Scaling up of global stability tools in order tocharacterize phenomena specific to turbomachinery

Valentin Fer

École doctorale n° 432 : Sciences des métiers de l'ingénieur

Doctorat ENSAM

THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure des Arts et Métiers

Spécialité doctorale «Mécanique des fluides»

présentée et soutenue publiquement par

Valentin Fer

le 23/01/2023

Scale up of efficient global stability tools in order to characterize specific turbomachineries phenomenons

Directeur de thèse : **Jean-Christophe Robinet**

Jury

M. Lutz Lesshafft,	Directeur de recherche, LadHyX	Président
Mme. Taraneh Sayadi,	Chargée de recherche, Sorbonne Université	Rapporteuse
M. Nicolas Gourdain,	Professeur, ISAE-Supaero	Rapporteur
M. François Gallaire,	Professeur, EPFL	Examinateur
M. Grégory Dergham,	Docteur, Safran Tech	Examinateur
M. Jean-Christophe Robinet,	Professeur, DynFluid	Examinateur
M. Cédric Content,	Docteur, ONERA Châtillon	Examinateur
M. Sébastien Bourasseau,	Docteur, ONERA Châtillon	Examinateur
M. Samir Beneddine,	Docteur, ONERA Meudon	Invité
M. Denis Sipp,	Directeur de recherche, ONERA Meudon	Invité

Arts et Métiers - Centre de Paris
DynFluid - Laboratoire de Dynamique des Fluides

THESE

Contents

Chapter I

Introduction

Contents

I.1 Rotating stall phenomenon

The prediction of unsteady aerodynamic phenomena specific to turbomachinery is a key issue for aircraft manufacturers. In axial compressors, under the effect of a three-dimensional disturbance, a pocket of fluid with a lower rotational speed than that of the machine might stall on one or more blades. The stalls may coalesce and modify the angle of incidence seen by the upstream fluid, which is then deflected, giving rise to aerodynamic blockages that result in significant flow rate drops and consequent energy losses. The flow becomes unstable and the phenomenon, known as

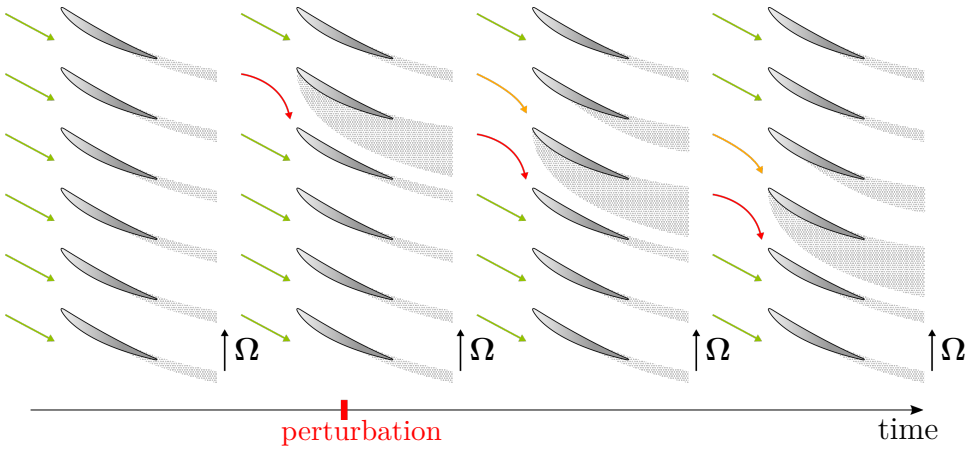


Figure I.1: Illustrative representation of the rotating stall phenomenon occuring in a compressor that rotates at speed $\mathbf{\Omega}$.

Figure I.2: Compressor stall vizualization on a Sukhoi Su-57 ptototype, by Rulexip *via* Wikipedia Commons.

rotating stall, occurs (schematically represented in fig. I.1). In severe cases, the fluid may not have a high enough flow rate to counteract adverse pressure gradients, causing the flow to oscillate at amplitudes that can be destructive for structures. These intense loads result in compressor stall, which can even damage the turbomachinery (as shown in fig. I.2).

Understanding these mechanisms is crucial to expand operating conditions and ensure system safety. The causes of these phenomena have been extensively investigated over the last century [1]. Greitzer provided in [2], a theoretical criterion based on the mass flow coefficient for which when the value exceeds a critical level, the system begins to pulse at frequencies characteristic of the compressor surge. This criterion was validated using experimental data in [3]. Courtiade & Gourdain [4] conducted experiments on rotating stall on the compressor CREATE (for *Compresseur de Recherche pour l'Étude des effets Aérodynamiques et Technologiques*) and established a stability limit. They observed that the entrance into surge regime caused the appearance of twelve stall cells successively shrinking and growing, spinning at 85% of compressor speed. When loads increased beyond the critical level, this pseudo-stable state degenerated into a full-span cell, triggering the surge. Other studies of the low-subsonic axial compressor CME2 [5] were carried out as part of the European ACONITE project [6], which aims to improve the stable operating range of axial compressor and reduce the surge margin.

Numerical investigations have also been documented. Hamilton [7] shows that a two-dimensional simulation of unsteady incompressible Reynolds-averaged Navier-Stokes (RANS) equations can effectively model the rotating stall on a portion of an isolated rotor. The code simulated the sudden appearance of a stall cell on a 15 rotor channels configuration. Other quasi-three-dimensional approaches, like that of He [8], explores the rotor-stator interaction with a time-marching Navier-Stokes (NS) method and a mixing-length turbulence model. Various configurations were tested and the study shows that the rotating stall is strongly influenced by the stator. It also demonstrates that the introduction of a perturbation upstream from the compressor plays a role in the instability process. In [9], Gourdain presents quasi-3-D simulations of the CME2 compressor. The computations were carried out over 60 revolution periods on a 6.5×10^5 points mesh. By establishing a "compressor characteristic", he predicts the limit cycle of the rotating stall and shows that the basic mechanism is intrinsically two-dimensional. However, these methods face difficulties in comparison with experimental data. To improve the accuracy of computational experiments and take into account three-dimensional effects and losses, three-dimensional simulations must be performed.

Analyzing the onset of rotating stall on a rotor/stator configuration, Marty et al. showed in [10]

that a 3-D RANS simulation on a single blade channel is sufficient to predict the stability margin. However, to capture the phenomenon, unsteady RANS simulations in a computational domain that matches the engine periodicity (at least 1/10 for the CME2 compressor, resulting in 3 rotor channels and 4 stator channels) must be performed. This agrees with the calculations by Gourdain [9] on the same configuration. However, above the stability threshold, such reduced computational domains don't accurately predict the flow behavior in the compressor. To better understand the phenomenon and have a simulation that is more representative above the instability onset, 3-D simulations are mandatory. Gourdain conducted such calculations in [9] on a 31×10^6 points computational domain and observed the complete coalescence of every stall in about 13 machine rotations. These calculations were carried out on a vectorial machine in approximately 900 hours. Marty et al. [10] performed simulations on a full-annulus rotor-stator configuration to investigate the control of the phenomenon. The computational domain consisted of 110×10^6 points, and the complete coalescence was observed in 10 machine rotations, which took approximately 390 hours on ONERA's clusters. At the price of high restitution times for design phases, these studies improved the understanding of the phenomenon, but its precise origin remains unknown. To reduce the level of modeling induced by RANS computations, LES (for Large Eddy Simulation) simulations can also be used [11]. However, these methods are about 500 times more expensive than RANS computation, and therefore too costful for industrial use.

Alternatively, the emergence of unsteady phenomena is often linked to the formation of at least one unstable mode in the flow. The exact source of these temporally self-sustained oscillations can be understood using linear stability theory. Applying global stability analysis to the flow in the CME2 compressor may help identify precursors to the rotating stall phenomenon. The various concepts of global stability theory are described below.

I.2 Linear stability theory

Linear stability theory focuses on the stability of solutions of differential equation-governed dynamic systems under the influence of small perturbations. A specific branch of stability theory is the study of the stability of fixed points, which are steady solutions of the governing equations in an equilibrium state. These states are considered stable if they return to equilibrium after being perturbed, as with a damped pendulum that returns to its original position after being moved away from its equilibrium state, or a ball that returns to its starting point in a well (see fig. I.3a). If the perturbation results in a motion with large amplitudes, the system is considered unstable, like a ball on the top of a hill (see fig. I.3b). The equilibrium may also be in a neutral state (see fig.

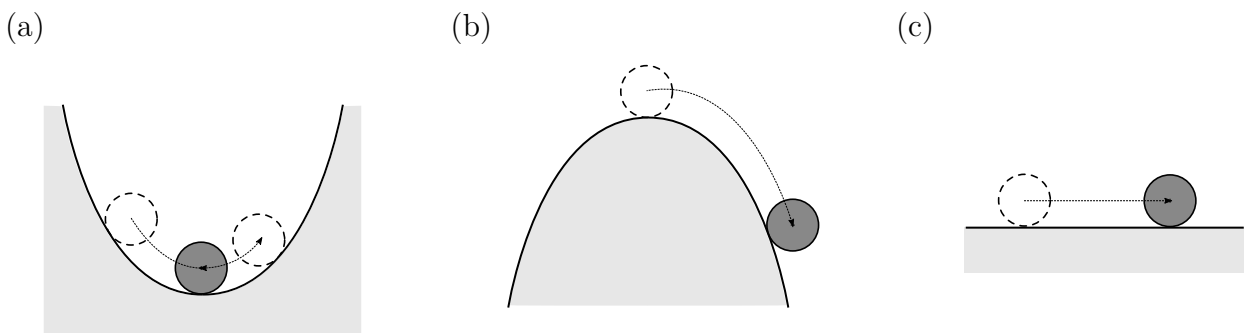


Figure I.3: Illustration of stable (a), unstable (b) and neutral (c) equilibrium.

(a) (b)

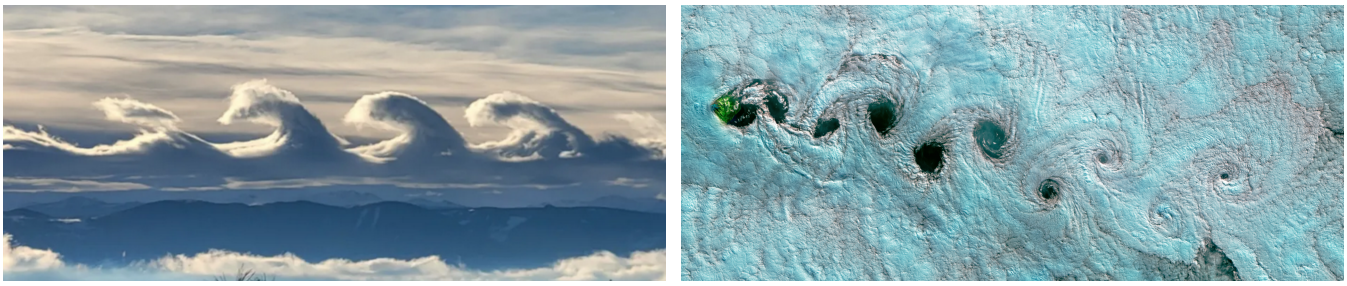


Figure I.4: (a) Kelvin-Helmholtz cloud waves over the Bighorn Mountains in Wyoming, by R. Gordon. (b) NASA satellite observation of a von Kàrmàn vortex street on the lee side of Tristan da Cunha volcano, by J. Stevens & J. Allen.

I.3c), in which a perturbation would move the ball away from its starting position with dampened motion.

