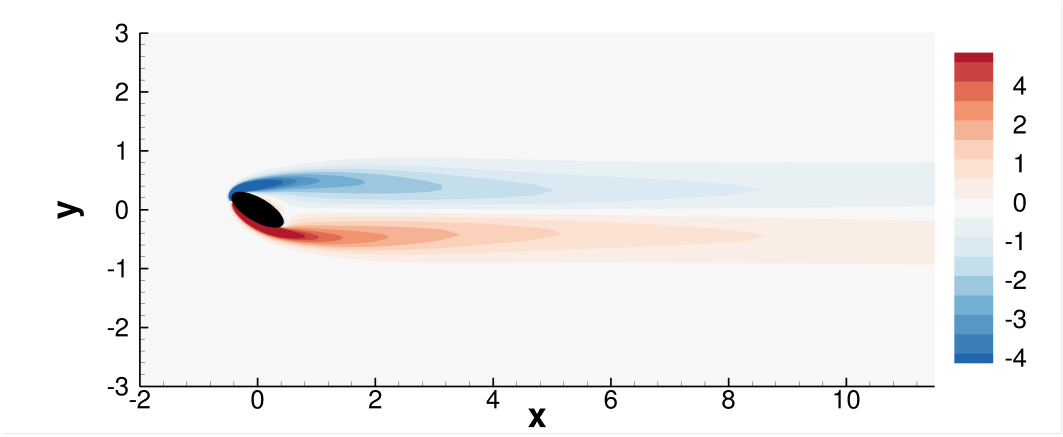
The last approximation of the dominant eigenvalue (computed using a "partially converged" steady-state when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-2}$) is $\tilde{\sigma}_5 = 0.165$ and $\tilde{f}_5 = 0.493$, which is very similar to the dominant eigenvalue computed using the true steady-state (when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$) as the base flow, when $\sigma = 0.163$ and $f = 0.470$ (which corresponds to the optimum parameters $\chi_{\text{opt}} = 0.334$ and $\Delta_{\text{opt}} = 5.951$).

As an extension to Sec. 3.3.1, we showed here that the adaptive SFD method is able to automatically select values of the control coefficient and filter width to control the evolution of the least stable eigenmode of the flow past a cylinder when the Reynolds number is increased. However a long time is necessary to reach convergence.

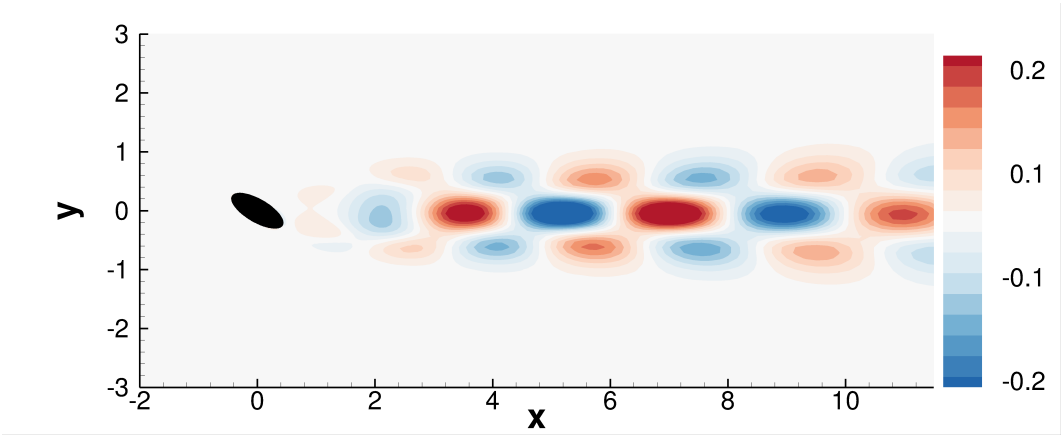
3.3.3 Incompressible flow past an ellipse at $Re = 150$ with an angle of attack

One may argue that if the geometry studied contains axial symmetry (like for the flow past a cylinder), using the SFD method, or any method that provides a steady-state solution of the Navier-Stokes equations, is unnecessary. For example, Mao and Blackburn [74] presented a stability analysis (global and local) of the incompressible flow past a square cylinder (up to $Re = 300$) using a base flow obtained on a semi-domain with symmetric boundary conditions imposed on the horizontal axis. In this section we apply the adaptive method presented in Sec. 3.2 to the incompressible flow past an ellipse at $Re = 150$ with an angle of attack of 30°. The major axis of the ellipse is 1 length unit long and the minor axis is 0.4 length unit long. The major axis is used as characteristic length to calculate the Reynolds number. This test case is presented here because it is a simple example of an unstable flow whose steady-state cannot be obtained with the help of symmetry planes.

The computational domain considered is composed of 856 elements and its dimensions are $-15 \leq x \leq 45$ and $-25 \leq y \leq 25$. The mesh is made of structured curved quadrilaterals close to the ellipse boundary and in the wake and of triangles elsewhere. No-slip boundary conditions are imposed at the ellipse surface and



(a) Unstable steady-state obtained by SFD.



(b) Corresponding dominant eigenmode.

Figure 3.12: Vorticity of the steady-state of the incompressible flow past a two-dimensional ellipse at $Re = 150$ with an angle of attack of 30°.

Dirichlet boundary conditions $(u,\ v) = (1,\ 0)$ are set at the left, top and bottom edges. An outflow boundary condition is defined at the right edge of the domain. The initial conditions are such that $(u_0,\ v_0) = (0,\ 0)$. A polynomial order of 7 is used and the time-step is $\Delta t = 0.0025$.

Nothing is assumed about the stability properties of the flow. We only know that the flow is unstable at this Reynolds number because if DNS simulation is computed, we can observe formation of vortex streets. The initial parameters are chosen to be $\chi_{\text{init}} = 1$ and $\Delta_{\text{init}} = 1$ and the modified Arnoldi iteration method is executed ev-

ery $T = 50$ time units. With these settings, the adaptive SFD method calls the stability analysis method twice and converges in a total of 577 time units (including 82 Arnoldi steps). The steady-state solution and the corresponding dominant eigenmode obtained are shown on Fig. 3.12.

Note that the second approximation of the dominant eigenvalue (computed using a "partially converged" steady-state when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-2}$) is $\tilde{\sigma}_2 = 0.169$ and $\tilde{f}_2 = 1.287$, which is very similar to the dominant eigenvalue computed using the true steady base flow (when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$) where $\sigma = 0.168$ and $f = 1.283$ (which corresponds to the optimum parameters $\chi_{\text{opt}} = 0.792$ and $\Delta_{\text{opt}} = 1.673$).

As this test case was selected essentially to prove that the adaptive SFD method is able to converge towards an unstable steady-state for a geometry that does not contain axial symmetry, we did not find results in the literature to compare with ours. However, through DNS, it is possible to verify the outcome of global stability analysis.

The idea is to introduce a small perturbation, solve in time the linearised Navier-Stokes equations using the steady-state as base flow and track down the perturbation energy evolution. The energy should grow at the same rate as the growth rate returned by the global stability analysis.

At $t = 0$ we introduce a perturbation bump $p(\mathbf{x})$ of the form

$$\begin{cases} p(x) = \varepsilon \; e^{-(x-x_p)^2-(y-y_p)^2}, \\ p(y) = \varepsilon \; e^{-(x-x_p)^2-(y-y_p)^2}, \end{cases} \tag{3.2}$$

where $\varepsilon = 10^{-5}$ and $(x_p,\ y_p) = (-2,\ 0)$.

Then the time evolution of the energy integrated over the computational domain such that

$$E = \frac{1}{2} \int_{\Omega} ||\mathbf{u}||^2 \mathrm{d}\mathbf{x}, \tag{3.3}$$

is evaluated during the execution of the linearised solver. This method has been successfully used by Bagheri *et al.* to retrieve the growth rate of a jet in a crossflow [53].

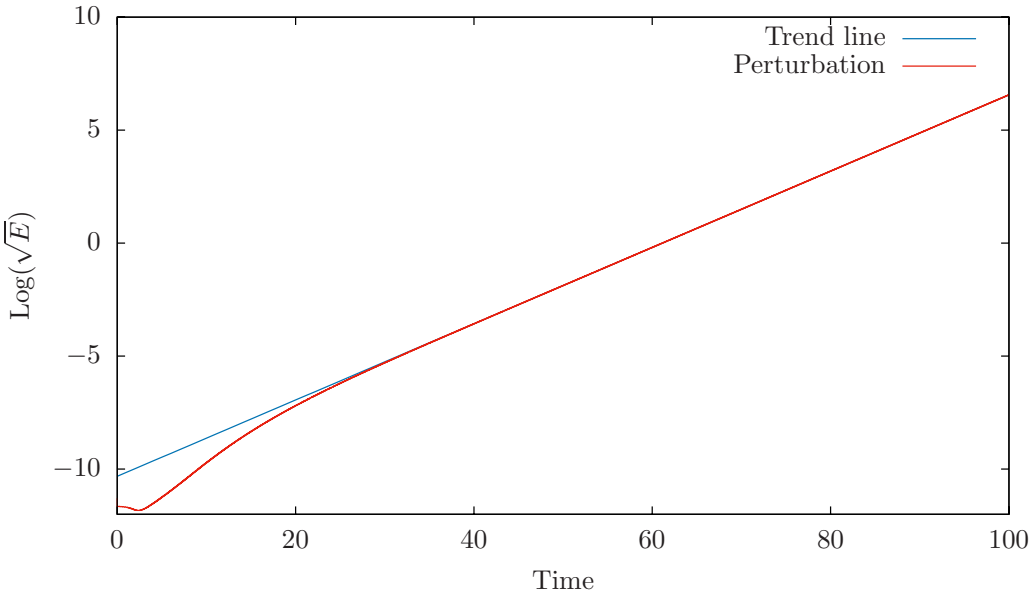


Figure 3.13: Time evolution of the perturbation energy (red) compared to the trend line (blue). The slope of the trend line is 0.169, which is very close to the growth rate of the dominant unstable eigenmode ($\sigma = 0.168$).

This time evolution of the energy is shown in Fig. 3.13 and superimposed with the trend line. The slope of this trend line is 0.169. We recall that the growth rate of the dominant eigenmode is $\sigma = 0.168$. Hence this linear simulation validates the result returned by the global stability analysis solver.

Note that on Fig. 3.13 we observe that the perturbation energy and the trend line do not closely match at the first instants of the simulation. This is explained by the form of the perturbation that is added on top of the base flow at $t = 0$. If this perturbation is chosen to be exactly the dominant eigenmode returned by the stability analysis, we would have observed a perfect matching between the perturbation energy and the trend line with a slope of 0.169 even at the very beginning of the simulation.

Also note that on Fig. 3.13, no oscillations of the energy are observed. This could probably be explained by the fact that the energy, formulated on Eq. 3.3, is spatially averaged over the whole computational domain, reducing the amplitude of

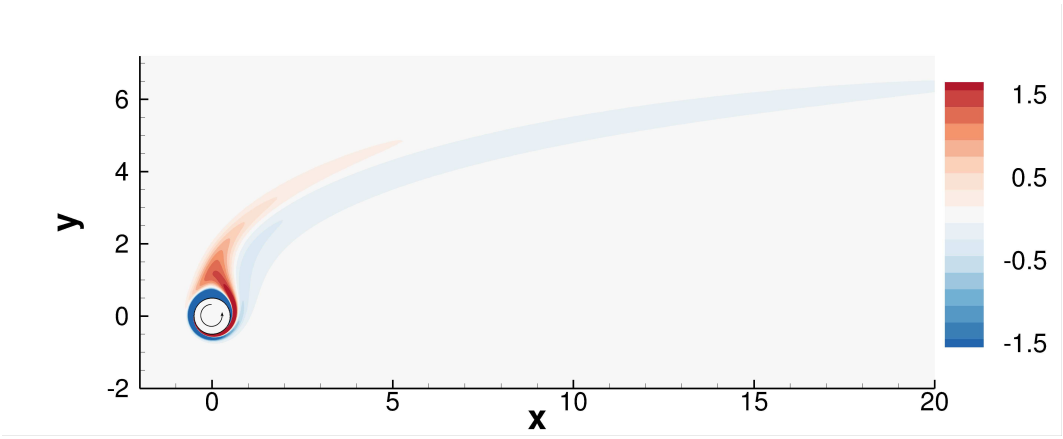
the oscillations.

This test case shows that an adaptive SFD method can easily converge towards an unstable steady flow with no axial symmetry and without any *a priori* knowledge of the dominant eigenvalue.