Linear stability theory thus examines the behavior of small perturbations in a dynamic system. In fluid dynamics, the stability of equilibrium states have been studied for over a century. For instance, Kelvin and Helmholtz [12, 13] studied an hydrodynamic instability arising when two fluids are superimposed and one has a velocity above a critical level, giving rise to the so-called Kelvin-Helmholtz instability shown in fig. I.4a. Another example studied by Bénard and von Kàrmàn [14, 15] is the flow past a body that, when reaching a critical Reynolds number [16], triggers a vortex shedding, known as the von Kàrmàn vortex street, shown in fig. I.4b. Initially, instability was associated with parallel shear flows or axisymmetric jet for instance with the work of Lin [17], Batchelor & Gill [18] or Drazin & Reid [19]. In the 1990s, the stability theory about parallel flows has been extended with nonmodal perturbation development by Schmid & Henningson in [20]. All these studies were conducted on the basis of local stability analysis that lies on the hypothesis of parallel or weakly nonparallel flows. For more complex configurations, such assumption is seldom valid and the so-called global stability analysis is required. Huerre & Monkewitz, in [21], made the distinction between convectively unstable flow, i.e. flows that are spatially unstable and convected out of the domain; and absolutely unstable flows, i.e. flows which grow unsteadily all over the domain and are amplified, without getting convected out (by analogy with the rotating stall and surge presented lately). Absolutely unstable flows are a type of unstable global mode.

Direct global stability analysis. Let us consider a fluid governed by the so-called Navier-Stokes (NS) equations, which is a nonlinear system of PDE (for Partial Differential Equations). Considering a state vector $\boldsymbol{w}$ containing the conservative variables of the flow, its evolution can be described by the following conservative equations:

$$\frac{\partial \boldsymbol{w}}{\partial t} = \boldsymbol{R}(\boldsymbol{w}), \tag{I.1}$$

where $\boldsymbol{R}$ is the NS operator. The linear stability analysis aims to study the temporal evolution of a small perturbation around a base flow, which is a steady solution $\boldsymbol{w}_b$ of the system such that $\boldsymbol{R}(\boldsymbol{w}_b) = 0$. This perturbation is governed by a linear equation, expressed as:

$$\frac{\partial \boldsymbol{w}'}{\partial t} = \mathcal{J}(\boldsymbol{w}_b)\boldsymbol{w}', \tag{I.2}$$

where $\mathcal{J}(\boldsymbol{w}_b) = \partial \boldsymbol{R}/\partial \boldsymbol{w}|_{\boldsymbol{w}_b}$ is the linearized NS operator, otherwise known as the Jacobian of the flow. In global stability analysis, solutions are supposed to grow exponentially in time, under the form $\boldsymbol{w'} = \widehat{\boldsymbol{w}}(x, y, z, t)e^{\lambda t}$, resulting in the so-called direct eigenvalue problem:

$$\mathcal{J}(\boldsymbol{w}_b)\widehat{\boldsymbol{w}} = \lambda \widehat{\boldsymbol{w}}. \tag{I.3}$$

The stability of the k-th direct mode $\widehat{\boldsymbol{w}}_k$ is described by its corresponding eigenvalue λ_k. The latter is composed of an amplification rate $\sigma_k = \Re(\lambda_k)$ and an angular frequency $\omega_k = \Im(\lambda_k)$, while the eigenfunction $\widehat{\boldsymbol{w}}_k(x, y, z)$ provides the spatial structure of the mode. If at least one eigenmode has a positive amplification rate, the base flow is said to be asymptotically unstable. Global modes are poorly sensitive to external perturbations and describe the asymptotical behaviour of the flow. They give rise to self-sustained oscillations, making the flow act as an oscillator [22]. On the other hand, the base flow is asymptotically stable when only stable modes are present. Additionally, if the Jacobian is normal (i.e. $\mathcal{J}\mathcal{J}^\dagger = \mathcal{J}^\dagger\mathcal{J}$, where $\mathcal{J}^\dagger$ is the adjoint of $\mathcal{J}$ as defined below), its eigenvectors define an orthogonal modal basis. This means that expanding the perturbation $\boldsymbol{w'}$ onto such a basis composed only of stable modes will result in a monotonically decreasing amplitude of the perturbation. However, when the Jacobian is non-normal (i.e. does not commute with its adjoint), the modal basis is no longer orthonormal and energy growths may occur in the flow even in the absence of unstable modes. This phenomenon is known as short-term instability and can be studied using resolvent analysis, as described later.

With the improvement of computational resources, the global stability analysis has been increasingly successful over the past decades [23]. The first documented investigation was made by Pierrehumbert and Widnall in [24], studying a spatially periodic shear layer. Theofilis reviewed, in [25], the various types of global stability analysis and some methods to address the problems. When a 3-D domain is considered with two inhomogeneous directions, the analysis is referred to as "biglobal", while it is referred to as "triglobal" if each spatial direction is inhomogeneous. The first reported study of 3-D global stability analysis was carried out on the flow around a spheroid by Tezuka and Suzuki [26]. Paladini [27] also showed that this technique successfully predicted the appearance of the buffet phenomenon. In the context of turbomachinery, Xie et al. [28] recently used the global stability analysis method to predict the stall inception of the transonic axial compressor NASA Rotor 37.

Adjoint global stability analysis. It is also of interest to study the spectral components of the adjoint operator $\mathcal{J}^\dagger$ in global stability analysis. It is defined relatively to a given scalar product $\langle \cdot, \cdot \rangle$, as $\forall \boldsymbol{a}, \boldsymbol{b}$:

$$\langle \boldsymbol{a}, \mathcal{J}\boldsymbol{b} \rangle = \langle \mathcal{J}^\dagger \boldsymbol{a}, \boldsymbol{b} \rangle. \tag{I.4}$$

Eigenpairs of $\mathcal{J}^\dagger$ are found by solving the adjoint eigenproblem:

$$\mathcal{J}^\dagger(\boldsymbol{w}_b)\widehat{\boldsymbol{w}}^\dagger = \lambda^\dagger \widehat{\boldsymbol{w}}^\dagger. \tag{I.5}$$

Each direct eigenpair $(\lambda_k, \widehat{\boldsymbol{w}}_k)$ has a corresponding adjoint eigenpair $(\lambda_k^\dagger, \widehat{\boldsymbol{w}}_k^\dagger)$ with $\lambda_k^\dagger = \overline{\lambda}_k$ and $\widehat{\boldsymbol{w}}_k^\dagger$ the adjoint mode's spatial structure. In fluid mechanics, $\mathcal{J}$ is frequently non-normal, making it difficult to expand the perturbation over the nonorthogonal direct modal basis. The difficulty is overcome by normalizing each adjoint global mode such that $\langle \widehat{\boldsymbol{w}}_k, \widehat{\boldsymbol{w}}_k^\dagger \rangle = 1$. The direct and adjoint modal basis $(\widehat{\boldsymbol{w}}_k, k \geq 1)$ and $(\widehat{\boldsymbol{w}}_k^\dagger, k \geq 1)$ form a bi-orthogonal basis verifying $\langle \widehat{\boldsymbol{w}}_k, \widehat{\boldsymbol{w}}_l^\dagger \rangle = \delta_{kl}$[1]. The

[1] δ_{kl} being the Kroenecker symbol: $\delta_{kl} = 1$ if $k = l$, and $\delta_{kl} = 0$ if $k \neq l$

perturbation $\boldsymbol{w}'$ can be uniquely decomposed in the dual basis as:

$$\boldsymbol{w}' = \sum_{k\geq 1} \langle \widehat{\boldsymbol{w}}_k^{\dagger}, \boldsymbol{w}' \rangle \widehat{\boldsymbol{w}}_k. \tag{I.6}$$

The adjoint mode provides the exact field to trigger the corresponding direct mode in the flow. Its spatial structure shows the flow regions most sensitive to external forcing or initial condition changes. The adjoint mode is useful for studying the receptivity of the direct mode and can be used to establish a control strategy to delay unsteadiness [29, 22].

On the other hand, the computation of both direct and adjoint global modes gives access to the so-called structural sensitivity introduced by Giannetti & Lucchini in [30]. It is defined by the equation:

$$\boldsymbol{S}(x_0, y_0, z_0) = \|\widehat{\boldsymbol{w}}(x_0, y_0, z_0)\|_2 \cdot \|\widehat{\boldsymbol{w}}^{\dagger}(x_0, y_0, z_0)\|_2 \tag{I.7}$$

and represents the spatial overlap of two given direct and adjoint modes. It enables the detection of the so-called wavemaker, which identifies the regions directly responsible for generating self-sustained oscillations. Citro has shown an example of adjoint modal analysis and structural sensitivity calculation on a 3-D case for laminar flow around a spherical body in [31].

Resolvent analysis. As mentioned earlier, in some cases, transient phenomena associated with short-term instabilities result in an increase of energy in the flow even in the absence of unstable mode [32]. This occurs in the planar Couette flow, which is globally stable [33] but transitions to a turbulent regime around $\mathrm{Re} = 350$ [34]. The understanding of such phenomena was first improved by Reddy et al. [35] through the study of the pseudo-spectrum of the Orr-Sommerfield operator (see [20] for further details); or by Trefethen et al. [36], who proposed studying the pseudospectra of the linearized problem. As briefly introduced in the paragraph on direct global modes, the explanation lies in the non-normal nature of the Jacobian operator.

When $\mathcal{J}$ is normal, its eigenmodes are orthogonal and the flow dynamics is entirely governed by the direct global modes. This means that an increase of energy in the system corresponds to the existence of at least one unstable mode. However, when $\mathcal{J}$ is non-normal, the direct modes are nonorthogonal and the spectrum of $\mathcal{J}$, which provides the asymptotic behavior of the system, is not sufficient to exhaustively study transient phenomena. Perturbations on a stable base flow may thus result in pseudo-resonances, as in the case of the Couette flow. Fig. I.5 gives a geometric interpretation of this particular type of transient instability. The perturbation $\boldsymbol{w}'$ is represented as a linear combination of two stable modes whose amplitudes are exponentially damped over time:

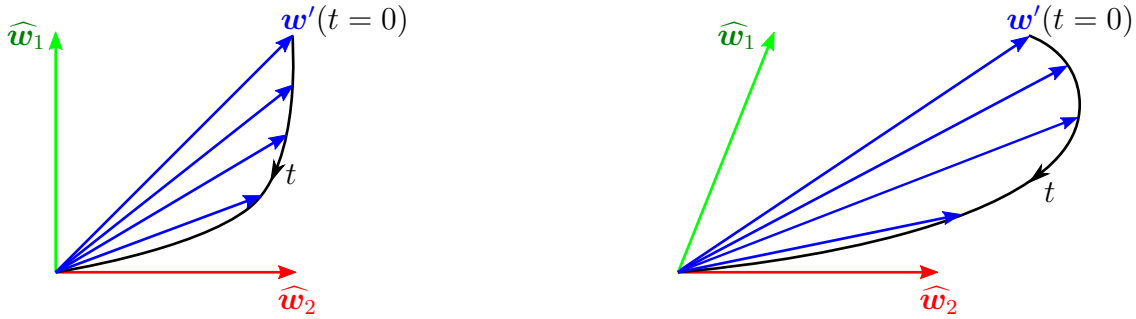


Figure I.5: Geometric illustration of the temporal evolution of $\|\boldsymbol{w}'\|$ in the normal (left) and non-normal (right) cases.

$\boldsymbol{w}' = \widehat{\boldsymbol{w}}_1 e^{\lambda_1 t} + \widehat{\boldsymbol{w}}_2 e^{\lambda_2 t}$, with $\sigma_1, \sigma_2 < 0$. When the modal basis is orthogonal, the amplitude of $\boldsymbol{w}'$, defined by $\|\boldsymbol{w}'\|_2 = \sqrt{\|\widehat{\boldsymbol{w}}_1\|^2 + \|\widehat{\boldsymbol{w}}_2\|^2}$, decreases monotonically in time. When the modal basis is nonorthogonal, the amplitude is defined by $\|\boldsymbol{w}'\|_2 = \sqrt{\|\widehat{\boldsymbol{w}}_1\|^2 + \|\widehat{\boldsymbol{w}}_2\|^2 + 2\langle \widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2 \rangle \cos(\widehat{\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2})}$, where $\widehat{\widehat{\boldsymbol{w}}_1, \widehat{\boldsymbol{w}}_2}$ is the angle between the two damped modes. In this case, depending on the level of non-normality of the global modes, an increase of energy may be observed before the amplitude of $\boldsymbol{w}'$ asymptotically decreases to zero. In this case, the flow is said to be a noise-amplifier.

The resolvent analysis, first introduced by Trefethen [36] in 1993, is suitable for studying short-term instabilities. Let us apply a forcing term $\boldsymbol{\phi}'$ to the linearized equations (I.1):

$$\frac{\partial \boldsymbol{w}'}{\partial t} = \mathcal{J}(\boldsymbol{w}_b)\boldsymbol{w}' + \boldsymbol{\phi}'. \tag{I.8}$$

Using the Fourier transforms $\boldsymbol{w}' = \widehat{\boldsymbol{w}}(x, y, z)e^{i\omega t}$ and $\boldsymbol{\phi}' = \widehat{\boldsymbol{\phi}}(x, y, z)e^{i\omega t}$, the relation between the harmonic response $\widehat{\boldsymbol{w}}$ and forcing $\widehat{\boldsymbol{\phi}}$ becomes:

$$\widehat{\boldsymbol{w}} = \mathcal{R}(\omega)\widehat{\boldsymbol{\phi}}, \tag{I.9}$$

where $\mathcal{R}(\omega) = (i\omega\mathcal{I} - \mathcal{J}(\boldsymbol{w}_b))^{-1}$ is the so-called resolvent operator, $\mathcal{I}$ the identity matrix and ω a given real frequency. The resolvent operator is a selective noise-amplifier that allows the identification of the optimal forcing of the flow, i.e. the forcing that maximizes the energy gain associated to $\mathcal{R}$. This gain is defined given a certain Hermitian norm $\|\cdot\|$ (usually related to kinetic or acoustic energy), as $\eta^2 = \|\widehat{\boldsymbol{w}}\|^2 / \|\widehat{\boldsymbol{\phi}}\|^2$. Finding the optimal forcing is thus equivalent to solve the following optimization problem:

$$\eta^2 = \max_{\widehat{\phi}} \frac{\langle \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{w}} \rangle}{\langle \widehat{\boldsymbol{\phi}}, \widehat{\boldsymbol{\phi}} \rangle}, \tag{I.10}$$

which in turns, is adressed computing the singular value decomposition of $\mathcal{R}$ [37, 22]. To do so, the following eigenvalue problem is solved:

$$\mathcal{R}^\dagger \mathcal{R} \widehat{\boldsymbol{\phi}} = \eta^2 \widehat{\boldsymbol{\phi}}, \tag{I.11}$$

where $\mathcal{R}^\dagger$ is the adjoint of $\mathcal{R}$. The operator $\mathcal{R}^\dagger\mathcal{R}$ is Hermitian, meaning its spectrum is real, and the set of optimal forcings $(\widehat{\boldsymbol{\phi}}_k, k \geq 1)$ forms an orthonormal basis over which the forcing $\widehat{\boldsymbol{\phi}}$ can be expanded $\widehat{\boldsymbol{\phi}} = \sum_{k\geq 1} \langle \widehat{\boldsymbol{\phi}}_k, \widehat{\boldsymbol{\phi}} \rangle \widehat{\boldsymbol{\phi}}_k$. In a similar way, it may be shown that the set of optimal responses $(\widehat{\boldsymbol{w}}_k, k \geq 1)$ forms an orthonormal basis and $\widehat{\boldsymbol{w}}$ can be uniquely decomposed as $\widehat{\boldsymbol{w}} = \sum_{k\geq 1} \langle \widehat{\boldsymbol{w}}_k, \widehat{\boldsymbol{w}} \rangle \widehat{\boldsymbol{w}}_k$. For a given eigenpair of $\mathcal{R}^\dagger\mathcal{R}$ composed of the k-th optimal forcing $\widehat{\boldsymbol{\phi}}_k$ and gain η_k^2, the k-th optimal response $\widehat{\boldsymbol{w}}_k$ is recovered by:

$$\widehat{\boldsymbol{w}}_k = \eta_k^{-1} \mathcal{R} \widehat{\boldsymbol{\phi}}_k. \tag{I.12}$$

Using both forcing and response spaces, equation (I.9) may be recast as:

$$\widehat{\boldsymbol{w}} = \sum_{k\geq 1} \eta_k \langle \widehat{\boldsymbol{\phi}_k}, \widehat{\boldsymbol{\phi}} \rangle \widehat{\boldsymbol{w}}_k \tag{I.13}$$

As stated by Bennedine in [38], when a strong convective instability is present in the flow, the optimal gain η_1 is such that $\eta_1 \gg \eta_{k\geq 2}$, meaning the structure of $\widehat{\boldsymbol{\phi}}$ is close to $\widehat{\boldsymbol{\phi}}_1$, which yields a response $\widehat{\boldsymbol{w}}$ close to $\widehat{\boldsymbol{w}}_1$.

In a nutshell, the resolvent analysis improves the understanding of the genesis of nonmodal and nonlinear phenomenon within a linear framework. Such analysis has been used to describe the pseudo-resonance of Blasius boundary layer [39], incompressible jets [40], flow around OAT15A airfoil [41], or to make separation control on a NACA0012 airfoil [42]. More recently, Lugrin [43] exploited the resolvent analysis to explain the amplification mechanism of oblique modes, which are linearly amplified convective instabilities appearing on hypersonic vehicles.

The tools of global stability may hence be useful to characterize the rotating stall phenomenon on axial compressors like the CME2. However, as depicted hereafter, there are few methods available for analyzing such large-scale configurations.

I.3 Methods for high dimensional eigenvalue problems

The global stability have been introduced within a continuous framework. In the context of Computational Fluid Dynamics (CFD), each operator and vectors presented earlier must be discretized. If the computational domain consists of N_c cells, and the problem involves d conservative variables, the number of degrees-of-freedom (DOF) of the problem is $N = N \times d$. The discretized state vectors are N-length vectors and the Jacobian operator is a N-by-N matrix. When N_c is high, the modal and nonmodal global stability problems are therefore based on the resolution of large eigenvalue problems. As only the least stable eigenvalues of the spectrum are of interest, it would be counterproductive to use direct methods like QR factorization [44] that returns the entire spectrum of the considered operator. Iterative eigensolvers such as Arnoldi [45] or Krylov-Schur (KS) [46] algorithms are preferred instead. Many methods related to the resolution of large eigenvalue problems have been described by Saad in [47].

The counterpart of Arnoldi-like eigensolvers is that they are best at finding the eigenvalues of the largest magnitude of the considered operators. As discussed by Theofilis in [25], the operators of modal analysis are preferentially modified with spectral transformations like shift-and-invert as to solve the eigenvalue problem. The latter, first used in the context of global stability analysis by Christodoulou & Scriven [48], involves selecting a complex scalar (called shift) s, and solving the transformed problem:

$$\left(\mathcal{J} - s\mathcal{I}\right)^{-1} \widehat{\boldsymbol{w}} = \widetilde{\lambda}\widehat{\boldsymbol{w}}, \tag{I.14}$$

where $\widetilde{\lambda}$ is linked to the eigenvalue of the original problem by the relation $\widetilde{\lambda} = (\lambda - s)^{-1}$. This technique allows for the calculation of only a portion of the spectrum of interest, as the largest magnitude eigenvalues of the transformed operator are those closest to s. Thus, whether dealing with modal or resolvent analysis, inversions of large complex-valued operators must be carried out. For 2-D problems and small 3-D ones, the usual strategy is to use a direct method like LU decomposition [49]. This can be achieved using the efficient parallel MUMPS solver (see `https://mumps-solver.org/`) may be used.

As stated by L'Excellent in [50], if $\mathrm{NE}(\mathcal{J})$ is the number of entries of the Jacobian, the memory cost of its LU factorization is of the order of $(\mathrm{NE}(\mathcal{J})/7)^{4/3}$. When dealing with complex configurations exceeding the 10×10^6 DOF, such as the full-annulus compressor CME2, storage of LU factorization thus becomes almost impossible due to memory limitations and inversions must be managed with iterative methods instead. As pointed out by Guilbert [51], due to the non-symmetrical, non-normal and not necessarily diagonal dominant nature of the operators encountered in compressible CFD, the most efficient current methods are preconditioned Krylov ones. Saad describes such methods in details in [52]. For example, Mack & Schmid [53] used an ILU-based preconditioned Krylov technique, the BICGSTAB, to conduct global stability analysis

about a swept parabolic body. Timme & Thormann discussed in [54] the efficiency of various preconditioned Krylov methods for solving global stability problems. Timme also carried out the largest known direct and adjoint global stability analysis in [55] to characterize the buffet onset of a full airplane configuration, resulting in approximately 50×10^6 mesh points for the finest grid studied. To the author's knowledge, no resolvent analysis has been performed on such large cases.

I.4 Positioning of the present work

Based on the previous sections, the understanding of the origin of rotating stall can be improved through the use of global stability analysis. However, due to the size of the problem, traditional direct factorization methods cannot be utilized, and the entire global stability process must be adapted.

At ONERA, work is underway to improve the high performance computing (HPC) layers of the compressible elsA CFD code [56, 57]. This includes cache-blocking and vectorization optimizations. Guilbert [51] has also developed preconditioning tools to efficiently solve the large real-valued sparse linear systems involved in computing fixed points in compressible CFD using the GMRES algorithm [58]. This is achieved by computing the Jacobian matrix with Algorithmic Differentiation (AD), which preserves the HPC layer, and using preconditioners based on ILU(k) [52] and local-LU factorization. These developments could potentially be adapted for solving linear systems related to global stability problems.

The main obstacle to this is that the linear systems involved in global stability analysis are complex-valued, which is incompatible with vectorization as complex arithmetic is not vectorizable. To take advantage of both Guilbert's work and HPC optimizations, the complex operators need to be reformulated in a real-equivalent version, which should be propagated throughout the entire eigenvalue computation process.

I.5 Objectives and organization of the thesis

The objective of this work is to develop a method in elsA software to scale up the overall global stability process to industrial configurations with more than 10×10^6 DOF like the compressor CME2. To achieve this, the manuscript is divided into three chapters. The first chapter starts with a presentation of numerical methods used to discretize governing equations of compressible CFD. Then, efficient fixed point computations, a pre-requisite for global stability analysis, are discussed using Newton-like methods. The concept of AD for Jacobian computation is also covered, followed by an overview of various tools for global stability analysis in a discrete framework. Finally, the efficient iterative eigensolver KS algorithm [46], in its complex version, is presented. Each concept is illustrated using the viscous Burgers case [59], a 1-D nonlinear advection equation that resembles the Navier-Stokes equations.