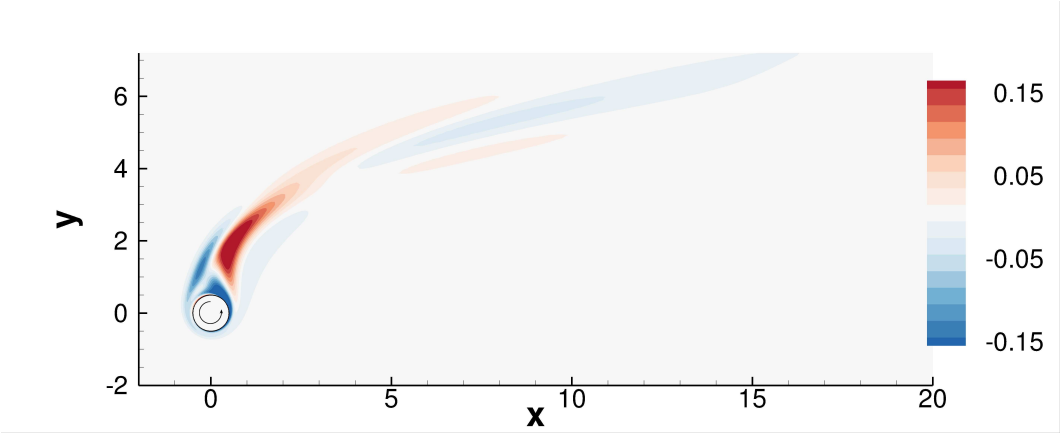
3.3.4 Incompressible flow past a rotating cylinder at $Re = 100$

In this section we aim to find the steady-state solution of an other flow that can not be studied with the help of symmetry planes. This test case is the incompressible two-dimensional flow past a rotating cylinder at $Re = 100$. The rotation of the cylinder impacts the stability of the flow. As shown by Pralits *et. al* [75], for a rotation rate $0 \leqslant \alpha \lesssim 1.8$, the flow is unstable and von Kármán vortex streets are present. They become weaker as α increases. This instability is called shedding mode I. If the rotation rate is in the range $1.8 \lesssim \alpha \leqslant 4.85$, the flow becomes stable. If the rotation rate is increased again, a second unstable mode appears (called shedding mode II) for a range $4.85 \leqslant \alpha \leqslant 5.17$. And eventually, for rotation rates above 5.17, the flow returns to a stable state. Note that this behaviour is only true in two-dimensions. In three-dimensions the presence of shedding mode I and the range of shedding mode II depend on the spanwise wave number [76].

Here we are interested in finding a steady-state solution of the unstable mode II, hence we consider a rotation rate $\alpha = 5$. The computational domain considered is composed of 1044 elements and its dimensions are $-15 \leq x \leq 45$ and $-25 \leq y \leq 25$. The mesh is made of structured curved quadrilaterals close to the cylinder boundary and in the wake and of triangles elsewhere. The mesh is fine close to the cylinder boundary because its rotation induces a strong velocity gradient. Also, the region on the right of the cylinder and for $0 \leq y \leq 8$ is refined because that is where the shedding vortices appear when a DNS simulation is executed. Slip boundary conditions are imposed at the cylinder surface such that $\mathbf{u}\cdot\mathbf{t} = \alpha$ and $\mathbf{u}\cdot\mathbf{n} = 0$, where $\mathbf{u}$ is the velocity vector, $\mathbf{t}$ and $\mathbf{n}$ are the tangential and normal vectors to the surface



(a) Unstable steady-state obtained by SFD.



(b) Corresponding dominant eigenmode.

Figure 3.14: Vorticity of the steady-state of the incompressible flow past a two-dimensional rotating cylinder at $Re = 100$ with a rotation rate $\alpha = 5$.

respectively. Dirichlet boundary conditions $(u,\ v) = (1,\ 0)$ are set at the left, top and bottom edges. An outflow boundary condition is defined at the right edge of the domain. The initial conditions are such that $(u_0,\ v_0) = (0,\ 0)$. A polynomial order of 7 is used and the time-step is $\Delta t = 0.0005$.

To compute the adaptive method presented in Sec. 3.2, the initial control coefficient of the SFD method is chosen to be $\chi_{\text{init}} = 1$. However some care is taken to the selection of the initial filter width. When DNS simulation is executed, we observe that the frequency of the shedding mode II is very low. Thus the initial filter width

of the SFD method must be chosen to be relatively high (*e.g.* $\Delta_{\text{init}} = 5$). Larger filter widths enable us to control instabilities that arise on a larger time scale, but may require an impractically long time to converge. The SFD method is well suited to obtain steady-state solutions of flows with high unstable frequencies. Hence the flow past a cylinder at the unstable shedding mode II is challenging for the SFD method. For this test case the initial filter width is carefully chosen, but nothing is assumed about the dominant eigenvalue of the flow.

Note that choosing a filter width based on the unsteady behaviour of a flow was already suggested by Åkervik *et al.* [10]. Here we use this approach to select Δ_{init} but nothing guarantee that the unadapted SFD method can converge keeping the filter width fixed. The adaptive algorithm presented in this chapter then allows us to update the SFD parameters ensuring convergence.

As a long time is necessary for the flow to be established, a relatively large time $T = 300$ is chosen. This means that the stability analysis method is executed for the first time after 300 time units. With these settings, the adaptive SFD method calls the stability analysis method twice, and converges in a total of 907 time units (including 145 Arnoldi steps). The steady-state solution and the corresponding dominant eigenmode obtained are shown on Fig. 3.14. Note that if a shorter time T is chosen, the stability analysis using a "partially converged" base flow does not capture relevant features of the flow. Hence the approximation of the dominant eigenvalue is not good enough and the corresponding SFD parameters only enable a slow convergence towards the steady-state (*i.e.* about six times slower).

The second approximation of the dominant eigenvalue (computed using a "partially converged" steady-state when $||\mathbf{q}-\bar{\mathbf{q}}||_{\text{inf}} < 10^{-2}$) is $\tilde{\sigma}_2 = 0.039$ and $\tilde{f}_2 = 0.239$, which is very similar to the dominant eigenvalue computed using the true steady-state (when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}} < 10^{-8}$) as the base flow, when $\sigma = 0.036$ and $f = 0.241$ (which corresponds to the optimum parameters $\chi_{\text{opt}} = 0.140$ and $\Delta_{\text{opt}} = 9.610$). This dominant eigenvalue is similar to the one reported by Pralits *et al.* [75].

3.4 Chapter summary

An adaptive procedure to address the issue of selecting appropriate parameters for the SFD method is presented. This algorithm links together a SFD method, a stability analysis method and a one-dimensional model that evaluates the optimum parameters of the SFD method when the dominant eigenvalue is known. This adaptive method is based on several successive computations of the stability analysis method using "partially converged" base flows. This approximation is then used to tune the parameters of the SFD method to ensure an optimum convergence towards the steady-state solution.

This adaptive method was successfully applied to obtain unstable steady-states of several flows. The steady-state of the flow past a cylinder was obtained up to the Reynolds number 300. The steady-state of the flow past an ellipse with an angle of attack at $Re = 150$ was also presented. This test case illustrates the fact that the adaptive SFD method is an appropriate tool to study flows that do not contain axial symmetry. As, to our best knowledge, no results were available in the literature to compare with ours, the outcome of the stability analysis was confirmed by computing a linear simulation to track down the perturbation energy evolution and observing that its trend line agrees with the growth rate. Finally, the steady-state solution of the unstable mode II of the rotating cylinder was presented. This test case was fairly challenging for our algorithm but a careful definition of Δ_{init} and T enables the SFD method to reach a steady-state at an optimum convergence rate. This set of test cases aims to validate our adaptive method, hence we can now use it to investigate the stability three-dimensional vortex dominated flows.

We also provide some guidelines to use the adaptive algorithm detailed in this chapter. If information about the frequency of the flow field is available, it could be used to define the initial value of the filter width. A large Δ_{init} (*e.g.* about 5) should be used for cases with a low frequency and a Δ_{init} close to 1 should be used cases with a relatively high frequency. As large filter widths render the system evolution slow, the time between two consecutive execution of the stability analysis

method should also be large in order to give the flow sufficient time to evolve. In our simulations we used $T \simeq 50 \times \Delta_{\text{init}}$ and were always able to converge towards a steady state solution. If little or no information is available about the frequency of the unstabilised flow field, or if it would be too expensive to compute, we suggest to use $\Delta_{\text{init}} = 2$ and to compute the stability analysis method every 100 time units. For our adaptive algorithm we found that using $\chi_{\text{init}} = 1$ was always appropriate. We also suggest to fix the SFD parameters when $||\mathbf{q} - \bar{\mathbf{q}}||_{\text{inf}}$ becomes smaller than 10^{-2} and to set up a tolerance of 10^{-3} for the execution of the Arnoldi method using "partially converged" base flows.

Our simulations suggest that if the adaptive SFD method fails to converge with a given set of parameters, it should be restarted with a larger time interval T. We found out that a too small T does not give enough time to the flow to properly evolve which may result in a "partially converged" base flow which is a too coarse approximation of the converged steady-state. But restarting the simulation with a larger T may solve the convergence issue. Otherwise, one should increase the initial filter width. By combining these two aspects we were able to find unstable steady-state solutions of all the flows considered.

For all our simulations, the steady-state solution was found in several hundreds time units (several thousands for some cases). Even if the convergence rate of the SFD method is optimal, the time (and the computational cost) necessary to reach the steady-state solution may still be important. It may be that the adaptive SFD method could be then used as a procedure to find an initial guess of a Newton's method. This might allow us to design algorithms capable of reaching unstable steady-state solutions much faster than only using the SFD method. Further research will be necessary to investigate the feasibility of this idea.

CHAPTER 4

TRI-GLOBAL STABILITY ANALYSIS OF TWO CO-ROTATING VORTICES

In this last chapter we present a fully three dimensional global stability analysis of a co-rotating vortex pair. The main challenge of this problem (besides the computational resources needed) is to obtain an unstable steady-state solution of the incompressible Navier-Stokes equations to be used as base flow for the modified Arnoldi method.

Wing-tip vortices, also called trailing vortices, are generated by the displacement of air streams coming from the high-pressure region below the wing into the low-pressure region above the wing. This circulation of air causes the apparition of vortical structures that contribute to a deterioration of the lift coefficient. A trailing vortex is characterised by a high velocity and a low pressure at the core. This pressure drop is associated with a local temperature drop. In some specific conditions the water dew point could be reached (*i.e.* vapour water becomes liquid), then the vortex becomes visible to the naked eye.

During cruising, large commercial aircrafts create vast vortex structures. Any plane (regardless its size) that crosses this vortex wake is susceptible to lose direc-



Figure 4.1: Visualisation of vortices coming from the deflected flaps (top) and from the wing tips (bottom). Photography courtesy of Jeremy Jones (snapshots of videos taken from flugsnug.com).

tion control. During landing and take off, these vortices can interfere with other aircrafts, whether on the ground or approaching. To avoid incidents, regulations restricting the traffic are applied by airports. For example, after a plane takes



Figure 4.2: Landing of a McDonnell Douglas MD-11 (Schiphol airport (NL), July 2005). Vortices coming from the wing tip, the deflected flap and the horizontal tailplane are visible. Photography courtesy of Tim de Groot.

off, a time window of typically two minutes is required before another plane may cross the departure track. Understanding the physical phenomenon behind vortex evolution could improve safety regulations and enable more effective airport traffic management.

In this work we only focus of the interaction between two co-rotating vortices. Co-rotating trailing vortices appear, for example, in the near wake of aircrafts at high lift configurations. Downstream of that near wake region, the co-rotating vortices merge to form a strong vortex that may interact with other surrounding aircrafts.

In Fig. 4.1 the vortices generated by the deflected flaps and the wing-tips are observable. If we consider only one longitudinal half of a plane, these vortices are co-rotating. In Fig. 4.2 both vortices coming from the flap and the wing-tip are visible. By studying the interaction between two co-rotating trailing vortices, we

aim to provide a deeper physical understanding of this flow and of the resulting merged vortex.

Co-rotating trailing vortices also concern the Formula 1 (F1) industry. The front wing of a F1 car is designed to help prevent any "perturbed" upstream flow from propagating below the car which could have dramatic effects on the downforce (*i.e.* the negative lift). If polluted air goes below the car, it could literally take off. This is what happened, for example, to Peter Dumbreck's Mercedes CLR during the 1999 24 Hours of Le Mans race [77]. Hence Formula 1 engineers want to control the evolution of the air coming from the front wing by generating strong vortices. The engineers also have to follow very rigid safety regulations issued by the International Automobile Federation. Understanding the properties of the spatial evolution of co-rotating vortices might be helpful to design efficient F1 front wings.

This chapter is decomposed into four parts. In Sec. 4.1 we present the Q-criterion technique which aims to visualise a vortex into a three dimensional numerical simulation. In Sec. 4.2 we present some generalities about trailing vortices and explain why we made the choice of conducting fully three dimensional numerical simulations. In Sec. 4.3 we detail the settings of the simulations computed and the results are presented in Sec. 4.4.

4.1 Vortex detection

In this section we review the *Q-criterion* technique used to visualise vortices in a three dimensional solution field. This represents a challenge because it consists in extracting a scalar parameter from a vector solution.

There exists a broad variety of vortex detection methods. Some techniques aim to identify the vortex core line, other define a vortical region. Some techniques are Galilean invariant (*i.e.* constant if a uniform velocity field is added), some are not. Some techniques are local (*i.e.* only information of the neighbour element are required), some are global. For example, identifying vorticity magnitude maxima is a local, Galilean invariant line-type method. A description of some of these vortex

detection techniques can be found in [78, 79].

This diversity may be explained by the fact that physical characterisation of a vortex is a very complicated topic. There is no universally accepted definition of a vortex. For example, Lugt defined a vortex as *the rotating motion of material particles around a common center* [80]. Alternatively, Robinson stated that *a vortex exists when its streamlines mapped onto a plane normal to its core exhibit a circular or spiral pattern* [81]. In [82], Portela asserts that *a vortex is comprised of a central core region surrounded by swirling streamlines.*

In this study we chose to use the Q-criterion extraction technique introduced by Hunt in 1988 [83] to identify a vortex within a flow. It is a local, Galilean invariant region-type method, based on the flow field gradient. The gradient of the velocity vector is a tensor that can be decomposed into a symmetric part (denoted $\mathbf{S}$) and an anti-symmetric part (denoted $\mathbf{\Omega}$) such that

$$\nabla \mathbf{u} = \mathbf{S} + \mathbf{\Omega}. \tag{4.1}$$

The tensor $\mathbf{S}$ represents the *rate of strain* (also called the deformation tensor). Note that in the literature, symbols $\mathbf{D}$ or $\mathbf{E}$ can also be used instead of $\mathbf{S}$. This tensor is defined by

$$\mathbf{S} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\top}\right) \quad \Longleftrightarrow \quad \mathrm{S}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{4.2}$$

The anti-symmetric tensor $\mathbf{\Omega}$ represents the *rate of rotation* and is defined by

$$\mathbf{\Omega} = \frac{1}{2}\left(\nabla \mathbf{u} - (\nabla \mathbf{u})^{\top}\right) \quad \Longleftrightarrow \quad \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right). \tag{4.3}$$

We next introduce the characteristic equation of $\nabla \mathbf{u}$ such that

$$\lambda^3 + P\lambda^2 + Q\lambda + R = 0, \tag{4.4}$$

where P, Q and R are the three invariants of the velocity gradient tensor. These invariants can be expressed

$$\begin{aligned} P &= -\mathrm{tr}(\nabla \mathbf{u}) = -S_{ii} \quad (= 0 \text{ for incompressible flows}), \\ Q &= \frac{1}{2}\left(\mathrm{tr}(\nabla \mathbf{u})^2 - \mathrm{tr}(\nabla \mathbf{u}^2)\right) = \frac{1}{2}\left(||\mathbf{\Omega}||^2 - ||\mathbf{S}||^2\right), \\ R &= \det(\nabla \mathbf{u}), \end{aligned} \tag{4.5}$$

where $||\cdot||$ is the Frobenius norm such that $||\mathbf{S}|| = \sqrt{\sum_{i,j} S_{i,j}^2}$.

Hence the Q-criterion defines a vortex as the flow region where the second invariant of the velocity gradient tensor is positive. If we introduce the vorticity tensor $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, then the rate of rotation tensor (4.3) may be written

$$\mathbf{\Omega} = \frac{1}{2} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{4.6}$$

Then the second invariant of the velocity gradient tensor can be expressed directly as function of the vorticity such that

$$Q = \frac{1}{4}||\boldsymbol{\omega}||^2 - \frac{1}{2}||\mathbf{S}||^2. \tag{4.7}$$

When $Q > 0$ then the vorticity dominates the rate of strain, which indicates the presence of a vortex. The Q-criterion technique is widely used in the CFD community because it is easy to implement and relatively cheap to compute (*i.e.* about the same cost as evaluating the vorticity field). However it requires good quality derivatives and may be hard to interpret when several vortices are present in a close vicinity. We made the choice of using this detection method because the problem considered contains only two strong vortices and the underlying high-order spatial discretisation enables highly accurate velocity derivatives.

Note that Jeong & Hussain [84] explained that to use the Q-criterion technique, it is assumed that there is a pressure local minima along the vortex core. Although this condition is not always necessary for a vortex to exist, it applies for the test case studied in this chapter. Hence using a more elaborated vortex detection method, such as the λ_2 criterion, is not necessary [84].

4.2 Generalities on trailing vortices

A reasonable mathematical model of a trailing vortex is known as the *Batchelor vortex* [85]. This model provides an analytical asymptotic (*i.e.* $x >> 1$) steady-state solution of the incompressible Navier-Stokes equations. In contrast to many vortex

models (*e.g.* the Lamb-Oseen vortex [86]), Batchelor's contains an axial component, which is an essential feature of trailing vortices. As detailed be Heaton *et al.* [87], this model is defined, in cylindrical coordinates (x, r, θ), by

$$\begin{cases} U_x \sim 1 + \gamma(x) \exp\left(- \left[\frac{-r}{\delta(x)} \right]^2 \right), \\ U_r \sim 0, \\ U_\theta \sim \frac{\kappa}{r} \left(1 - \exp\left(- \left[\frac{-r}{\delta(x)} \right]^2 \right) \right), \end{cases} \tag{4.8}$$

where

$$\begin{aligned} \kappa &= \text{constant}, \\ \gamma(x) &= \kappa^2 \frac{Re \log(x/Re)}{8x}, \\ \delta(x) &= 2\sqrt{x/Re}. \end{aligned} \tag{4.9}$$

The physical interpretation of these parameters is such that κ is a swirl strength parameter, $\gamma(x)$ is the axial velocity excess parameter at the vortex core and $\delta(x)$ is the local vortex core radius. Also note that, for this coordinate system, $r^2 = (y - y_c)^2 + (z - z_c)^2$, where (y_c, z_c) are the coordinates of the vortex axis.

System (4.8) defines the *true* Batchelor vortex, which is function of the axial spatial coordinate. In [88], Theofilis explains that this model is often misquoted and confused with the so-called *q-vortex*. A q-vortex is a simplified version of Batchelor's model for which the variation in the axial coordinate is removed (*i.e.* the parallel flow approximation is done). It takes the form of an infinitely long "vortex tube", where the Batchelor vortex core shrinks along the stream-wise direction.

In the q-vortex model, the axial velocity excess parameter $\gamma(x)$ and the local vortex core radius $\delta(x)$ are both approximated to be constant. If we introduce the constant dimensionless swirl strength parameter $q = \kappa/(\gamma\delta)$ and the dimensionless free stream velocity a, then a q-vortex is defined by

$$\begin{cases} U_x = a + e^{-r^2}, \\ U_r = 0, \\ U_\theta = \frac{q}{r} \left(1 - e^{-r^2} \right). \end{cases} \tag{4.10}$$

The characteristic velocity γ of a q-vortex represents the velocity excess at the core. This parameter represents the velocity difference between the free stream surrounding flow and the vortex core (where the axial velocity is maximal). Note that we only consider cases where $\gamma > 0$, which means that the vortex evolves faster and in the same direction as the surrounding flow. The characteristic length is the vortex core radius δ. Hence along a q-vortex the Reynolds number is constant and defined by

$$Re = \frac{\gamma\delta}{\nu}. \tag{4.11}$$

Since the interaction of two co-rotating trailing vortices does not have rotational symmetry, using a cylindrical coordinate system is not appropriate. In this thesis, we consider vortices evolving along the z-axis. Then, in Cartesian coordinates, such an isolated q-vortex is defined by

$$\begin{cases} u(x,y,z) = -\dfrac{q(y-y_c)}{r^2}\left(1-e^{-r^2}\right), \\ v(x,y,z) = \dfrac{q(x-x_c)}{r^2}\left(1-e^{-r^2}\right), \\ w(x,y,z) = a + e^{-r^2}, \\ r = \sqrt{(x-x_c)^2+(y-y_c)^2}. \end{cases} \tag{4.12}$$

It is generally the case that isolated, co-rotating or counter-rotating trailing vortices are studied numerically by imposing stream-wise homogeneity to the flow. It means that the simulation is initialised with a vortex (generally a q-vortex) within the computational domain and periodic boundary conditions are used at both ends of the vortex. For example, Laporte & Corjon [89] used this approach to study the three dimensional elliptic instability of a counter-rotating vortex pair and they obtained a good agreement with experimental measurements and theoretical predictions. This instability has also been studied by Le Dizès & Laporte [90] in an attempt to provide a formula to predict the growth rate of any kind of tow-vortex flows. Meunier *et al.* [91] studied the fundamental aspects of two co-rotating vortices and presented a two-dimensional analysis of the merging phenomenon. The three dimensional dynamics is also considered by imposing stream-wise homogeneity to the flow. The differences between two and three dimensional merging were also studied by Orlandi [92] using

DNS simulations and by Nybelen [93, 94] with a LES approach. A linear stability analysis of a co-rotating vortex pair was presented by Roy *et al.* [95] and a transient growth analysis by Mao *et al.* [96].

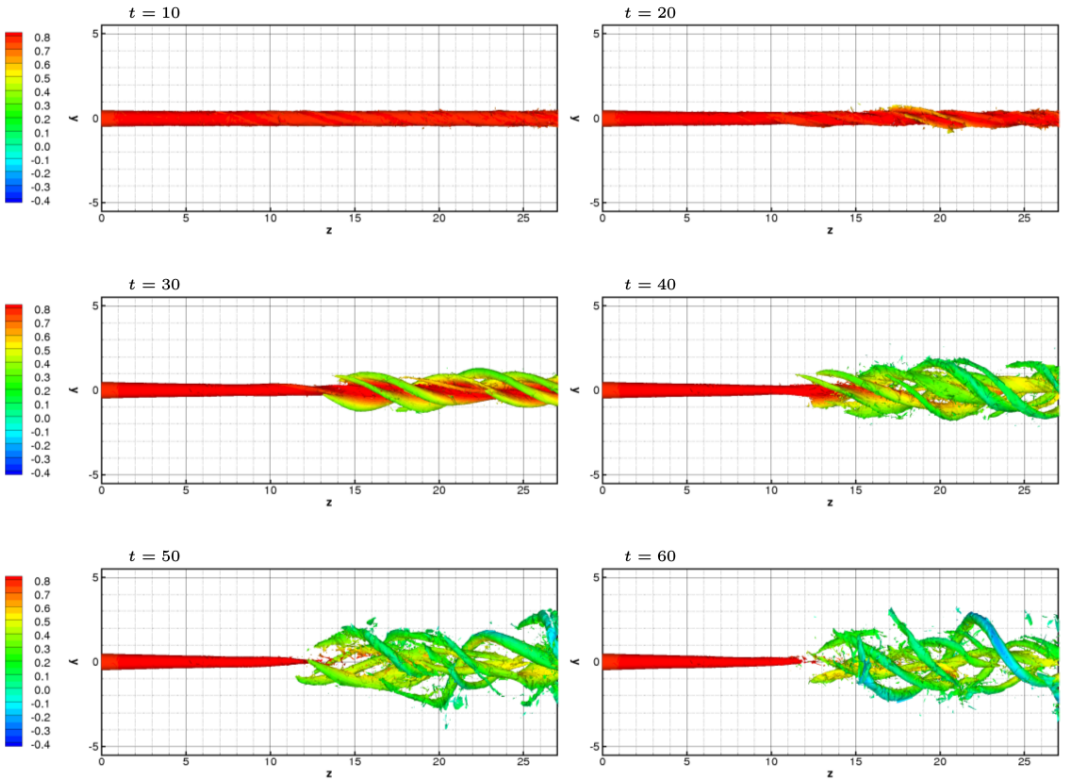
Imposing periodic boundary conditions at both ends of a vortex is computationally advantageous because it allows the use of two-dimensional meshes with a Fourier expansion in the third (stream-wise) dimension. However the assumption of streamwise periodicity does not allow uniform deceleration often associated with vortex breakdown, which is a fundamental physical feature of trailing vortices.

Broadhurst [98], showed that the treatment of the axial direction plays a key role in the numerical simulations. If homogeneity is imposed, then we study the temporal evolution of an infinitely long vortex. However, if the expression of a q-vortex is imposed at the inflow (*i.e.* Eq. 4.12 at $z = 0$) and a standard outflow boundary condition is adopted at the other end, then vortex breakdown may be observed. When homogeneity is imposed in the axial direction, introducing a perturbation leads to the development of a spiral rotating in the same direction as the vortex. However when this condition is replaced by the hard constraint of a fixed inflow condition, introducing a perturbation leads to a localised vortex breakdown. Hence the treatment of the boundary condition may produce fundamental qualitative differences in the output of the numerical simulations. This phenomenon is illustrated on Fig. 4.3.

In this chapter we aim to study the stability of a co-rotating vortex pair. Hence the numerical simulations realised are all fully-three dimensional, with no homogeneity imposed in any direction. The numerical set-up is detailed in Sec. 4.3. Note that at the outflow, we use the boundary condition developed by Dong *et al.* [43] and summarised in Sec. 1.5.3. To our knowledge, the only other study considering the fully three dimensional spatial evolution of a co-rotating vortex pair without imposing periodicity in the steam-wise direction has been presented by Deniau & Nybelen [99]. The authors were able to observe a vortex breakdown and simulated the development of short-wavelength elliptic instabilities. However their simulations were restricted by extremely high computational costs.



(a) Two-dimensional mesh with Fourier expansion in the stream-wise direction.



(b) Fully three dimensional DNS, with classical outflow boundary condition.

Figure 4.3: Evolution of a perturbation within a q-vortex at $Re = 1000$, $q = 0.8$ and $a = 0$. The two simulations aim to compute identical problems with different discretisations. Axial homogeneity is imposed on (a) and classical outflow is used on (b). Vortex visualised by iso-surfaces of $\lambda_2 = -0.4$ (and coloured by axial velocity component for the bottom figure). Figures courtesy of Michael Broadhurst (taken from [97] (top) and his PhD thesis [98] (bottom)).

4.3 Numerical set-up

In this section we present the details of the numerical set-up. The computational domain has a box-shape of $(L_x, L_y, L_z) = (100, 100, 250)$ and the flow evolves along the z-axis. These large dimensions have been chosen to minimise the influence of the boundary conditions. The three dimensional domain has been constructed by extruding the two-dimensional mesh presented on Fig. 4.4. This domain is composed uniquely of quadrilateral elements refined at the center where most of the physics is contained. Long stretched elements are used close to the lateral boundaries. The refined region is a square $[-7, 7]^2$ surrounded by a circle of radius 12.5 because the flow is expected to have a somewhat cylindrical general shape. The central square area counts 11×11 squares. The smallest element of this mesh has a comparable size to the vortex core which we define to have a dimension of one unit length. Through high-order spatial discretisation we are able to obtain accurate solutions. In total, the two-dimensional domain counts 385 elements. Extruded along the z-axis for 166 equispaced layers, the three dimensional domain is composed of a total of 63910 hexahedral elements. The elements in the central refined region are nearly cubical, with a stretching coefficient of 1.18 along the z-axis.