In chapter II, after establishing the mathematical background for laminar and RANS computations, the fixed point methods as implemented in elsA are discussed. The focus is on the implicit backward Euler scheme solved with the approximate LU-SGS method and the exact GMRES-based [58] method. A good preconditioner, based on the exact Jacobian matrix, is required to ensure convergence of the latter method. The Jacobian is computed using multicoloring [51], while the preconditioning is based on ILU(k) and hybrid local-LU methods. A KS algorithm compatible with HPC optimizations is also developed. As mentioned earlier, the operators of global stabil-

ity analysis have to be reformulated in a real-equivalent version for effective vectorization. To do so, the K formulation, introduced by Day & Heroux in [60] is selected, which conserves the sparsity pattern of the original complex operator and thus allows the reuse of the preconditioning routines developed by Guilbert [51]. This formulation is first implemented to be operative with the GMRES algorithm. The spectral properties of the K-formulated operators are then described, and detailed in the case of modal and resolvent analysis. Implementations of the KS algorithm applied to global stability in elsA are provided, giving rise to a new method named K-formulated global stability analysis method (KFGSM). The method is further validated on the canonical case of laminar cylinder flow for direct, adjoint, and resolvent computations.

In the last chapter, the KFGSM is tested on more challenging configurations. First, the RANS case of OAT15A airfoil flow is addressed, and the method is validated for direct, adjoint, and resolvent computations. The KFGSM is then tested on the subsonic laminar sphere flow, a 3-D configuration that reaches 10×10^6 DOF, representing industrial configurations for which the KFGSM was designed. The method is used to thoroughly characterize the two bifurcations of this case using direct, adjoint, and resolvent analysis. The code is even tested on a refined mesh with about 125×10^6 DOF for demonstration purposes.

Chapter II

Overview of global stability analysis tools and their implementation

Contents

II.1 Introduction

The present chapter builds a mathematical background to introduce the various tools of global stability analysis. The purpose is to describe each step of the process in a general way, in order to set these ideas before the K-formulated global stability method (KFGSM), as implemented in elsA [61] software is detailed in chapter II. To this end, general considerations about conservation laws as well as numerical discretization, in the context of compressible Computational Fluid Dynamics (CFD) are first addressed. Secondly, some fixed point computation methods with the strategy

developed to get linearized operators are detailed. Finally, the main tools of global stability analysis are discussed, and the Krylov-Schur (KS) [46] algorithm with complex arithmetics is introduced.

II.2 Governing equations in compressible CFD

The general conservation equations arising in fluid mechanics and their discretization within the framework of compressible CFD are presented here. Unlike the incompressible version of Navier-Stokes (NS) equations that can be elliptic or parabolic, their compressible counterpart is hyperbolic [62]. In the latter description, the convective part must be addressed using Riemann solvers [63]. This section presents the general concepts of spatial discretizations, particularly for compressible CFD solvers. These are illustrated on the one-dimensional Burgers equation [59], which mimics the NS equations. Mathematical formulations of the CFD problem treated with the elsA solver are presented in III.3.

II.2.1 Conservation laws

Let us consider the state vector $\boldsymbol{w}(\boldsymbol{M},t) \in \mathbb{R}^d$ containing d conservative variables, where $\boldsymbol{M} \in \mathbb{R}^{N_{\text{dim}}}$ is the N_{dim}-dimensional vector representing the spatial position and $t \in \mathbb{R}^+$ is the time parameter. The transformation $\boldsymbol{p}$ is defined in the following such that the primitive variables are contained in the vector $\boldsymbol{p}(\boldsymbol{w}) \in \mathbb{R}^d$. For the sake of conciseness, the notation $\boldsymbol{p}$ will be used to denote $\boldsymbol{p}(\boldsymbol{w})$ in the following. From Newton's second law, the evolution of such vector is governed by the following system of partial differential equations (PDE):

$$\frac{\partial \boldsymbol{w}}{\partial t} + \nabla \boldsymbol{f} = \boldsymbol{t}_s, \tag{II.1}$$

where $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{p}, \nabla \boldsymbol{p}) \in \mathbb{R}^d \times \mathbb{R}^{N_{\text{dim}}}$ is the physical flux based on the primitive variables and $\boldsymbol{t}_s = \boldsymbol{t}_s(\boldsymbol{M},t) \in \mathbb{R}^d$ represents source terms ($\boldsymbol{t}_s = 0$ leads to the homogeneous case). The conservation law (II.1) is only valid for sufficiently smooth solutions. When discontinuities like shock waves occur, the flow solution becomes discontinuous and the previous relation must be integrated over a control volume Ω to get the integral form of the conservative equations:

$$\int_\Omega \frac{\partial \boldsymbol{w}}{\partial t} \mathrm{d}\Omega + \int_\Omega \nabla \boldsymbol{f} \mathrm{d}\Omega = \int_\Omega \boldsymbol{t}_s \mathrm{d}\Omega. \tag{II.2}$$

Considering the fluxes function to be of class C^1, the time derivative can be decoupled from the space integral, and the divergence theorem can be applied to get:

$$\frac{\partial}{\partial t} \int_\Omega \boldsymbol{w} \mathrm{d}\Omega + \oint_{\partial\Omega} \boldsymbol{f} \cdot \boldsymbol{n} \mathrm{d}S - \int_\Omega \boldsymbol{t}_s \mathrm{d}\Omega = 0, \tag{II.3}$$

with $\boldsymbol{n}$ the unit external normal defined on the frontier $\partial\Omega$ of the domain. Physically, equation (II.3) links the temporal variation of $\boldsymbol{w}$ to its flux crossing the frontier $\partial\Omega$ and its production or destruction *via* the source term. Additionally, the flux term can be seen as the sum of two contributions: the convective part $\boldsymbol{f}_c$, and the diffusive part $\boldsymbol{f}_d$. The system can be recast as:

$$\frac{\partial}{\partial t} \int_\Omega \boldsymbol{w} \mathrm{d}\Omega + \oint_{\partial\Omega} (\boldsymbol{f}_c - \boldsymbol{f}_d) \cdot \boldsymbol{n} \mathrm{d}S - \int_\Omega \boldsymbol{t}_s \mathrm{d}\Omega = 0. \tag{II.4}$$

As a reminder, PDE can be classified as hyberbolic, parabolic or elliptic depending on their properties. Hirsch presents the physical meaning of each of these classifications in [62]. A system of equations is said to be hyperbolic if its homogeneous part admits wave-like solutions. The physics of such system is dominated by convection phenomena. For instance, the so-called inviscid 1-D Burgers equation which describes the convection without diffusion of a scalar quantity u is purely hyperbolic:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0. \tag{II.5}$$

When the wave-like solution is damped in time, the equation is said to be parabolic. Such property is associated with time-dependent diffusion. A canonical case is the parabolic heat equation that governs the evolution of the temperature T in a fluid at rest of thermal coefficient κ:

$$\frac{\partial T}{\partial t} - \kappa \Delta T = 0. \tag{II.6}$$

If the system does not admit any wave-like solution, it is said to be elliptic which is associated to pure spatial diffusion. Such behaviour describes the spatial diffusion of the pressure p in an incompressible 2-D flow:

$$\Delta p + \rho \left(\left(\frac{\partial u_x}{\partial x} \right)^2 + 2 \frac{\partial u_x}{\partial y} \frac{\partial u_y}{\partial x} + \left(\frac{\partial u_y}{\partial y} \right)^2 \right) = 0. \tag{II.7}$$

The nature of the conservation laws varies depending on the physical system of interest. In relation (II.4), the convective flux is an hyperbolic operator while the diffusive flux is parabolic. The system is purely hyperbolic when the latter is zero, as in the stationary Euler equations. But in the case of the NS equations, the presence of $\boldsymbol{f}_d$ induces a hybrid mix hyperbolic-parabolic describing a convection-diffusion phenomenon. This behaviour can be reproduced by adding a viscosity parameter ν to the Burgers equation (II.5):

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} - \nu \frac{\partial u}{\partial x} \right) = 0. \tag{II.8}$$

In this case, $f_c = u^2/2$ and $f_d = -\nu \partial u / \partial x$. For the sake of better comprehension, the latter equation will be used to illustrate the various concepts presented in this chapter.

II.2.2 Spatial discretization and Riemann solvers

The following presents the spatial discretization by means of the finite volumes method. A solution to the problem of interest is searched on the domain Ω, divided into N_c cells. The integral conservation law (II.3) applied in a cell Ω_k at a time t writes:

$$\frac{\partial}{\partial t} \int_{\Omega_k} \boldsymbol{w} \mathrm{d}\Omega + \oint_{\partial \Omega_k} \boldsymbol{f} \cdot \boldsymbol{n} \mathrm{d}S - \int_{\Omega_k} \boldsymbol{t}_s \mathrm{d}\Omega = 0. \tag{II.9}$$

If Ω_k contains N_int interfaces where the j-th one is denoted by I_j, this relation becomes:

$$\frac{\partial}{\partial t} \int_{\Omega_k} \boldsymbol{w} \mathrm{d}\Omega + \sum_{j=1}^{N_\mathrm{int}} \oint_{I_j} \boldsymbol{f} \cdot \boldsymbol{n} \mathrm{d}S - \int_{\Omega_k} \boldsymbol{t}_s \mathrm{d}\Omega = 0. \tag{II.10}$$

We consider in addition the average value $\boldsymbol{w}_k$ of $\boldsymbol{w}$ in cell Ω_k:

$$\boldsymbol{w}_k = \frac{1}{V(\Omega_k)} \int_{\Omega_k} \boldsymbol{w} \mathrm{d}\Omega, \tag{II.11}$$

where $V(\Omega_k)$ stands for the volume of Ω_k. The general formulation of the finite volume method is thus given by:

$$\frac{\partial (V(\Omega_k)\boldsymbol{w}_k)}{\partial t} + \sum_{j=1}^{N_{\text{int}}} \left(\boldsymbol{h}_c(\boldsymbol{w}_{I_j}^L, \boldsymbol{w}_{I_j}^R) - \boldsymbol{h}_d(\boldsymbol{w}_{I_j}^L, \boldsymbol{w}_{I_j}^R) \right) S_j - V(\Omega_k)\boldsymbol{T}_k = 0, \tag{II.12}$$

where $\boldsymbol{h}_c$ and $\boldsymbol{h}_d$ are functions that compute convective part and diffusive part of the numerical approximations of the physical fluxes $\oint_{I_j} (\boldsymbol{F}_c - \boldsymbol{F}_d)\cdot\boldsymbol{n}\mathrm{d}S$ crossing each interface I_j of surface S_j. $\boldsymbol{w}_{I_j}^L$ and $\boldsymbol{w}_{I_j}^R$ are the left and right (from I_j) state of conservative variables. In compressible CFD, the equations are dominated by convective fluxes, which makes them mainly hyperbolic. Special care must be taken to build $\boldsymbol{h}_c$ to avoid oscillations across a discontinuity in the solution. The diffusive part $\boldsymbol{h}_d$ can be seen as a numerical dissipation that damps discontinuities. Hereafter, some types of numerical schemes used in this thesis are described in a general way. Hyperbolic convective and parabolic viscous operators are separately considered to build each flux. These schemes are illustrated for Burgers equation. An extension to 3-D configurations, such as implemented in elsA solver, is presented in III.3.