Note that the computational domain considered is uniform along the streamwise direction. It has been noticed by Broadhurst [98] that numerical resolution may influence the vortex breakdown location. Breakdown arises earlier on coarse grids. Thus we avoided using coarser elements near the outflow boundary to prevent a "numerical breakdown" due to insufficient resolution. However this comes with a heavy compensation in terms of computational resources required to execute the numerical simulations.

The lateral boundaries (*i.e.* normal to the x and y-axis) are treated with symmetry conditions. The dimensions of the computational domain in x and y are such that the co-rotating vortex pair may be considered isolated. At the outflow, Dong's condition is used (see Sec. 1.5.3). This condition is appropriate to model truncation of unbounded domains. At the inflow, two trailing q-vortices separated by a distance

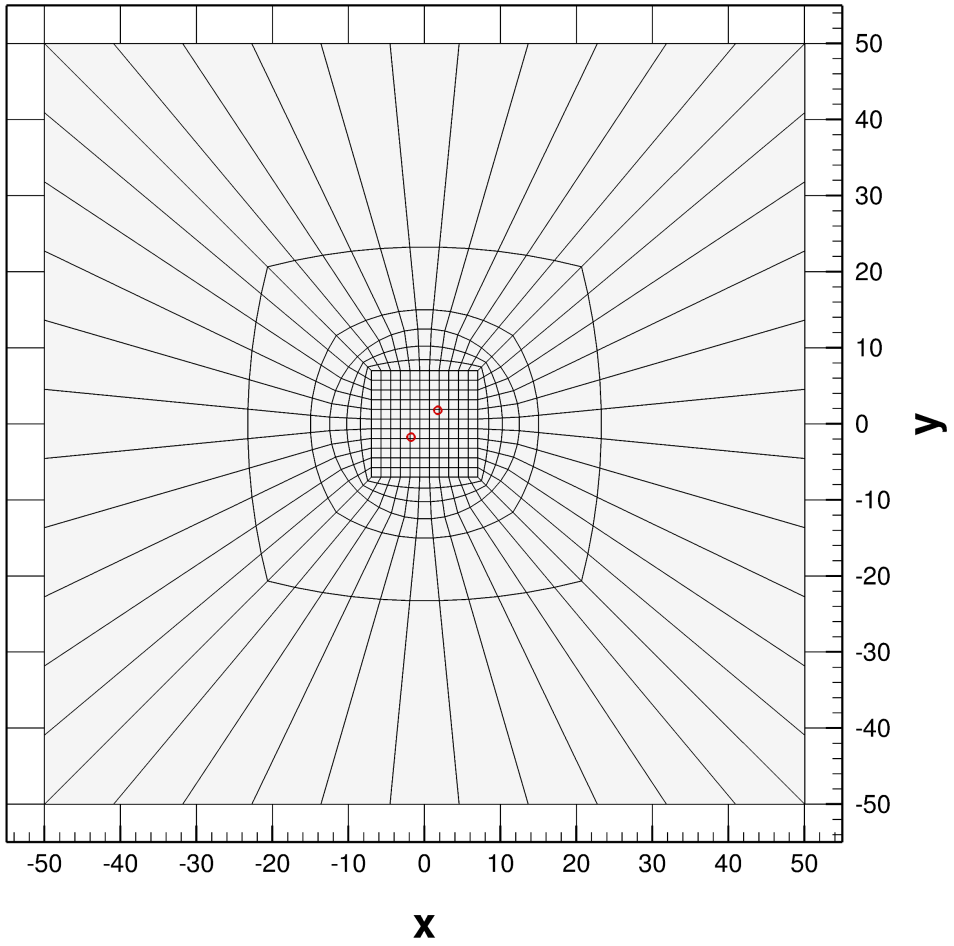


Figure 4.4: High-order computational domain at $z = 0$. The red circles represent the position of the vortices cores imposed at the inflow.

$d = 5$ are imposed. The equations of this Dirichlet boundary condition are

$$\begin{cases} u(x,y,z) = -\dfrac{q(y-y_{c1})}{r_1^2}\left(1-e^{-r_1^2}\right) - \dfrac{q(y-y_{c2})}{r_2^2}\left(1-e^{-r_2^2}\right), \\ v(x,y,z) = \dfrac{q(x-x_{c1})}{r_1^2}\left(1-e^{-r_1^2}\right) + \dfrac{q(x-x_{c2})}{r_2^2}\left(1-e^{-r_2^2}\right), \\ w(x,y,z) = a + e^{-r_1^2} + e^{-r_2^2}, \\ r_1 = \sqrt{(x-x_{c1})^2 + (y-y_{c1})^2}, \\ r_2 = \sqrt{(x-x_{c2})^2 + (y-y_{c2})^2}, \end{cases} \tag{4.13}$$

where $(x_{c1},\ y_{c1}) = (-2.5/\sqrt{2},\ -2.5/\sqrt{2})$ and $(x_{c2},\ y_{c2}) = (2.5/\sqrt{2},\ 2.5/\sqrt{2})$ are the center of the two vortices. Note that the two-vortices studied are identical (*i.e.*

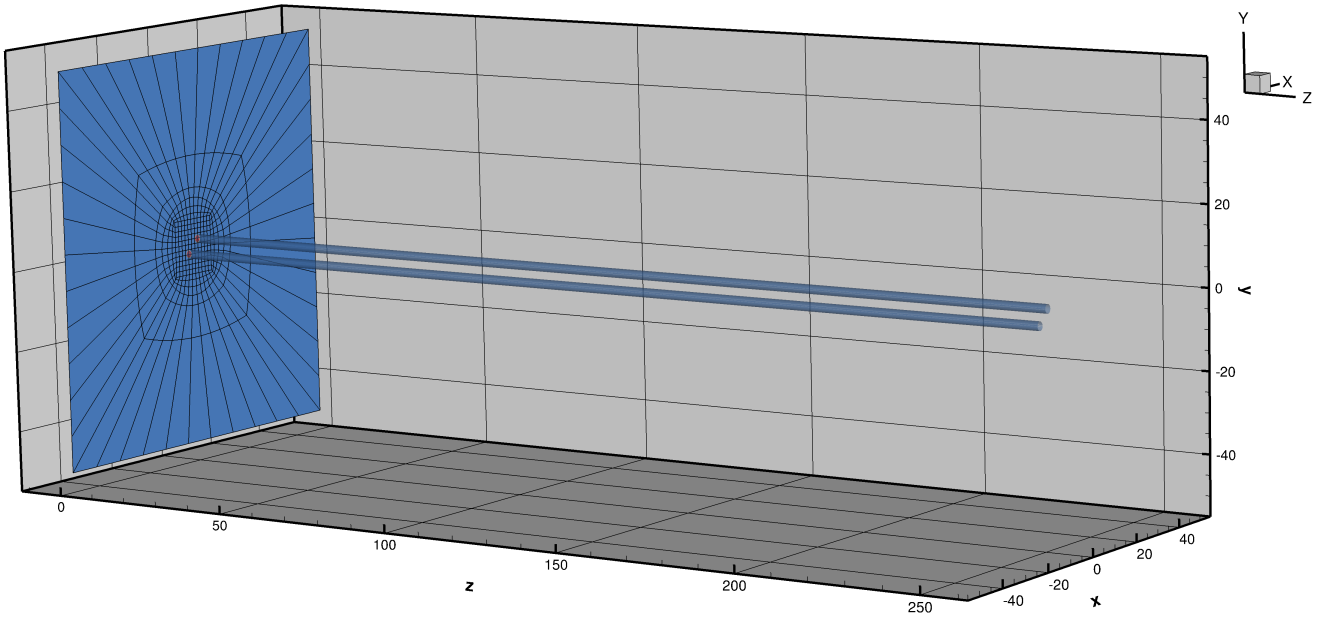


Figure 4.5: Initial conditions (iso-surface of Q-criterion). The left-hand side plane represents the position of the inflow boundary. The vortices evolve along the z-axis (from left to right).

same dimensions, same intensity, same sense of rotation).

Although we impose the analytical expression of q-vortices at the inflow, nothing is assumed in terms of stream-wise evolution. Hence we can say that this simulation models a co-rotating Batchelor pair.

At $t = 0$, the simulation is initialised with two q-vortices as shown on Fig. 4.5. This is imposed by defining (4.13) in the whole domain. Note that even if one isolated q-vortex is a good approximation of solution of the Navier-Stokes equations (*i.e.* because it is a simplification of the true Batchelor vortex), this is not the case for two co-rotating vortices. These two initial "tubes" are chosen because it enables a "gentle" initialisation of the simulation. If the simulation is initialised simply with a uniform stream-wise velocity, the vortices coming from the inflow generate a wavefront (until they reach the outflow) that may be numerically problematic.

The second-order velocity correction scheme presented in Sec. 1.2 is used to disctretise the problem in time. A continuous Galerkin method with a modified

modal expansion basis (see Sec. 1.1) is chosen as space discretisation scheme. The spatial expansion basis is such that the polynomial order is $P = 5$ (*i.e.* with 6 expansion points of each vertices of the hexahedron of the mesh). In our simulations, the characteristic velocity (which represents the difference of velocity between one vortex core and the free-stream region), is chosen such that $\gamma = 1$. The radius of one vortex is the characteristic length and is selected such that $\delta = 1$. Then the Reynolds number is simply defined by $Re = 1/\nu$.

4.4 Numerical simulations

In this section we present the numerical simulations leading to the establishment of a steady-state solution of a flow containing two isolated co-rotating trailing vortices and present the results of a Tri-Global stability analysis.

The steady-state solution has been obtained using the encapsulated SFD method presented in Chapter 2 and implemented in Nektar++. The two vortices are identical and such that the dimensionless swirl strength parameter $q = 0.8$ and the dimensionless free stream velocity $a = 0.5$. These vortex parameters are selected to match the settings of the simulations presented by Hoessler [100].

The size of this problem is such that it requires relatively high computational capacities. To execute the simulations we used the Imperial College's "CX2" highly parallel processing system. We used between 256 and 384 CPUs. On this system, depending on the Reynolds number, the SFD method took between three and seven days to converge and the modified Arnoldi solver took between one and nine days to converge. The queuing time, which could go up to a week, also needs to be added on top of the running time. Hence obtaining a steady-state solution and computing a Tri-Global stability analysis was a very time consuming task.

The flow is considered at two different Reynolds numbers that are $Re = 250$ and $Re = 600$. We show that those cases present qualitative differences in terms of stability properties.

4.4.1 Two co-rotating trailing vortices at Re = 250

In this section we present the steady-state solution of the interaction between two co-rotating trailing vortices at $Re = 250$ and present a stability analysis of this flow. At this Reynolds number, no stabilisation technique was required to ensure good behaviour of the numerical simulation since no abrupt divergence of the solution was observed.

The SFD method was used to obtain a steady-state solution. The SFD parameters selected are a control coefficient $\chi = 1$ and a filter width $\Delta = 2$. As with these parameters the simulation was converging fairly fast, we did not have to use the adaptive procedure presented in Chapter 3 to ensure convergence.

The steady-state solution obtained is presented on Fig. 4.6. The vortices are visualised through four iso-surfaces of Q-criterion ($Q = 0.02$, $Q = 0.02 + 8/300$, $Q = 0.02 + 16/300$ and $Q = 0.1$). We recall that the Q-criterion is defined such that positive values represent regions where the vorticity dominates the rate of strain. Hence the largest value of Q allows us to detect the strongest vortical regions. Alternatively, when Q is close to zero, we can spot regions where the vorticity only marginally dominates the strain. This region is larger but generally harder to obtain as it requires a very high accuracy of the computed derivatives. The iso-surfaces shown are smooth due to the high-order underlying spatial disctretisation scheme used.

On Fig. 4.6 we observe that the two co-rotating trailing vortices imposed on the inflow (*i.e.* on the left) rapidly merge and the flow evolves towards the outflow as one single vortex. Note that the steady-state presented on Fig. 4.6 is constant in time. The inflow on the left is fixed. The different angles shown on this figure are simply presented to allow us to apprehend the three dimensional features of the flow.

This steady-state solution has been used as base flow to compute a Tri-Global stability analysis. This analysis is realised using the modified Arnoldi iteration method summarised in Sec. 1.4.2 and implemented in Nektar++. We recall that by

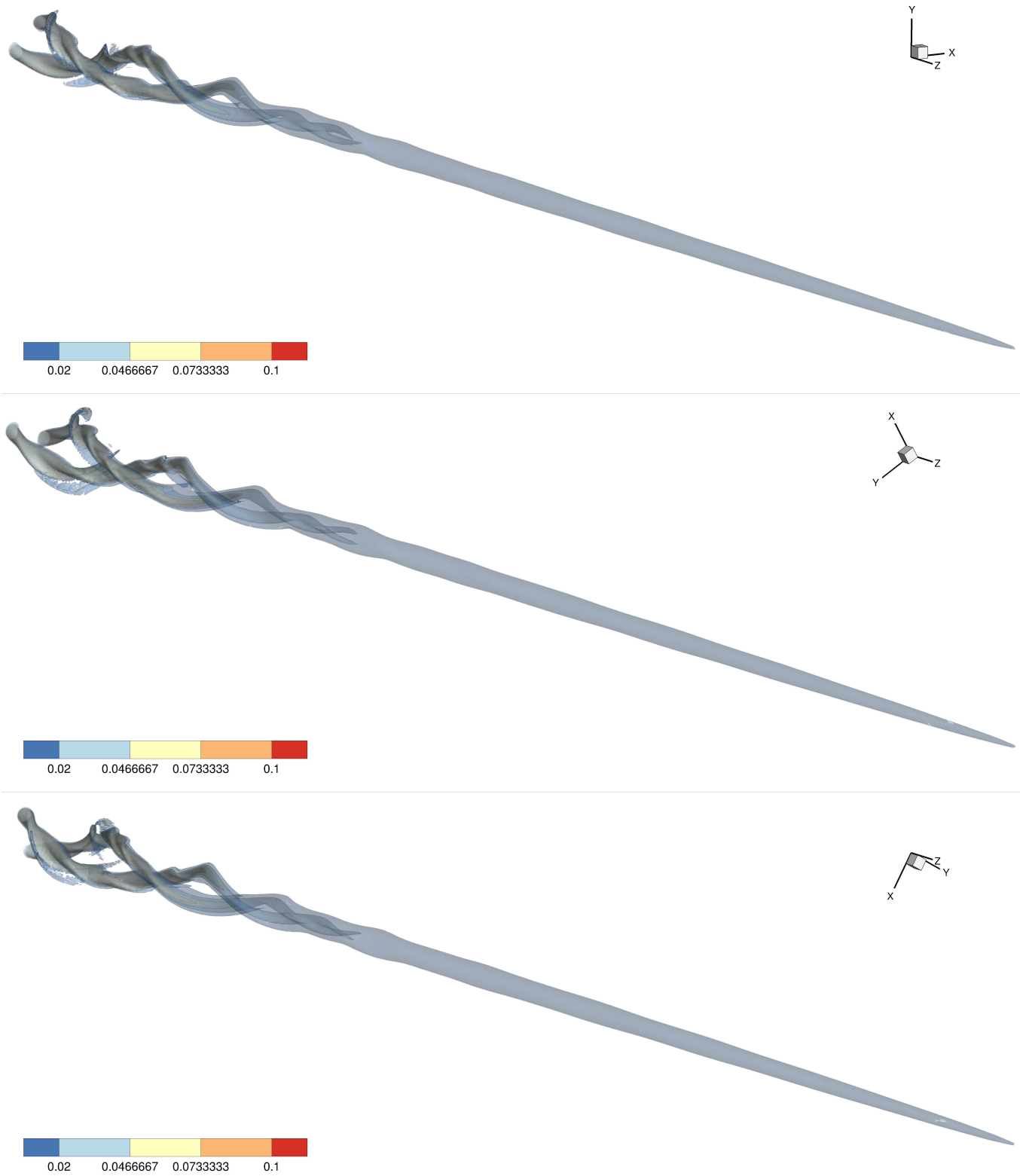


Figure 4.6: Steady-state of a co-rotating vortex pair at $Re = 250$ for $a = 0.5$ and $q = 0.8$ visualised from three different angles with iso-surfaces of Q-criterion.

Tri-Global we mean that the analysis computed is fully three dimensional and the perturbations are not constrained to be periodic in any specific dimension.

Due to the very large size of the problem, we had to reduce the accuracy of the stability analysis solver in order to obtain a result in a reasonable time frame. Even though we used a polynomial order $P = 5$ to compute the steady-state, we used $P = 4$ to execute the modified Arnoldi iteration method. With this modification, we were able to have the algorithm converged in approximately nine days. The dominant eigenvalue returned by this Tri-Global stability analysis is such that

$$\sigma = -0.023 \quad \text{and} \quad f = 0.763. \tag{4.14}$$

The fact that the growth rate of the dominant eigenvalue is negative indicates that the flow is *stable*. Hence the use of the SFD method was unnecessary. Running the DNS code and waiting until the flow becomes constant in time would have been enough to obtain this steady-state solution. However the fact that this flow is stable was unexpected. That is why a control method was used to obtain a steady base flow.

On Fig. 4.7 we show the superimposition of the steady-state solution shown on Fig. 4.6 and the eigenmode returned by the modified Arnoldi method. For clarity, only one iso-surface of Q-criterion of the steady-state is displayed (the one $Q = 0.02$) and coloured in red to clearly distinguish the base flow and the eigenmode. Iso-surfaces of the stream-wise vorticity are used to display the dominant eigenmode.

This superimposition allows us to assert that the least stable mode of the interaction between two co-rotating vortices at $Re = 250$ appears after merging. Even though this mode is stable, it is of interest to note that it does not come directly from the interaction between the two inflow vortices. This mode is located on the part of the flow where the two vortices are fully merged into a unique trailing vortex. This behaviour can be seen as two co-rotating vortices acting as perturbations for the resulting merged single vortex.

As, to our best knowledge, no results are available in the literature to compare with ours, we used DNS to increase confidence in the stability analysis presented

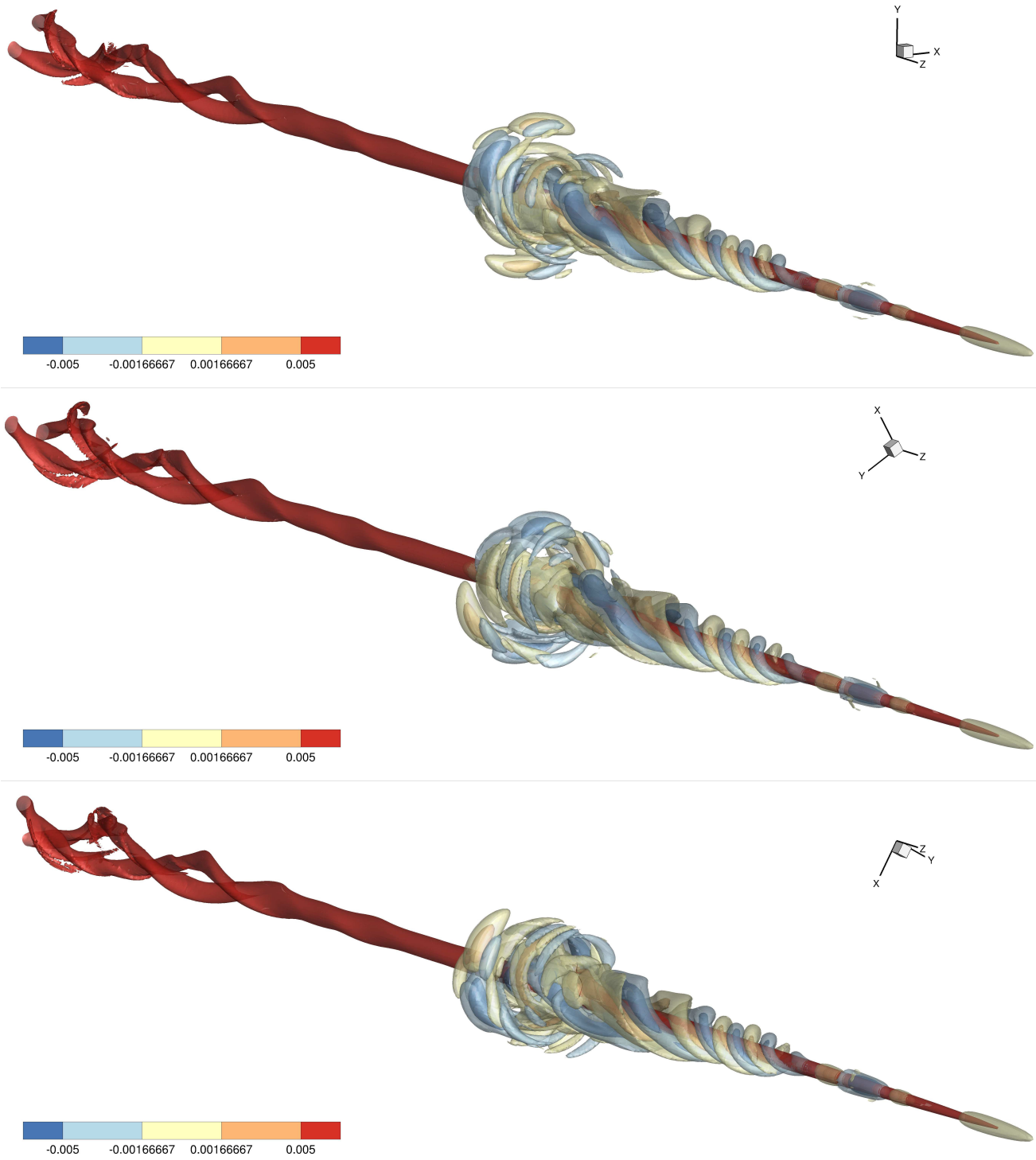


Figure 4.7: Superimposition of the steady-state and the dominant eigenmode of a co-rotating vortex pair at $Re = 250$ for $a = 0.5$ and $q = 0.8$ visualised from three different angles. Iso-surfaces of z-vorticity of the eigenmode and iso-surface of the Q-criterion $Q = 0.02$ of the steady-state.

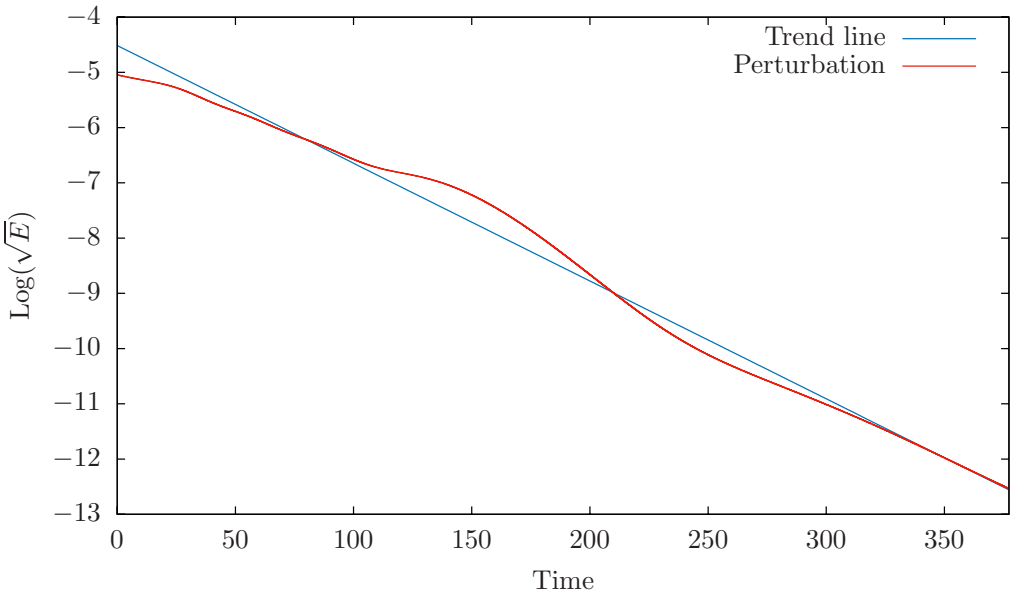


Figure 4.8: Time evolution of the perturbation energy (red) compared to the trend line (blue). The slope of the trend line is -0.021, which is very close to the growth rate of the dominant unstable eigenmode ($\sigma = -0.023$).

in this section. We used the procedure presented by Bagheri *et al.* for the case of a jet in a crossflow [53] and presented in 3.3.3 to compare the value of the growth rate. We added a three dimensional perturbation bump $p(\mathbf{x})$ (see Eq. 3.2), with $(x_p,\ y_p,\ z_p) = (0,\ 0,\ 1)$, at $t = 0$ on top of the steady base flow and tracked down the perturbation energy evolution while executing the linearised Navier-Stokes solver (using a polynomial order $P = 5$). The time evolution of the energy is shown in Fig. 4.8 and superimposed with the trend line.

Note that on Fig. 4.8, similarly as on Fig. 3.13, only marginal oscillations of the energy are observed. This is probably due to the spatial averaging of the energy, reducing the oscillations amplitude.

The slope of this trend line is -0.021. We recall that the growth rate of the dominant eigenmode is $\sigma = -0.023$. Hence this linear simulation increases the confidence in the result returned by the Tri-Global stability analysis. The difference between the two values could be explained by the fact that different polynomial

orders were used to execute the stability solver and the linearised Navier-Stokes solver.

4.4.2 Two co-rotating trailing vortices at Re = 600

In this section we present the steady-state solution of the interaction between two co-rotating trailing vortices at $Re = 600$ and present a stability analysis of this flow. In comparison to Sec. 4.4.1, this higher Reynolds number makes the numerical simulation abruptly diverge.

To remove spurious numerical oscillations we used the de-aliasing technique (presented in Sec. 1.5.1) that consists of adding quadrature points when the non-linear term of the Navier-Stokes equations is evaluated. By using this technique we were able to postpone the moment when the simulation would diverge but not to completely suppress this phenomenon. Hence we also used the spectral vanishing viscosity method (summarised in Sec. 1.5.2) which adds viscosity to the higher modes of the expansion basis. The SVV kernel used is presented on Fig. 4.9 and is such that $M = 5$ and $M_{cut} = 2$. The viscosity added on the highest expansion mode is a hundred times greater than the viscosity of the rest of the flow. Combining these two stabilisation techniques allowed us to avoid divergence of the numerical simulations.

Note that these techniques are used to stabilise the *numerical scheme*, not the *flow*. The response to infinitesimal perturbation of the flow is not addressed with the de-aliasing or SVV methods. They only guarantee a good behaviour of the code in cases marginally resolved where the viscous operator is dominated by the advection operator.

The SFD method was used to obtained a steady-state solution. The SFD parameters are such that the control coefficient $\chi = 1$ and the filter width $\Delta = 2$. As for the case $Re = 250$, the SFD method was converging towards a steady-state solution with these control parameters at $Re = 600$. Hence the adaptive procedure presented in Chapter 3 was not necessary to ensure convergence.