Convective fluxes. We focus in the following on the homogeneous hyperbolic part of the conservation law (II.1). The finite volume method uses the average value of $\boldsymbol{w}$ in each cell so the solution is a constant piecewise function, which creates discontinuities at each interface. Godunov [64] introduced a first-order upwind scheme to compute the intercell numerical flux $\boldsymbol{h}_{cI_j}$, requiring the resolution of a nonlinear hyperbolic local Riemann problem [63]. In 1-D (i.e. $N_{\text{dim}} = 1$), this problem writes:

$$\begin{cases} \dfrac{\partial \boldsymbol{w}}{\partial t} + \mathcal{A}_c \dfrac{\partial \boldsymbol{w}}{\partial x} = 0, \\ \boldsymbol{w}(x,0) = \begin{cases} \boldsymbol{w}_L \text{ if } x < 0 \\ \boldsymbol{w}_R \text{ if } x > 0 \end{cases} \end{cases} \tag{II.13}$$

where $\mathcal{A}_c = \partial \boldsymbol{F_c}/\partial \boldsymbol{w} \in \mathbb{R}^d \times \mathbb{R}^d$ stands for the Jacobian matrix of the convective fluxes. The latter is diagonalizable with real eigenvalues, meaning there exist matrices $\mathcal{X}, \mathcal{D} \in \mathbb{R}^d \times \mathbb{R}^d$ verifying $\mathcal{A}_c = \mathcal{X}\mathcal{D}\mathcal{X}^{-1}$ and where $\mathcal{D}$ is a diagonal matrix containing the eigenvalues of $\mathcal{A}_c$ in its diagonal. The expression for the Godunov flux based on the center of the interface I_j then writes:

$$\boldsymbol{h}_{cI_j} = \frac{1}{2}\left(\boldsymbol{f}_c(\boldsymbol{p}_{I_j}^L) + \boldsymbol{f}_c(\boldsymbol{p}_{I_j}^R) \right) \cdot \boldsymbol{n}_{I_j} - \frac{1}{2}|\mathcal{A}_c| \left(\boldsymbol{w}_{I_j}^R - \boldsymbol{w}_{I_j}^L \right), \tag{II.14}$$

where $|\mathcal{A}_c| = \mathcal{X}|\mathcal{D}|\mathcal{X}^{-1}$, and $|\mathcal{D}|$ is the diagonal matrix containing the modulus of each eigenvalue. Such a discretization conserves the local information of wave propagation through the entire domain, and physical discontinuities like shock waves can be captured without generating additional spurious oscillations. The counterpart is the need to solve as many Riemann problems as there are interfaces in the domain. This strategy is time-consuming as it requires $\mathcal{A}_c$ at each interface.

Instead of looking for the exact solution of problem (II.13), another possibility is to search for approximate solutions by solving approximated Riemann problems. This idea is behind the Roe scheme [65] widely used in this work. The concept is to replace the Riemann problem with an

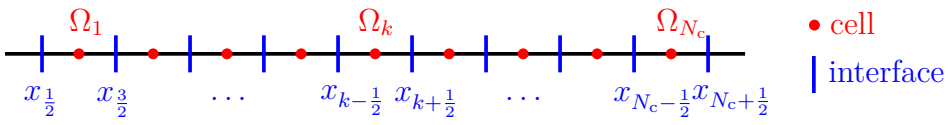


Figure II.1: Segment discretization.

approximated linearized one, which is achieved by replacing $\mathcal{A}_c$ in (II.13) by a so-called Roe matrix $\widetilde{\mathcal{A}}_c$. The latter only depends on $\boldsymbol{w}^L$ and $\boldsymbol{w}^R$ and must satisfy the three following properties:

$$\begin{aligned} &\text{(i) } \widetilde{\mathcal{A}}_c(\boldsymbol{w}^L, \boldsymbol{w}^R) \text{ is diagonalisable with real eigenvalues;} \\ &\text{(ii) } \widetilde{\mathcal{A}}_c(\boldsymbol{w}^L, \boldsymbol{w}^R)(\boldsymbol{w}^R - \boldsymbol{w}^L) = \boldsymbol{f}_c(\boldsymbol{w}^R) - \boldsymbol{f}_c(\boldsymbol{w}^L); \\ &\text{(iii) } \widetilde{\mathcal{A}}_c(\boldsymbol{w}, \boldsymbol{w}) = \mathcal{A}(\boldsymbol{w}). \end{aligned} \tag{II.15}$$

By analogy with the Godunov flux, the Roe intercell flux is defined as:

$$\boldsymbol{h}_{cI_j} = \frac{1}{2}\left(\boldsymbol{f}_c(\boldsymbol{p}^R_{I_j}) + \boldsymbol{f}_c(\boldsymbol{p}^R_{I_j})\right) \cdot \boldsymbol{n}_{I_j} - \frac{1}{2}\left|\widetilde{A}_c\right|\left(\boldsymbol{w}^R_{I_j} - \boldsymbol{w}^L_{I_j}\right), \tag{II.16}$$

Let us pinpoint that a lot of schemes based on the formulation of various Riemann problems at the interfaces can be used, like the so-called HLLC scheme, and more details about their derivation can be found in [63].

Such a scheme ensures a first-order-accurate discretization in space. They can be extended to second-order accuracy using the Monotonic Upstream-centered Scheme for Conservation (MUSCL) reconstruction method introduced by Van Leer in [66]. The latter assumes that the left and right states are no longer constant but may vary linearly through the cells. Second-order accuracy is obtained using a reconstruction at the faces of left and right states. If the left and right cells of interface I_j respectively are Ω_k and Ω_l, then one may write:

$$\begin{cases} \boldsymbol{p}^L_{I_j} = \boldsymbol{p}_k + \nabla \boldsymbol{p}|_k \cdot \boldsymbol{d}^k_{I_j} \\ \boldsymbol{p}^R_{I_j} = \boldsymbol{p}_l + \nabla \boldsymbol{p}|_l \cdot \boldsymbol{d}^l_{I_j}, \end{cases} \tag{II.17}$$

where $\boldsymbol{d}^k_{I_j}$ is the vector going from the center of cell Ω_k to the center of interface I_j. The use of MUSCL reconstruction leads to spurious oscillations close to flow discontinuities. This can be countered by using limitor functions for the reconstructed values [62].

In the case of Burgers equation, the domain Ω is a segment $[x_1, x_{N_c}]$ in a cartesian coordinates system with a uniform space grid Δx, and each cell Ω_k has two interfaces in $x_{k-\frac{1}{2}}$ and $x_{k+\frac{1}{2}}$, as shown in fig. II.1. At second-order accuracy without limiting function, the Roe flux writes:

$$h_{ck+\frac{1}{2}} = \frac{1}{2}\left(f_c(u^L_{k+\frac{1}{2}}) + f_c(u^R_{k+\frac{1}{2}})\right) - \frac{1}{4}\left|u_k + u_{k-1}\right|(u^R_{k+\frac{1}{2}} - u^L_{k+\frac{1}{2}}), \tag{II.18}$$

with $u^L_{k+\frac{1}{2}}$ and $u^R_{k+\frac{1}{2}}$ defined as:

$$\begin{cases} u^L_{k+\frac{1}{2}} = u_k - \dfrac{\Delta x}{2}\nabla u|_k \\ u^R_{k+\frac{1}{2}} = u_{k+1} + \dfrac{\Delta x}{2}\nabla u|_k\,. \end{cases} \tag{II.19}$$

Diffusive fluxes. Diffusive fluxes are calculated by a discretization purely centered at interface I_j, based on the primitive variables:

$$\boldsymbol{h}_{dI_j} = \boldsymbol{f}_d\left(\boldsymbol{p}_{I_j}, \nabla\boldsymbol{p}|_{I_j}\right) \cdot \boldsymbol{n}_{I_j}. \tag{II.20}$$

The primitive variables at the center of I_j are estimated by arithmetic mean of their values at the center of each adjacent cells. Gradients are also computed at interfaces. Various methods can be applied, but the one used in elsA is a corrected version of the so-called five-points stencil ($5p$) method. It is described in III.3.2.

In the case of Burgers equation, the numerical diffusive flux writes:

$$h_{dk+\frac{1}{2}} = \nu\, \nabla u|_{k+\frac{1}{2}} \cdot \boldsymbol{n}_{k+\frac{1}{2}}. \tag{II.21}$$

The gradient term is obtained by a Taylor expansion of $u_{k+\frac{1}{2}}$. At second-order, it gives:

$$u_{k+\frac{1}{2}} = u_k + \frac{\Delta x}{2}\, \nabla u|_{k+\frac{1}{2}}\,. \tag{II.22}$$

From what precedes, the quantity u at the interface in $x_{k+\frac{1}{2}}$ is $u_{k+\frac{1}{2}} = (u_k + u_{k+1})/2$, from which one deduces the expression of the gradient at the face:

$$\nabla u|_{k+\frac{1}{2}} = \frac{1}{2}\,(u_{k+1} - u_k)\,. \tag{II.23}$$

Hence, the numerical diffusive flux reads:

$$h_{dk+\frac{1}{2}} = \frac{\nu}{\Delta x}\,(u_{k+1} - u_k)\,. \tag{II.24}$$

In a similar way, we get at fourth order:

$$h_{dk+\frac{1}{2}} = \frac{\nu}{24\Delta x}\,(-u_{k-1} + 26u_k - u_{k+1})\,. \tag{II.25}$$

Now that the finite volume method and flux discretization have been introduced in the context of compressible CFD, the next section presents methods to compute fixed points. In the following, the discrete representation of all vectors and scalars will be written in upper case. For example, the discretized state vector $\boldsymbol{W} \in \mathbb{R}^N$, with $N = d \times N_{\mathrm{c}}$, is composed of the N_{c} state vectors $\boldsymbol{w}$. We also introduce the discrete residual operator $\boldsymbol{R}(\boldsymbol{W}) \in \mathbb{R}^N$ whose k-th component reads:

$$\boldsymbol{t}_{sk} - \frac{1}{V(\Omega_k)} \sum_{j=1}^{N_{\mathrm{int}}} \boldsymbol{h}(\boldsymbol{w}_{I_j}^L, \boldsymbol{w}_{I_j}^R) S_j. \tag{II.26}$$

The general discretized governing equations can thus be recast under the following form:

$$\frac{\partial \boldsymbol{W}}{\partial t} - \boldsymbol{R}(\boldsymbol{W}) = 0. \tag{II.27}$$

II.3 Fixed point computation

As introduced in I.2, a global stability analysis studies the linear evolution of global harmonic perturbations about a baseflow $\boldsymbol{W}_b$ to determine its asymptotic stability. Therefore, being able to compute $\boldsymbol{W}_b$ accurately is a prerequisite. In this work, $\boldsymbol{W}_b$ is chosen to be a fixed point, i.e. a steady solution of equation (II.1). Note that it is also physically relevant to carry out a stability analysis on the mean flow instead of a fixed point (see for instance [38]). But this latter approach is out of the scope of the present thesis and will not be discussed.

Finding a steady-state solution of equation (II.27) results in the following equivalence:

$$\frac{\partial \boldsymbol{W}}{\partial t} = 0 \quad \Leftrightarrow \quad \boldsymbol{R}(\boldsymbol{W}) = 0. \tag{II.28}$$

Such a problem can be solved by Newton-like methods or a pseudo time-marching loop. However, from time integration, the use of explicit schemes such as the Runge-Kutta scheme [67] could be very expansive in terms of CPU usage. This is why only implicit Newton-like methods were used in this work.

Hereafter, the Newton method and the linearized first-order backward Euler method with local time-step are derived for the conservative equations. They are then illustrated on the resolution of the viscous Burgers problem defined by equation (II.8). We focus on its resolution on the segment $[-10, 10]$, with $u(-10, t) = 1$ and $u(10, t) = -1$, and where $N_c = 400$. A theoretical fixed point of such a boundary-valued problem is known to be $u_{\text{theo}} = -\tanh(x/2)$ [68]. Both convective and diffusive fluxes are discretized at second-order accuracy using respectively (II.18) and (II.24).