The steady-state solution obtained is presented on Fig. 4.10. The vortices are

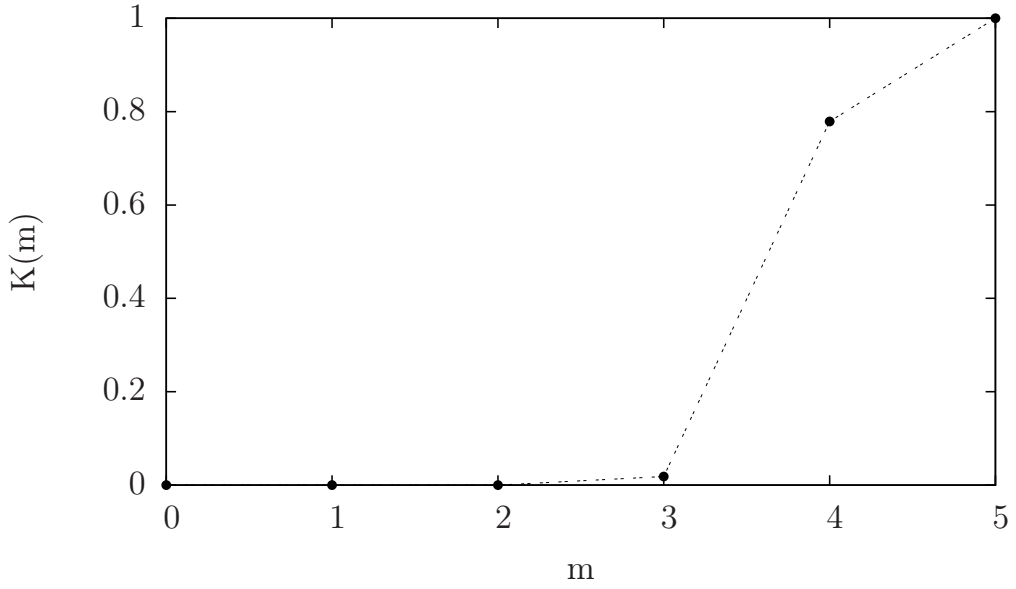


Figure 4.9: SVV kernel for $M = 5$ and $M_{cut} = 2$. This curve represents the percentage of viscosity added on each mode. On the highest mode a viscosity of $\nu_{SVV} = 100 \times \frac{1}{Re} = \frac{100}{600} = \frac{1}{6}$ is added.

visualised through four iso-surfaces of Q-criterion ($Q = 0.02$, $Q = 0.02 + 8/300$, $Q = 0.02 + 16/300$ and $Q = 0.1$). The two co-rotating vortices spiral around each other until $z \simeq 185$ before eventually merging and evolving as one single vortex towards the outflow. By comparing Fig. 4.6 and Fig. 4.10, we observe that increasing the Reynolds number delays the merging of the two vortices.

On Fig. 4.10, we also observe the presence of secondary smaller vortices in the vicinity of the two main vortices. These structures indicate the presence of strong shear stresses when the main vortices start interacting between each other. Close to the inflow we can detect the presence of numerical noise which is due to an insufficient resolution of the simulation in this region. However using a higher polynomial order was not considered because it would lead to an impractical increase of computational resources required to execute this case. Instead, we used the SVV methods which enabled a substantial reduction of the numerical noise. Without SVV, the solution is nearly entirely polluted by under-resolved small scale structures that generate a

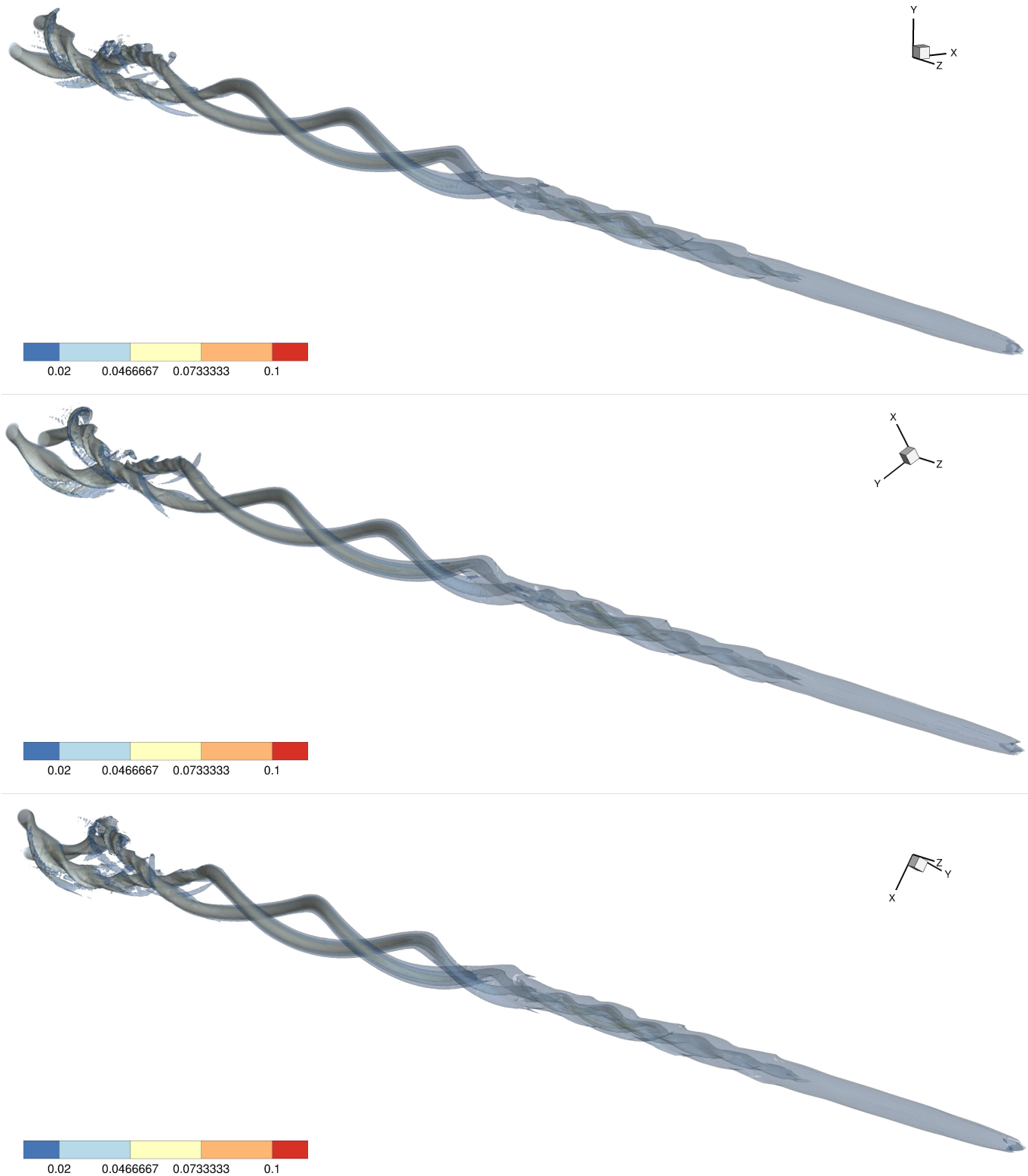


Figure 4.10: Steady-state of a co-rotating vortex pair at $Re = 600$ for $a = 0.5$ and $q = 0.8$ visualised from three different angles with iso-surfaces of Q-criterion.

build-up through time of the energy and eventually lead to an abrupt divergence of the solution.

This steady-state solution has been used as base flow to compute a Tri-Global stability analysis. This analysis was realised using the modified Arnoldi iteration method summarised in Sec. 1.4.2 and implemented in Nektar++. As in Sec. 4.4.1, a lower polynomial order of $P = 4$ was used to execute the stability analysis solver (where $P = 5$ was used to execute the SFD method). The dominant eigenvalue returned by the stability analysis solver is such that

$$\sigma = 0.003 \quad \text{and} \quad f = 1.262. \tag{4.15}$$

At $Re = 600$, the interaction between two co-rotating trailing vortices is an *unstable* phenomenon because this flow has a positive growth rate. Hence, contrary to the case $Re = 250$, the SFD method was necessary to obtain a steady-state solution of the Navier-Stokes equations. On Fig. 4.11 we present a superimposition of the unstable base flow and the dominant eigenmode related to (4.15). This eigenmode is identified by two iso-surfaces of the stream-wise vorticity whereas the base flow is displayed by the iso-surface of the Q-criterion $Q = 0.02$.

On this figure we observe that the dominant unstable eigenmode starts developing in between the two main vortices at $z \simeq 25$. Then the eigenmode evolves as localised wave packets spiralling around the steady base flow. In [53], Bagheri *et al.* present a Tri-Global stability analysis of a jet in a crossflow. The authors observed that the dominant unstable eigenmode takes the shape of symmetric wave packets localised along the counter-rotating vortex pair resulting from the interaction between the jet and the crossflow. Even though the dominant eigenmode of the flow presented in this section is not symmetric, the similar evolution into wave packets observed in our simulations is of interest to note.

These wave packets reveal a behaviour that is qualitatively different from the eigenmode presented on Fig. 4.7 where we observed that the (stable) eigenmode appears after the merging of the two co-rotating vortices. This fundamental difference indicates that the instability of the flow containing two co-rotating trailing vortices

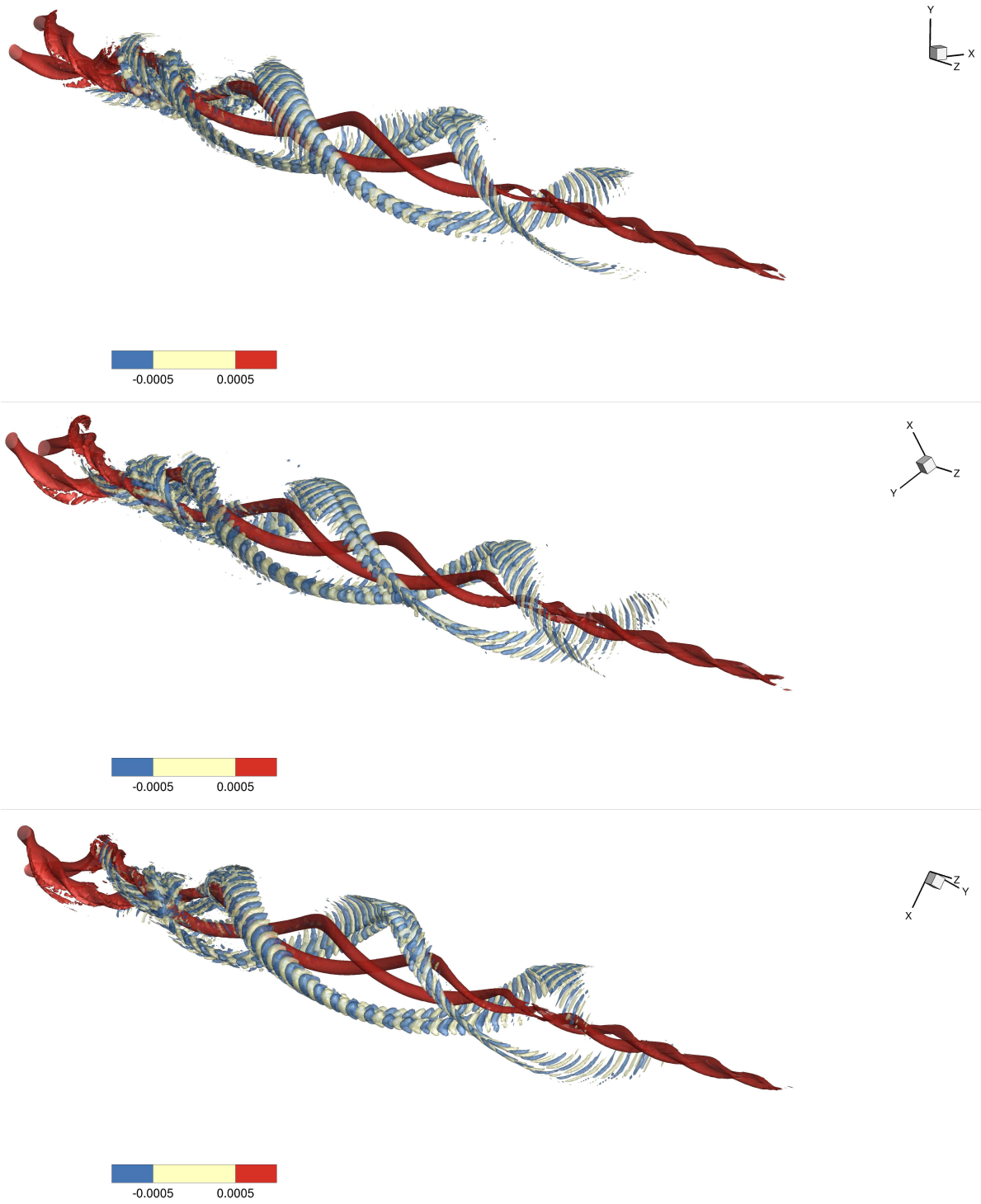


Figure 4.11: Superimposition of the steady-state and the dominant eigenmode of a co-rotating vortex pair at $Re = 600$ for $a = 0.5$ and $q = 0.8$ visualised from three different angles. Iso-surfaces of z-vorticity of the eigenmode and iso-surface of the Q-criterion $Q = 0.02$ of the steady-state.