II.3.1 Newton method

Given an initial guess $\boldsymbol{W}^0$, the Newton method iteratively evaluates the gradient direction toward a fixed point. The n-th Newton iteration is based on the following linearization:

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{W}^{n+1}) &= \boldsymbol{R}(\boldsymbol{W}^n + \Delta \boldsymbol{W}) \\ &\simeq \boldsymbol{R}(\boldsymbol{W}^n) + \mathcal{J}(\boldsymbol{W}^n)\Delta \boldsymbol{W}, \end{aligned} \tag{II.29}$$

where $\mathcal{J}(\boldsymbol{W}^n) = \partial \boldsymbol{R}/\partial \boldsymbol{W}|_{\boldsymbol{W}_n}$ is the Jacobian matrix of the residual. Each iteration consists in evaluating $\Delta \boldsymbol{W}$ through the resolution of the following linear system:

$$\mathcal{J}\Delta \boldsymbol{W} = -\boldsymbol{R}(\boldsymbol{W}^n), \tag{II.30}$$

and the solution of the nonlinear problem at the next iteration is given by $\boldsymbol{W}^{n+1} = \boldsymbol{W}^n + \Delta \boldsymbol{W}$. The algorithm stops when $\boldsymbol{R}(\boldsymbol{W}^{n+1})$ becomes lower than some user-defined tolerance. Under certain assumptions, the convergence is expected to be quadratic. This method has been developed on the Burgers case. Given the differentiated right Jacobian-vector product obtained using the Algorithmic Differentiation (AD, presented in II.4), the Jacobian $\mathcal{J}$ is computed by applying this product to each canonical vector $\boldsymbol{e}_i = \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}^T$. The system (II.30) is solved by LU factorization, and the nonlinear residual is $\|\Delta U\|_2$. Fig. II.2a shows the convergence curve resulting from a computation initialised with $\boldsymbol{U}_0 = 1$. The quadratic convergence is identified between iteration 8 and 16, where the error drops below 10^{-10}. Fig. II.2b plots the base flow obtained at the end of the computation, which compares well with the theoretical fixed point $\boldsymbol{U}_{\text{theo}}$.

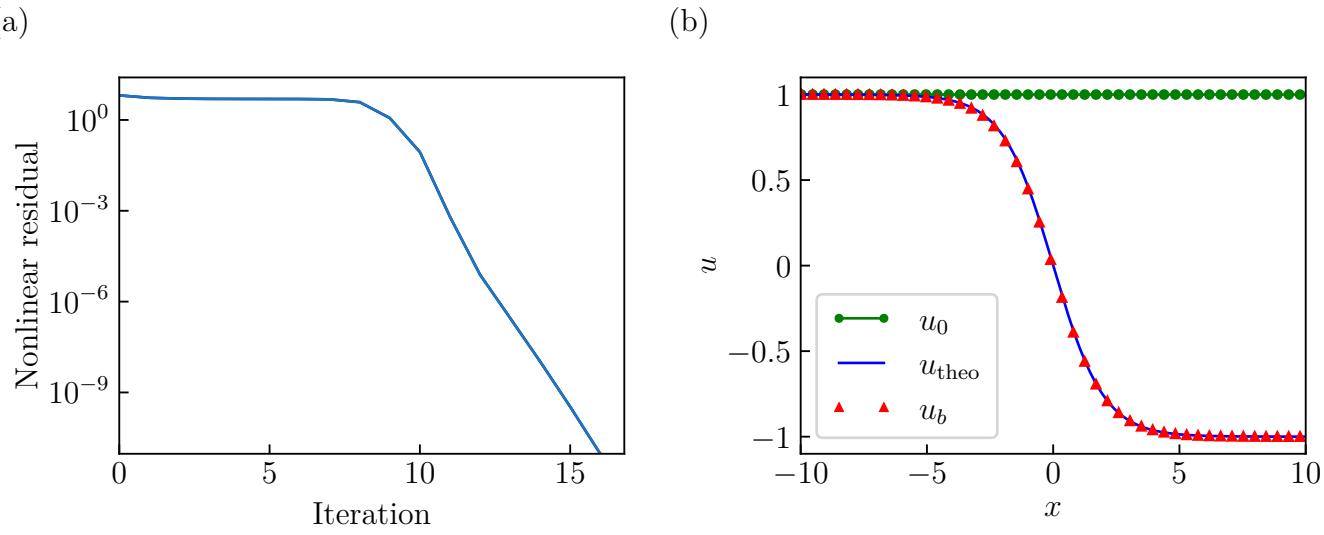


Figure II.2: Convergence curve (a) and initial, theoretical and converged field (b) after 16 iterations for the Newton method.

II.3.2 Implicit backward Euler method

The Newton method might not be robust enough when the physical problem is particularly stiff[2], or when the initial guess is too far from the searched solution. In fluid mechanics, a common practice to overcome this difficulty is solving the nonlinear system (II.27) with an implicit scheme in time. In particular, we focus in this thesis on the linearized first-order backward-Euler method with a local time-step that leads to the following equation:

$$\left(\frac{1}{\Delta t}\mathcal{I}_N - \mathcal{J}(\boldsymbol{W}^n)\right)\Delta\boldsymbol{W} = \boldsymbol{R}(\boldsymbol{W}^n), \tag{II.31}$$

where $\mathcal{I}_N$ is the N-by-N identity matrix. A linear system has to be solved here as well, and one may see that when $\Delta t \to +\infty$, this scheme is equivalent to a step of the Newton method (II.30).

To compute a steady-state solution, Crivellini et al. [69] developed a pseudo-transient continuation technique that uses an adaptative Courant–Friedrichs–Lewy (CFL) number [70] defined as:

$$\begin{cases} \text{CFL} = \min\left(\dfrac{\text{CFL}_{\text{min}}}{r^\beta}, \text{CFL}_{\text{max}}\right), \\ r = \max\left(\dfrac{\|\boldsymbol{R}(\boldsymbol{W}^n)\|_2}{\|\boldsymbol{R}(\boldsymbol{W}^0)\|_2}, \dfrac{\|\boldsymbol{R}(\boldsymbol{W}^n)\|_\infty}{\|\boldsymbol{R}(\boldsymbol{W}^0)\|_\infty}\right), \end{cases} \tag{II.32}$$

where CFL_{min}, CFL_{max} and β are user-defined input parameters. In this work, β is always 1. During computation, the CFL number grows while the residual norms decrease. Since the time-step Δt is proportional to the CFL number, its value grows as the algorithm becomes closer to a steady-state solution. This approach is physically relevant since the goal is to capture asymptotical phenomena that do not vary even during large periods of time. Quadratic convergence is obtained during this late part of the computation. Fig. II.3 shows a numerical experiment on Burgers equation using $\text{CFL}_{\text{min}} = 0.1$ and $\text{CFL}_{\text{max}} = 10^{20}$. The Jacobian operator is here again computed with AD, and the linear system (II.31) is solved by LU factorization.

[2]meaning being highly sensitive to numerical modeling.

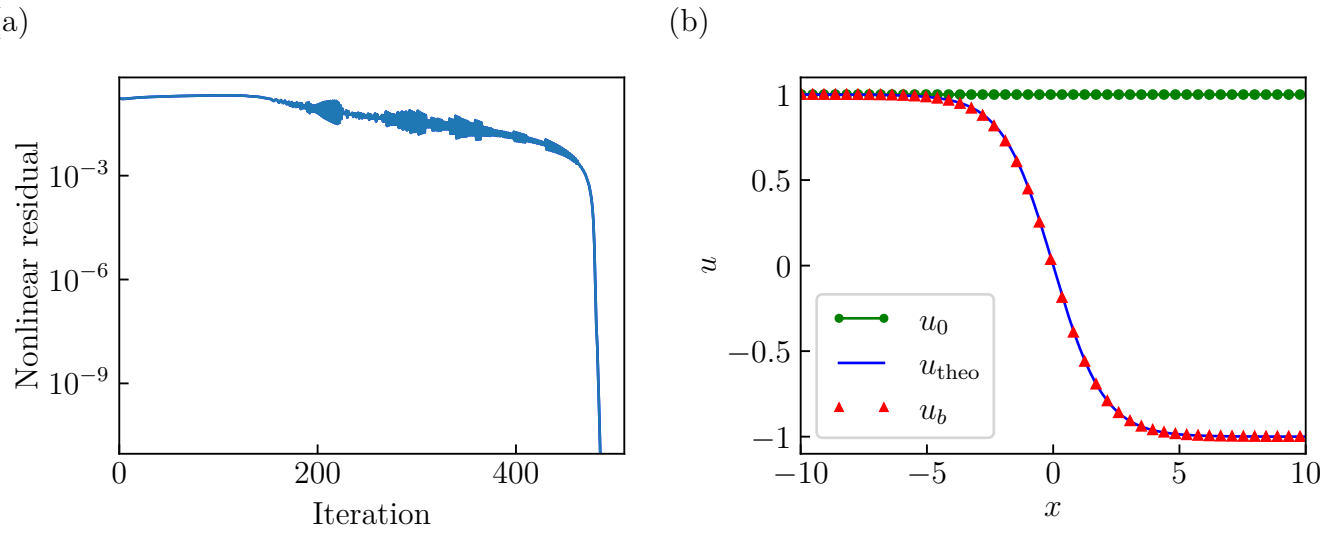


Figure II.3: Convergence curve (a) and initial, theoretical and converged field (b) for the implicit backward Euler with adaptative CFL method.

This fixed point method has been implemented on Burgers equation as well. Fig. II.3a shows the convergence curve resulting from a computation initialised with $\boldsymbol{U}_0 = 1$. During the first 450 iterations, the system is more relaxed than that in Newthod method and converges slower with a lot of oscillations. However, the solution reaches then a quadratical convergence toward the basin of attraction. The resulting field provided in fig. II.3b also presents a good match with $\boldsymbol{U}_{\text{theo}}$, proving the robustness of the method.

Newton-like methods perform well at finding steady solutions with good accuracy. However, to ensure a convergence below 10^{-10}, the Jacobian operator must be computed with high reliability. As mentioned earlier, the illustrative results from this section were obtained with AD to evaluate Jacobian-vectors products. The AD theory and its practical implementation are detailed in the following section.

II.4 Linearization procedure

Whether we work on fixed point computation or global stability analysis (see II.5), critical is the importance of the linearization procedure for direct and adjoint operators. One solution is differentiate equation (II.1) analytically, then to discretize the set of linearized equations [71, 72]. However, as stated in [73], in the compressible case, the consistency from continuous to discrete operator is not guaranteed. The solution used in this thesis is to work in a fully discretized framework in which equation (II.1) is discretized into equation (II.27), and then linearized. Mettot [74] states such strategy has the other advantage of being less error-prone and simpler to implement since all the complexity, such as boundary conditions or numerical schemes, is already accounted for in the discrete linearized equations without additional treatment.

As introduced in [37], one way to approximate the product between $\mathcal{J}$ and a test vector $\boldsymbol{\varphi}$ is to use the following finite differences relation:

$$\mathcal{J}\boldsymbol{\varphi} \approx \frac{1}{\epsilon}\left(\boldsymbol{R}(\boldsymbol{W}_b + \boldsymbol{\varphi}) - \boldsymbol{R}(\boldsymbol{W}_b)\right), \tag{II.33}$$

where ϵ is a small user-defined parameter. Nevertheless, Knoll & Keyes in [75] explained that the

choice of ϵ can be tedious and case-dependent. If too large, the matrix-vector product is poorly approximated, and if too small, floating-point round-off errors could contaminate the result. In this work, the issue has been avoided by using AD tools instead, which provides exact matrix-vector products.

II.4.1 Description of Algorithmic Differentiation with Tapenade

There are two main families of AD tools: the Source Code Transformation (SCT) which replaces the source code for a function by an automatically generated source code that includes statements for calculating the derivatives interleaved with the original instructions; and the Overloading Operator (OO) that computes the derivatives present in a computer code by appending a dual component to all the dependent variables. For performance issues [57], the SCT has been retained in this work. In particular, the choice was made to use the Tapenade software, developed by Hascoët & Pascual [76] as it is the most mature and reputable open-source tool in the community nowadays.