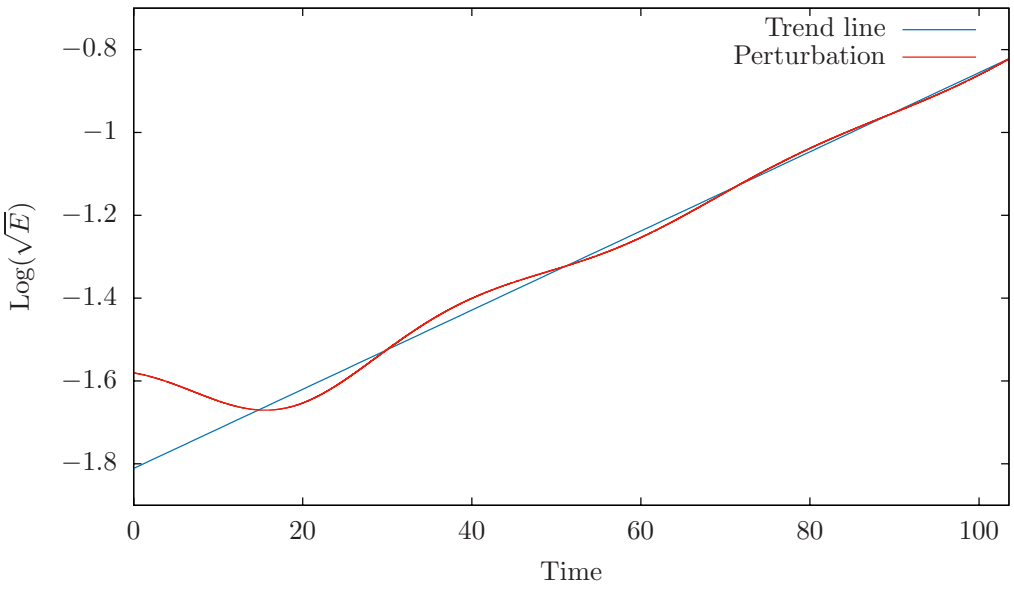


Figure 4.12: Time evolution of the perturbation energy (red) compared to the trend line (blue). The slope of the trend line is 0.009, which is of the same order of magnitude than the growth rate of the dominant unstable eigenmode ($\sigma = 0.003$).

originates directly from the interaction between these two vortices.

Similarly as for the case $Re = 250$ presented in Sec. 4.4.1, we used DNS as an attempt to validate the growth rate returned by the modified Arnoldi iteration method. We introduced a perturbation of the form of a bump on top of the steady base flow obtained by SFD and tracked down the evolution of the perturbation energy. The time evolution of the energy is shown in Fig. 4.12 and superimposed with the trend line.

Similarly as on Fig. 4.8, only marginal oscillations (decaying in time) are observable on Fig. 4.12. This phenomenon is probably due to the spatial averaging of the energy, reducing the oscillations amplitude.

The slope of this trend line is 0.009. We recall that the growth rate of the dominant eigenmode is $\sigma = 0.003$. These values are slightly different from one another but are of the same order of magnitude. The difference between the two values could be explained by the fact that different polynomial orders were used to

execute the stability solver and the linearised Navier-Stokes solver. Hence we can only consider this linear simulation is a preliminary attempt to validate the growth rate returned by the Tri-Global stability analysis. Further work would be necessary to confirm that result.

4.5 Chapter summary

In this chapter we presented a steady-state solution and a stability analysis of the flow containing two isolated co-rotating vortices at two different Reynolds number. The simulations are fully three dimensional and no axial periodicity is imposed.

At $Re = 250$ (based on the vortex core and the velocity excess at the core) the Tri-Global stability analysis computed revealed that the flow is stable because the growth rate corresponding to the dominant eigenmode is negative. This mode is situated after the complete merging of the two co-rotating vortices. Hence the least stable part of the system does not come from the interaction of the two co-rotating vortices but from the introduction of perturbations into the resulting merged vortex. Further analysis on the single trailing vortex would be necessary to classify this eigenmode and determine if it might be associated with a vortex breakdown.

For the case $Re = 600$ the steady-state solution was obtained with the SFD method by keeping the control parameters fixed to $\chi = 1$ and $\Delta = 2$. Adapting these parameters along the algorithm execution was not necessary to ensure convergence towards a steady base flow. The outcome is quantitatively different from the case $Re = 250$ though. At $Re = 600$ the dominant unstable eigenmode starts developing in between the two co-rotating vortices. This indicates that the instability mechanism of this flow originates directly from the interaction of the two main trailing vortices. Although additional work would be necessary to confirm the growth rate and frequency returned by the modified Arnoldi method.

CONCLUSION AND FUTURE PROSPECTS

In this thesis we presented flow stability analysis computed using high-order discretisation methods. The central part of this work was to design an algorithm capable of converging towards unstable steady-state solution of the Navier-Stokes equations.

In Chapter 1 we reviewed the numerical methods used throughout this thesis. The spectral/hp element method was used to discretise the fluid motion equations in space because it combines high accuracy and geometrical flexibility. The time discretisation used was a splitting method that decomposes the Navier-Stokes equations into a series of elliptic problems where the primary variables are separated. We reviewed the main aspects of global linear flow stability analysis and explained that the base flow around which the fluid system is linearised must be a genuine solution of the Navier-Stokes equations. That is why throughout this thesis we sought for steady-state solutions to use as base flows to evaluate the stability properties of a flow. Finally, we highlighted some stabilisation techniques that prevent numerical simulations to abruptly diverge and introduced the Nektar++ code that is currently being developed at Imperial College London in collaboration with the University of Utah. All the numerical simulations presented in this document have been computed using Nektar++

When we reviewed generalities about steady-state solvers and global linear stability analysis, we presented the time-stepper approach that has been introduced by Tuckerman and Barkley [9]. This approach consists in adapting an unsteady flow

solver in order to perform many more operations than simply evaluating the look of a flow field at a specific time. As developing a code to solve the fluid motion equations is generally a considerable task, it could be highly valuable to be able to perform different types of analysis with it. The stability analysis solver that was already implemented in Nektar++ is based on this method. The time-stepper approach strongly inspired the research on implementing a steady-state solver into Nektar++ in the least intrusive way possible.

In Chapter 2 we presented an encapsulated formulation of the selective frequency damping (SFD) method. This method consists of looking at the classical SFD method from a different angle and making it fit within the framework of splitting methods. With this approach, it is possible to separate the solution of the unsteady non-linear flow problem and the linear problem modelling the feedback control and the first order low-pass time filter. Hence the encapsulated SFD method only needs to call an existing code at each time step and apply a linear operator to its output. This method was implemented as a wrapper function into Nektar++ which means that no low level modifications of the unsteady solver were required.

The stability of the SFD method (*i.e.* its ability to converge towards a steady-state solution) was also analysed in Chapter 2. We showed that the SFD method does not introduce a loss of stability but that it can not be applied to problems with pure optimal growths of the instabilities. We observed that a relationship can be drawn between the ability of the SFD method to stabilise the unstable one-dimensional problem $u^n = \lambda_D u^{n+1}$ and the stabilisation of a flow whose dominant unstable eigenvalue is λ_D. However this observation is based on the assumption that the dominant eigenvalue of the flow studied is known. Which is generally not the case. Furthermore this analysis does not provide any information on the rate of convergence towards a steady base flow.

In Chapter 3 we introduced an adaptive procedure to address the issue of selecting appropriate parameters for the SFD method. This algorithm links together a SFD method, a stability analysis method and a one-dimensional model that evaluates the optimum parameters of the SFD method when the dominant eigenvalue is

known. This method has been implemented into Nektar++ and showed good convergence performance towards unstable steady-state solutions for a variety of bluff body flows. Very limited knowledge of the flow behaviour was necessary to ensure convergence at an optimum rate. We also computed stability analysis of these flows. The most challenging of all was the flow past a rotating cylinder at $Re = 100$ and a rotation rate $\alpha = 5$. We showed that the stability properties returned by our computations agree with results found in the literature.

In Chapter 4 we presented a steady-state solution and a Tri-Global stability analysis of two identical co-rotating vortices at two different Reynolds numbers. These simulations were fully three-dimensional and no periodicity was imposed in the stream-wise direction. At $Re = 250$ we showed that the flow is stable and that the least stable eigenmode is situated after complete merging of the two inflow vortices. Hence the least stable part of the system does not originate from the interaction of the two co-rotating vortices but from the introduction of perturbations into the resulting merged vortex. At $Re = 600$ the dominant unstable eigenmode starts developing in between the two co-rotating vortices. This behaviour indicates that the instability mechanism of this flow originates directly from the interaction of the two main trailing vortices. Note that the nature of this flow was such that the adaptive SFD method was not necessary to converge towards an unstable steady-state solution. The encapsulated SFD method was converging relatively fast even keeping the control coefficient and filter width fixed all along the simulations. At higher Reynolds number, this may not be the case and the adaptive SFD method may be necessary.

The future prospects of this PhD thesis can be divided into two categories. First, as the adaptive SFD method has been implemented in Nektar++ and is fully operational, it could be used to obtain unstable steady-state solutions for a whole range of problems (provided an oscillatory growth of the instabilities). As it has been designed to be a wrapper function, it could be applied to compressible flow problems even though in this work we only focused on incompressible flows.

Second, the analysis presented in Chapter 4 can be seen as a starting point

to study co-rotating trailing vortices which arise in aeronautical applications. For example, to study the very complex vortical structures originating from a Formula 1 front wheel, the idea would be to take a cut through a simulation, use these data as boundary conditions and apply the methodology described in Chapter 4 to analyse the three-dimensional stability of isolated vortices. Also further analysis on the single trailing vortex could be performed to determine if the stable eigenmode obtained at $Re = 250$ could be associated with the vortex breakdown phenomenon.

BIBLIOGRAPHY

[1] P. J. Schmid. Nonmodal stability theory. *Annu. Rev. Fluid Mech.*, 39:129–162, 2007.

[2] Nektar++. `http://www.nektar.info`, 2015.

[3] C. D. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. de Grazia, S. Yakovlev, J.-E. W. Lombard, D. Ekelschot, B. E. Jordi, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. M. Kirby, and S. J. Sherwin. Nektar++: An open-source spectral/hp element framework. *Comput. Phys. Commun.*, 192:205–219, 2015.

[4] J. N. Reddy. *An introduction to the finite element method*, volume 2. McGraw-Hill New York, 1993.

[5] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods for CFD.* Oxford University Press, second edition, 2005.

[6] G. E. Karniadakis, M. Israeli, and S. A. Orszag. High-order splitting methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 97(2):414–443, 1991.

[7] S. A. Orszag, M. Israeli, and M. O. Deville. Boundary conditions for incompressible flows. *SIAM J. Sci. Comput.*, 1(1):75–111, 1986.

[8] E. Ferrer, D. Moxey, R. H. J. Willden, and S. J. Sherwin. Stability of projection methods for incompressible flows using high order pressure-velocity pairs of same degree: Continuous and Discontinuous Galerkin formulations. *Commun. Comp. Phys.*, 16(3):817–840, 2014.

[9] L. S. Tuckerman and D. Barkley. Bifurcation analysis for timesteppers. *Numerical methods for bifurcation problems and large-scale dynamical system*, pages 453–566, 2000.

[10] E. Åkervik, L. Brandt, D. S. Henningson, J. Hœpffner, O. Marxen, and P. Schlatter. Steady solutions of the Navier-Stokes equations by selective frequency damping. *Phys. Fluids*, 18(6):068102, 2006.

[11] L. S. Tuckerman, F. Bertagnolio, O. Daube, P. Le Quéré, and D. Barkley. Stokes preconditioning for the inverse Arnoldi method. In D. Henry and A. Bergeon, editors, *Continuation Methods for Fluid Dynamics (Notes on Numerical Fluid Mechanics, Vol. 74)*, pages 241–255. Vieweg, 2000.

[12] H. A. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13(2):631–644, 1992.

[13] R. Schreiber and H. B. Keller. Driven cavity flows by efficient numerical techniques. *J. Comput. Phys.*, 49(2):310–333, 1983.

[14] K. A. Cliffe, A. Spence, and S. J. Tavener. The numerical analysis of bifurcation problems with application to fluid mechanics. *Acta Numerica*, 9(00):39–131, 2000.

[15] H. A. Dijkstra, F. W. Wubs, A. K. Cliffe, E. Doedel, I. F. Dragomirescu, B. Eckhardt, A. Y. Gelfgat, A. L. Hazel, V. Lucarini, A. G. Salinger, et al. Numerical bifurcation methods and their application to fluid dynamics: analysis beyond simulation. *Commun. Comput. Phys*, 15:1–45, 2014.

[16] C. Wales, A. L. Gaitonde, D. P. Jones, D. Avitabile, and A. R. Champneys. Numerical continuation of high reynolds number external flows. *Int. J. Numer. Meth. Fl.*, 68(2):135–159, 2012.

[17] C. Wales, A. L. Gaitonde, and D. P. Jones. An initial study of the flow around an aerofoil at high Reynolds number unsing continuation. *Int. J. Bifurcat. Chaos*, 22(10), 2012.

[18] C. Wales, A. L. Gaitonde, and D. P. Jones. Continuation methods applied to the 2D Navier–Stokes equations at high Reynolds numbers. *Int. J. Numer. Meth. Fl.*, 70(10):1258–1289, 2012.

[19] S. Cherubini, J.-C. Robinet, and P. De Palma. The effects of non-normality and nonlinearity of the Navier-Stokes operator on the dynamics of a large laminar separation bubble. *Phys. Fluids*, 22(1):014102, 2010.

[20] K.-T. Chen, C.-C. Tsai, W.-J. Luo, and C.-N. Chen. Multiplicity of steady solutions in a two-sided lid-driven cavity with different aspect ratios. *Theor. Comp. Fluid Dyn.*, 27(6):767–776, 2013.

[21] K.-T. Chen, C.-C. Tsai, W.-J. Luo, C.-W. Lu, and C.-H. Chen. Aspect ratio effect on multiple flow solutions in a two-sided parallel motion lid-driven cavity. *J. Mech.*, pages 1–8, 2014.

[22] C. K. Mamun and L. S. Tuckerman. Asymmetry and Hopf bifurcation in spherical Couette flow. *Phys. Fluids*, 7(1):80–91, 1995.