Given $m, l \in \mathbb{N}$ and an algorithm that implements a function $\boldsymbol{G} : \boldsymbol{X} \in \mathbb{R}^m \mapsto \boldsymbol{G}(\boldsymbol{X}) \in \mathbb{R}^l$, Tapenade produces differentiated codes in SCT that computes the exact Jacobian-vector product with a column (forward mode) vector by the right, or a line vector (backward mode) by the left, up to machine precision. Assuming $\boldsymbol{G}$ is differentiable, AD relies on the fact that its implementation is composed of a sequence of p elementary instructions, which are mathematical functions $\boldsymbol{g}$ that can be differentiated one by one. The composition of these functions returns $\boldsymbol{G}$:

$$\boldsymbol{G} = \boldsymbol{g}_p \circ \boldsymbol{g}_{p-1} \circ \cdots \circ \boldsymbol{g}_1. \tag{II.34}$$

From such a relation, the chain rule applies on the derivatives to get the Jacobian operator $\mathcal{J}_{\boldsymbol{G}} = \partial \boldsymbol{G}/\partial \boldsymbol{X}$:

$$\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X}) = \boldsymbol{g}'_p(\boldsymbol{X}_{p-1}) \times \boldsymbol{g}'_{p-1}(\boldsymbol{X}_{p-2}) \times \cdots \times \boldsymbol{g}'_1(\boldsymbol{X}_0) \tag{II.35}$$

with $\boldsymbol{X}_0 = \boldsymbol{X}$, $\boldsymbol{X}_k = \boldsymbol{g}_k(\boldsymbol{X}_{k-1})$ and $\boldsymbol{g}'_k = \partial \boldsymbol{g}_k/\partial \boldsymbol{X}_{k-1}$.

Forward mode. From a given column vector $\boldsymbol{X}^{\mathrm{F}} \in \mathbb{R}^m$, the quantity $\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})\boldsymbol{X}^{\mathrm{F}}$ is evaluated with successive right matrix-vector products:

$$\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})\boldsymbol{X}^{\mathrm{F}} = \boldsymbol{g}'_p(\boldsymbol{X}_{p-1}) \times \underbrace{\boldsymbol{g}'_{p-1}(\boldsymbol{X}_{p-2}) \times \cdots \times \overbrace{\boldsymbol{g}'_1(\boldsymbol{X}_0)\boldsymbol{X}^{\mathrm{F}}}^{=\boldsymbol{X}_1^{\mathrm{F}}}}_{=\boldsymbol{X}_{p-1}^{\mathrm{F}}}, \tag{II.36}$$

which, as shown in [76], is implemented using the following standardized sequence of instructions:

$$\begin{aligned}
\boldsymbol{X}_0 &= \boldsymbol{X} \\
\boldsymbol{X}_0^{\mathrm{F}} &= \boldsymbol{X}^{\mathrm{F}} \\
\boldsymbol{X}_1 &= \boldsymbol{g}_1(\boldsymbol{X}_0) \\
\boldsymbol{X}_1^{\mathrm{F}} &= \boldsymbol{g}_1'(\boldsymbol{X}_0)\boldsymbol{X}_0^{\mathrm{F}} \\
&\cdots \\
\boldsymbol{X}_k &= \boldsymbol{g}_k(\boldsymbol{X}_{k-1}) \\
\boldsymbol{X}_k^{\mathrm{F}} &= \boldsymbol{g}_k'(\boldsymbol{X}_{k-1})\boldsymbol{X}_{k-1}^{\mathrm{F}} \\
&\cdots \\
\boldsymbol{X}_p &= \boldsymbol{g}_p(\boldsymbol{X}_{p-1}) \\
\boldsymbol{X}_p^{\mathrm{F}} &= \boldsymbol{g}_p'(\boldsymbol{X}_{p-1})\boldsymbol{X}_{p-1}^{\mathrm{F}},
\end{aligned} \tag{II.37}$$

and $\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})\boldsymbol{X}^{\mathrm{F}} = \boldsymbol{X}_p^{\mathrm{F}}$.

Backward mode. This mode computes, for a given $\boldsymbol{X}^{\mathrm{B}} \in \mathbb{R}^l$, the quantity ${\boldsymbol{X}^{\mathrm{B}}}^T \mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})$ with successive left matrix-vector products:

$${\boldsymbol{X}^{\mathrm{B}}}^T \mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X}) = \underbrace{\overbrace{\boldsymbol{X}^{\mathrm{B}}\boldsymbol{g}_p'(\boldsymbol{X}_{p-1})}^{=\boldsymbol{X}_{p-1}^{\mathrm{B}}} \times \boldsymbol{g}_{p-1}'(\boldsymbol{X}_{p-2}) \times \ldots}_{=\boldsymbol{X}_1^{\mathrm{B}}} \times \boldsymbol{g}_1'(\boldsymbol{X}_0). \tag{II.38}$$

Note that actually, since ${\boldsymbol{X}^{\mathrm{B}}}^T \mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X}) = (\mathcal{J}_{\boldsymbol{G}}^T(\boldsymbol{X})\boldsymbol{X}^{\mathrm{B}})^T$, the backward mode provides the action of $\mathcal{J}^T$ on a given vector, which is particularly useful in the framework of global stability analysis (see II.5.2 and [77]). Unlike the forward mode, the vectors $\boldsymbol{X}_k$ must all be computed before calculating the first backward quantity $\boldsymbol{X}^{\mathrm{B}}\boldsymbol{g}_p'(\boldsymbol{X}_{p-1})$:

$$\begin{aligned}
\boldsymbol{X}_0 &= \boldsymbol{X} \\
\boldsymbol{X}_1 &= \boldsymbol{g}_1(\boldsymbol{X}_0) \\
&\cdots \\
\boldsymbol{X}_k &= \boldsymbol{g}_k(\boldsymbol{X}_{k-1}) \\
&\cdots \\
\boldsymbol{X}_p &= \boldsymbol{g}_p(\boldsymbol{X}_{p-1}) \\
\boldsymbol{X}_p^{\mathrm{B}} &= \boldsymbol{X}^{\mathrm{B}}\boldsymbol{g}_p'(\boldsymbol{X}^{\mathrm{B}}) \\
\boldsymbol{X}_{p-1}^{\mathrm{B}} &= \boldsymbol{X}_p^{\mathrm{B}}\boldsymbol{g}_{p-1}'(\boldsymbol{X}_{p-2}) \\
&\cdots \\
\boldsymbol{X}_k^{\mathrm{B}} &= \boldsymbol{X}_{k+1}^{\mathrm{B}}\boldsymbol{g}_k'(\boldsymbol{X}_{k-1}) \\
&\cdots \\
\boldsymbol{X}_0^{\mathrm{B}} &= \boldsymbol{X}_1^{\mathrm{B}}\boldsymbol{g}_1'(\boldsymbol{X}_0),
\end{aligned} \tag{II.39}$$

and ${\boldsymbol{X}^{\mathrm{B}}}^T \mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X}) = {\boldsymbol{X}_0^{\mathrm{B}}}^T$.

Appendix A presents an example of AD code in forward and backward modes obtained with Tapenade on a piece of FORTRAN code that computes the L^2 scalar product. One may check the accuracy of the produced routines by comparing their outputs to finite differences using equation (II.33) or to the complex-step derivative approximation introduced by Martins [78]. To validate the compatibility between forward and backward differentiated codes, a so-called duality test may be performed [79]. Given two random vectors $\boldsymbol{X}^{\mathrm{F}} \in \mathbb{R}^m$ and $\boldsymbol{X}^{\mathrm{B}} \in \mathbb{R}^l$, it consists in checking the accuracy of the following equality:

$$\boldsymbol{X}^{\mathrm{B}} \cdot \underbrace{\left(\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})\boldsymbol{X}^{\mathrm{F}}\right)}_{\text{forward}} = \underbrace{\left(\boldsymbol{X}^{\mathrm{B}}\mathcal{J}_{\boldsymbol{G}}(\boldsymbol{X})\right)}_{\text{backward}} \cdot \boldsymbol{X}^{\mathrm{F}}, \tag{II.40}$$

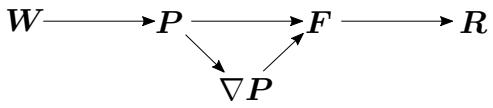
which is in practice achieved by evaluating the relative error $\epsilon_{\mathrm{rel}}^{\mathrm{AD}}$:

$$\epsilon_{\mathrm{rel}}^{\mathrm{AD}} = \frac{\boldsymbol{X}^{\mathrm{B}} \cdot \left(\mathcal{J}_{\boldsymbol{G}}\boldsymbol{X}^{\mathrm{F}}\right) - \left(\boldsymbol{X}^{\mathrm{B}}\mathcal{J}_{\boldsymbol{G}}\right) \cdot \boldsymbol{X}^{\mathrm{F}}}{\left\|\mathcal{J}_{\boldsymbol{G}}\boldsymbol{X}^{\mathrm{F}}\right\|_2}. \tag{II.41}$$

II.4.2 Application to the conservation equations

Significant efforts have been made at ONERA, based on the work of Mary [56], to challenge the two limiting factors of CFD applications: their CPU and memory costs. In elsA CFD solver, this so-called high performance computing (HPC) layer consists of cache-blocking and vectorization optimizations. The computational domain is divided into packs of tiles (cells or interfaces, depending on what is computed) independent of each other for vectorization. Once a pack is loaded in memory, the goal is to perform as many operations as possible before other packs have to be treated. Such a technique optimizes loadings and unloadings of memory cache at the L3 level (see fig. III.1), and allows for the vectorization of calculus loops since each tile is independent of others. In practice, the solver is organised as a chain of operators specified by user choices, which ensures these improvements on direct, forward, and backward computations.

The resulting chain of operators can be represented by a graph. For example, the graph of $\boldsymbol{R} : \boldsymbol{W} \in \mathbb{R}^N \mapsto \boldsymbol{R}(\boldsymbol{W}) \in \mathbb{R}^N$ with second-order schemes for fluxes, without source terms ($\boldsymbol{T}_s = 0$) is:



where $\boldsymbol{P}$ and $\boldsymbol{F}$ are respectively the discretized primitive vector and fluxes defined in II.2.1. Such graph illustrates how each operator must be successively computed and nested one by one to finally give $\boldsymbol{R}(\boldsymbol{W})$. This formulation allows the use of chain rules, and the Jacobian operator $\mathcal{J} = \partial\boldsymbol{R}/\partial\boldsymbol{W}$ can be computed with crossed derivatives:

$$\mathcal{J} = \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}} \left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{P}} + \frac{\partial \boldsymbol{F}}{\partial \nabla \boldsymbol{P}} \frac{\partial \nabla \boldsymbol{P}}{\partial \boldsymbol{P}} \right) \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{W}}. \tag{II.42}$$

Tapenade is able to generate differentiated code for each operators in forward or backward mode. The graph providing the right Jacobian-vector product, i.e. the mapping $\boldsymbol{\varphi} \in \mathbb{R}^N \mapsto \mathcal{J}\boldsymbol{\varphi} \in \mathbb{R}^N$, reads:

$$\boldsymbol{\varphi} \longrightarrow \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{W}}\boldsymbol{\varphi} \longrightarrow \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{P}}\left(\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{W}}\boldsymbol{\varphi}\right) + \frac{\partial \boldsymbol{F}}{\partial \nabla \boldsymbol{P}}\left(\frac{\partial \nabla \boldsymbol{P}}{\partial \boldsymbol{P}}\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{W}}\boldsymbol{\varphi}\right) \longrightarrow \mathcal{J}\boldsymbol{\varphi}$$
$$\searrow \quad \frac{\partial \nabla \boldsymbol{P}}{\partial \boldsymbol{P}}\left(\frac{\partial \boldsymbol{P}}{\partial \boldsymbol{W}}\boldsymbol{\varphi}\right) \quad \nearrow$$

while that of the left Jacobian-vector product, i.e. the mapping $\boldsymbol{\psi} \in \mathbb{R}^N \boldsymbol{\psi}^T \mapsto \mathcal{J} \in \mathbb{R}^N$, reads

$$\boldsymbol{\psi} \longrightarrow \boldsymbol{\psi}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}} \longrightarrow \left(\boldsymbol{\psi}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}}\right)\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{P}} + \left(\boldsymbol{\psi}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}}\frac{\partial \boldsymbol{F}}{\partial \nabla \boldsymbol{P}}\right)\frac{\partial \nabla \boldsymbol{P}}{\partial \boldsymbol{P}} \longrightarrow \boldsymbol{\psi}^T\mathcal{J}$$
$$\searrow \quad \left(\boldsymbol{\psi}^T\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{F}}\right)\frac{\partial \boldsymbol{F}}{\partial \nabla \boldsymbol{P}} \quad \nearrow$$

where the terms between parenthesis are line or column vectors following the mode used.