[23] L. S. Tuckerman and K. Boronska. Multiplicité extrême de motifs convectifs. *Résumés des exposés de la 11ème Rencontre du Non-Linéaire Paris 2008*, page 14, 2008.

[24] B. Hof, P. G. J. Lucas, and T. Mullin. Flow state multiplicity in convection. *Phys. Fluids*, 11(10):2815–2817, 1999.

[25] J. Sanchez, F. Marques, and J. M. Lopez. A continuation and bifurcation technique for Navier-Stokes flows. *J. Comput. Phys.*, 180(1):78–98, 2002.

[26] J. Kim and T. R. Bewley. A linear systems approach to flow control. *Annu. Rev. Fluid Mech.*, 39:383–417, 2007.

[27] C. D. Pruett, T. B. Gatski, C. E. Grosch, and W. D. Thacker. The temporally filtered Navier-Stokes equations: Properties of the residual stress. *Phys. Fluids*, 15(8):2127, 2003.

[28] C. D. Pruett, B. C. Thomas, C. E. Grosch, and T. B. Gatski. A temporal approximate deconvolution model for large-eddy simulation. *Phys. Fluids*, 18:028104, 2006.

[29] P. J. Schmid and L. Brandt. 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. *Appl. Mech. Rev.*, 66(2):024803, 2014.

[30] M. P. Juniper. Local linear stability analysis. Lecture notes from the AIM-ED International Graduate School in Flow Stability, Transition and Control, Stockholm, Sweden, 2013.

[31] M. P. Juniper and V. Hanifi, A. Theofilis. Modal Stability Theory. Lecture notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Appl. Mech. Rev.*, 66(2):024804, 2014.

[32] V. Theofilis. Global linear instability. *Annu. Rev. Fluid Mech.*, 43:319–352, 2011.

[33] D. Barkley, H. M. Blackburn, and S. J. Sherwin. Direct optimal growth analysis for timesteppers. *Int. J. Numer. Meth. Fl.*, 57(9):1435–1458, 2008.

[34] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al. *LAPACK Users' guide*, volume 9. SIAM, 1999.

[35] G. Rocco. *Advanced Instability Methods using Spectral/hp Discretisations and their Applications to Complex Geometries*. PhD thesis, Imperial College London, 2014.

[36] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. SIAM, 1998.

[37] C. D. Cantwell, D. Barkley, and H. M. Blackburn. Transient growth analysis of flow through a sudden expansion in a circular pipe. *Phys. Fluids*, 22(3):034101, 2010.

[38] R. M. Kirby and S. J. Sherwin. Aliasing errors due to quadratic nonlinearities on triangular spectral/*hp* element discretisations. *J. Eng. Math.*, 56(3):273–288, 2006.

[39] R. M. Kirby and G. E. Karniadakis. De-aliasing on non-uniform grids: algorithms and applications. *J. Comput. Phys.*, 191(1):249–264, 2003.

[40] G. Mengaldo, D. de Grazia, D. Moxey, P. Vincent, and S. J. Sherwin. Dealiasing techniques for high-order spectral element methods on regular and irregular grids. under review in J. Comput. Phys., Mar. 2015.

[41] E. Tadmor. Convergence of spectral methods for nonlinear conservation laws. *SIAM J. Numer. Anal.*, 26(1):30–44, 1989.

[42] R. M. Kirby and S. J. Sherwin. Stabilisation of spectral/*hp* element methods through spectral vanishing viscosity: Application to fluid mechanics modelling. *Comput. Method. Appl. M.*, 195(23):3128–3144, 2006.

[43] S. Dong, G. E. Karniadakis, and C. Chryssostomidis. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *J. Comput. Phys.*, 261:83–105, 2014.

[44] J.-E. W. Lombard, D. Moxey, J. Hoessler, S. Dhandapani, M. J. Taylor, and S. J. Sherwin. Transient simulation of a wingtip vortex at $Re_c = 1.2 \cdot 10^6$. Submitted to *AIAA Journal* in January 2015.

[45] D. Barkley. Linear analysis of the cylinder wake mean flow. *EPL (Europhysics Letters)*, 75(12):750, 2006.

[46] E. Åkervik, J. Hoepffner, U. Ehrenstein, and D. S. Henningson. Optimal growth, model reduction and control in a separated boundary-layer flow using global eigenmodes. *J. Fluid Mech.*, 579(1):305–314, 2007.

[47] D. S Henningson and E. Åkervik. The use of global modes to understand transition and perform flow control. *Phys. Fluids*, 20(3):031302, 2008.

[48] F. Alizard, J.-C. Robinet, and X. Gloerfelt. A domain decomposition matrix-free method for global linear stability. *Comput. Fluids*, 66:63–84, 2012.

[49] J. H. Tu and C. W. Rowley. An improved algorithm for balanced POD through an analytic treatment of impulse response tails. *J. Comput. Phys.*, 231(16):5317–5333, 2012.

[50] B. Pier. Local and global instabilities in the wake of a sphere. *J. Fluid Mech.*, 603:39–61, 2008.

[51] M. Embacher and H. F. Fasel. Direct numerical simulations of laminar separation bubbles: investigation of absolute instability and active flow control of transition to turbulence. *J. Fluid Mech.*, 747:141–185, 2014.

[52] J.-C. Loiseau, J.-C. Robinet, S. Cherubini, and E. Leriche. Investigation of the roughness-induced transition: global stability analyses and direct numerical simulations. *J. Fluid Mech.*, 760:175–211, 2014.

[53] S. Bagheri, P. Schlatter, P. J. Schmid, and D. S. Henningson. Global stability of a jet in crossflow. *J. Fluid Mech.*, 624:33–44, 2009.

[54] P. Schlatter, S. Bagheri, and D. S. Henningson. Self-sustained global oscillations in a jet in crossflow. *Theor. Comp. Fluid Dyn.*, 25(1-4):129–146, 2011.

[55] M. Ilak, P. Schlatter, S. Bagheri, and D. S. Henningson. Bifurcation and stability analysis of a jet in cross-flow: onset of global instability at a low velocity ratio. *J. Fluid Mech.*, 696:94–121, 2012.

[56] A. Peplinski, P. Schlatter, P. F. Fischer, and D. S. Henningson. Stability Tools for the Spectral-Element Code Nek5000: Application to Jet-in-Crossflow. In *Spectral and High Order Methods for Partial Differential Equations-ICOSAHOM 2012*, pages 349–359. Springer, 2014.

[57] A. Peplinski, P. Schlatter, and D. S. Henningson. Global stability and optimal perturbation for a jet in cross-flow. *Eur. J. Mech. B-Fluid.*, 49:438–447, 2015.

[58] L. Massa. Mach number effects on the global mixing modes induced by ramp injectors in supersonic flows. *J. Fluid Mech.*, 757:403–431, 2014.

[59] J. W. Nichols and P. J. Schmid. The effect of a lifted flame on the stability of round fuel jets. *J. Fluid Mech.*, 609:275–284, 2008.

[60] G. J. Chandler. *Sensitivity analysis of low-density jets and flames*. PhD thesis, University of Cambridge, 2011.

[61] G. J. Chandler, M. P. Juniper, J. W. Nichols, and P. J. Schmid. Adjoint algorithms for the Navier-Stokes equations in the low Mach number limit. *J. Comput. Phys.*, 231(4):1900–1916, 2012.

[62] P. F. Fischer, J. W. Lottes, and S. G. Kerkemeier. Nek5000 Web page, 2008. http://nek5000.mcs.anl.gov.

[63] B. E. Jordi, C. J. Cotter, and S. J. Sherwin. Encapsulated formulation of the selective frequency damping method. *Phys. Fluids*, 26(3):034101, 2014.

[64] I. Faragó. Splitting methods and their application to the abstract Cauchy problems. In *Numerical Analysis and Its Applications*, pages 35–45. Springer, 2005.

[65] S. Blanes and P. C. Moan. Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods. *J. Comput. Appl. Math.*, 142(2):313–330, 2002.

[66] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5(3):506–517, 1968.

[67] E. Vyazmina. *Bifurcations in a swirling flow*. PhD thesis, École Polytechnique, 2010.

[68] L. S. Tuckerman. Dynamical systems, `http://www.pmmh.espci.fr/~laurette/class/DynSys.pdf`. University Lecture, 2015.

[69] H. B. Keller. Lectures on numerical methods in bifurcation problems. *Applied Mathematics*, 217:50, 1987.

[70] C. Norberg. Fluctuating lift on a circular cylinder: review and new measurements. *J. Fluids Struct.*, 17:57–96, 2003.

[71] S. J. Sherwin and H. M. Blackburn. Three-dimensional instabilities and transition of steady and pulsatile axisymmetric stenotic flows. *J. Fluid Mech.*, 533:297–327, 2005.

[72] B. E. Jordi, C. J. Cotter, and S. J. Sherwin. An adaptive selective frequency damping method. *Phys. Fluids*, 27:094104, 2015.

[73] J. Nocedal and S. J. Wright. *Numerical optimization*, volume 2. Springer, 1999.

[74] X. Mao and H. M. Blackburn. The structure of primary instability modes in the steady wake and separation bubble of a square cylinder. *Phys. Fluids*, 26(7):074103, 2014.

[75] J. O. Pralits, L. Brandt, and F. Giannetti. Instability and sensitivity of the flow around a rotating circular cylinder. *J. Fluid Mech.*, 650:513–536, 2010.

[76] J. O. Pralits, F. Giannetti, and L. Brandt. Three-dimensional instability of the flow around a rotating circular cylinder. *J. Fluid Mech.*, 730:5–18, 2013.

[77] Video Mercedes CLR crash during the 1999 24 Hours of Le Mans. `https://www.youtube.com/watch?v=Ow3rxq7U1mA`, 2015.

[78] M. Jiang, R. Machiraju, and D. Thompson. Detection and visualization of vortices. *The Visualization Handbook*, page 295, 2005.

[79] S. J. Kamkar, A. M. Wissink, V. Sankaran, and A. Jameson. Feature-driven cartesian adaptive mesh refinement for vortex-dominated flows. *J. Comput. Phys.*, 230(16):6271–6298, 2011.

[80] H. J. Lugt. The dilemma of defining a vortex. In *Recent developments in theoretical and experimental fluid mechanics*, pages 309–321. Springer, 1979.

[81] S. K. Robinson. Coherent motions in the turbulent boundary layer. *Annu. Rev. Fluid Mech.*, 23(1):601–639, 1991.

[82] L. M. Portela. *Identification and characterization of vortices in the turbulent boundary layer*. PhD thesis, Stanford University, 1998.

[83] J. C. R. Hunt, A. Wray, and P. Moin. Eddies, stream, and convergence zones in turbulent flows. *Center for turbulence research report CTR-S88*, pages 193–208, 1988.

[84] J. Jeong and F. Hussain. On the identification of a vortex. *J. Fluid Mech.*, 285:69–94, 1995.

[85] G. K. Batchelor. Axial flow in trailing line vortices. *J. Fluid Mech.*, 20(04):645–658, 1964.

[86] H. Tryggeson. *Analytical Vortex Solutions to the Navier-Stokes Equation*. PhD thesis, Vaxjo University, 2007.

[87] C. J. Heaton, J. W. Nichols, and P. J. Schmid. Global linear stability of the non-parallel Batchelor vortex. *J. Fluid Mech.*, 629:139–160, 2009.

[88] V. Theofilis. Progress in flow instability analysis and laminar-turbulent transition modeling. *Von Karman Institute for Fluid Dynamics, Lecture Series 2014-07, June 2-6 2014.*

[89] F. Laporte and A. Corjon. Direct numerical simulations of the elliptic instability of a vortex pair. *Phys. Fluids*, 12(5):1016–1031, 2000.

[90] S. Le Dizès and F. Laporte. Theoretical predictions for the elliptical instability in a two-vortex flow. *J. Fluid Mech.*, 471:169–201, 2002.

[91] P. Meunier, S. Le Dizès, and T. Leweke. Physics of vortex merging. *Comptes Rendus Physique*, 6(4):431–450, 2005.

[92] P. Orlandi. Two-dimensional and three-dimensional direct numerical simulation of co-rotating vortices. *Phys. Fluids*, 19(1):3101, 2007.

[93] L. Nybelen. *Etude numérique d'écoulements tourbillonnaires de sillage d'avion.* PhD thesis, Université Paul Sabatier-Toulouse III, 2008.

[94] L. Nybelen and R. Paoli. Direct and large-eddy simulations of merging in corotating vortex system. *AIAA Journal*, 47(1):157–167, 2009.

[95] C. Roy, N. Schaeffer, S. Le Dizès, and M. Thompson. Stability of a pair of co-rotating vortices with axial flow. *Phys. Fluids*, 20(9):094101, 2008.

[96] X. Mao, S. J. Sherwin, and H. M. Blackburn. Non-normal dynamics of time-evolving co-rotating vortex pairs. *J. Fluid Mech.*, 701:430–459, 2012.

[97] M. S. Broadhurst, V. Theofilis, and S. J. Sherwin. Spectral element stability analysis of vortical flows. In *IUTAM Symposium on Laminar-Turbulent Transition*, pages 153–158. Springer, 2006.

[98] M. S. Broadhurst. *Vortex stability and breakdown: direct numerical simulation and stability analysis using biglobal and parabolised formulations*. PhD thesis, Imperial College London, 2007.

[99] H. Deniau and L. Nybelen. Strategy for spatial simulation of co-rotating vortices. *Int. J. Numer. Meth. Fl.*, 61(1):23–56, 2009.

[100] J. F. Hoessler. *Direct numerical analysis and stability analysis of vortex dominated flows around complex geometries*. PhD thesis, Imperial College London, 2011.