Once coded and assembled, the chain of operators can be integrally differentiated in forward or backward mode to get the action of both $\mathcal{J}$ and $\mathcal{J}^T$ (with the relation $\mathcal{J}^T\boldsymbol{\psi} = (\boldsymbol{\psi}^T\mathcal{J})^T$) on a given vector at machine precision. The use of SCT allows the preservation of all HPC optimizations as in the direct solver, unlike the OO differentiation [57]. As depicted in II.4.1, the validity of each differentiated operator is verified by comparing with finite differences, and the compatibility between forward and backward operators is checked using duality test (see equation (II.40)). Such a strategy has been implemented for the Burgers equation and the resulting duality error associated with $\mathcal{J}(U_{\text{theo}})$ is $\epsilon_{\text{rel}}^{\text{AD}} = 1.6467 \times 10^{-16}$. This is below machine precision, validating the differentiated codes produced by Tapenade.

II.5 Global stability analysis

The stability of a base flow can be studied by analyzing its behavior under the effect of a perturbation. As discussed in I.2, one or more global modes are necessarily involved in the asymptotical behaviour of an unstable base flow. Their spatial support, amplification rate and frequency can be computed by finding the least stable eigenpairs of $\mathcal{J}$ [22]. When the baseflow is stable, the global modes are exponentially and asymptotically damped, and are not sufficient to give explanation of transient energy growth in the system. The resolvent analysis [36, 20] examines such short-term instability providing the most amplified frequency (and the associated forcing and response) of a given sub-critical flow. The direct and adjoint global stability as well as the resolvent analysis in their discrete form are detailed hereafter.

II.5.1 Direct modal analysis

By analogy with I.2, the discrete formulation of the direct eigenvalue problem is detailed hereafter. We assume first the existence of a steady solution $\boldsymbol{W}_b$ of (II.27) perturbed by a small disturbance $\boldsymbol{W}'$. The discrete residual can be linearized about $\boldsymbol{W}_b$ using:

$$\boldsymbol{R}(\boldsymbol{W}_b + \boldsymbol{W}') \simeq \underbrace{\boldsymbol{R}(\boldsymbol{W}_b)}_{=0} + \mathcal{J}(\boldsymbol{W}_b)\boldsymbol{W}'. \tag{II.43}$$

For the sake of conciseness, the notation $\mathcal{J}$ will be used to denote $\mathcal{J}(\boldsymbol{W}_b)$ in the following. Injecting this last quantity in equation (II.27) results in the so-called linearized Navier-Stokes (LNS) equations that govern the temporal evolution of $\boldsymbol{W}'$:

$$\frac{\partial \boldsymbol{W}'}{\partial t} = \mathcal{J}\boldsymbol{W}'. \tag{II.44}$$

As stated for instance by Sipp et al. in [22], these disturbances may be decomposed into modes of the form $\boldsymbol{W}'(\boldsymbol{x},t) = \widehat{\boldsymbol{W}}(\boldsymbol{x})e^{\lambda t}$, where $\widehat{\boldsymbol{W}}$ is an eigenfunction of $\mathcal{J}$ representing the spatial structure of this direct mode. The quantities $\sigma = \Re(\lambda)$ and $\omega = \Im(\lambda)$ respectively stand for the growth rate and angular frequency associated with $\boldsymbol{W}'$. Equation (II.44) then reduces to the following eigenproblem:

$$\lambda\widehat{\boldsymbol{W}} = \mathcal{J}\widehat{\boldsymbol{W}}. \tag{II.45}$$

$\boldsymbol{W}_b$ is said to be unstable (respectively stable) if there exists at least one eigenvalue of $\mathcal{J}$ such that $\sigma > 0$ (respectively $\sigma < 0$). The global stability of $\boldsymbol{W}_b$ is thus analyzed by computing the spectrum of $\mathcal{J}$. In this work for the sake of generality, we assume that the flow does not exhibit any spatial homogeneity, but when there is one or more, the study simplifies into bi-global or local stability analysis [80].

Usually, only the least stable modes of $\mathcal{J}$ are of interest, and the eigenproblem (II.45) is solved with an iterative eigensolver[3] like the Krylov-Schur algorithm (KS, presented later in section II.6). This algorithm is best at finding the eigenvalues of the largest magnitudes of the considered operator, which are not necessarily the least stable ones. To target the least stable part of the spectrum, the spectral shift-and-invert transformation is applied: a complex-valued shift s is introduced, and the following transformed complex eigenproblem is solved instead:

$$\widetilde{\lambda}\widehat{\boldsymbol{W}} = (\mathcal{J} - s\mathcal{I}_N)^{-1}\widehat{\boldsymbol{W}}, \tag{II.46}$$

where $\widetilde{\lambda} = (\lambda - s)^{-1}$. This strategy is further explained in II.6.5, which highlights its efficiency to filter the eigenvalues closest to s.

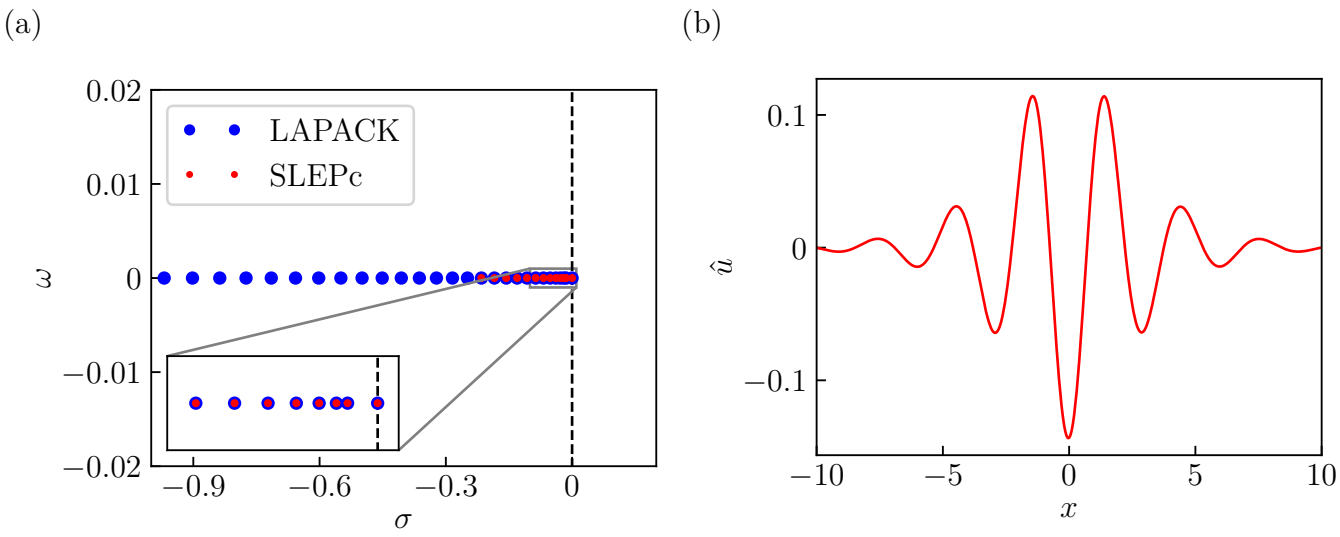


Figure II.4: Spectrum of $\mathcal{J}$ (a) computed by SLEPc and LAPACK and real part of eigenmode calculated by SLEPc (b) corresponding to the least stable eigenvalue $\lambda = -4.55 \times 10^{-6}$.

[3] rather than a direct eigensolver that returns the entire spectrum.

The stability of the Burgers equation's fixed point $\boldsymbol{W}_b$ computed in II.3.2, has been tested. The computation of eigenpairs was carried out by both the direct QR algorithm (given the relatively small dimensions of $\mathcal{J}$) with LAPACK [81]; and by SLEPc [82] with a KS algorithm with shift-and-invert transformation, where only the least stable eigenvalues are looked for. The inversions of the complex operator are managed in SLEPc by LU factorization, as in section II.3. The resulting spectra are showed in fig. II.4a, where a good match between LAPACK and SLEPc for the least stable eigenvalues. We also notice that $\boldsymbol{W}_b$ is a stable base flow. The least stable eigenvector computed with SLEPc is presented in fig. II.4b. The largest part of the global mode is located is located at the center of the domain.

II.5.2 Adjoint modal analysis

To describe the adjoint modal analysis in its discrete form, we first introduce a discrete scalar product $\langle \cdot, \cdot \rangle_{\mathcal{Q}} : \mathbb{C}^{N \times N} \to \mathbb{C}$, represented by the mass matrix $\mathcal{Q} \in \mathbb{R}^N \times \mathbb{R}^N$. For two given vectors $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{C}^N$, it reads:

$$\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\mathcal{Q}} = \boldsymbol{A}^H \mathcal{Q} \boldsymbol{B}, \tag{II.47}$$

where the superscript H denotes the conjugate transpose. Assuming that $\mathcal{Q}$ is a non-singular symmetric matrix, the discrete adjoint operator $\mathcal{J}^\dagger$ of $\mathcal{J}$ is defined relatively to $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$ such that $\langle \boldsymbol{A}, \mathcal{J}\boldsymbol{B} \rangle_{\mathcal{Q}} = \langle \mathcal{J}^\dagger \boldsymbol{A}, \boldsymbol{B} \rangle_{\mathcal{Q}}$ for all vectors $\boldsymbol{A}$ and $\boldsymbol{B}$. Since $\mathcal{J}$ is real and $\mathcal{Q}$ is non-singular, an explicit relation between $\mathcal{J}^\dagger$ and $\mathcal{J}^T$ is obtained:

$$\mathcal{J}^\dagger = \mathcal{Q}^{-1} \mathcal{J}^T \mathcal{Q}. \tag{II.48}$$

The matrices $\mathcal{J}^\dagger$ and $\mathcal{J}^T$ are thus similar and share the same spectrum. We can therefore focus on computing the eigenvalues of $\mathcal{J}^T$ formed with the differentiated code provided by Tapenade in backward mode, and pre-multiply the resulting modal basis with $\mathcal{Q}^{-1}$ to recover adjoint eigenfunctions. For a given direct eigenpair $(\lambda_k, \widehat{\boldsymbol{W}}_k)$, a corresponding adjoint eigenpair $(\overline{\lambda}_k, \widehat{\boldsymbol{W}}_k^\dagger)$ solution of the following adjoint eigenproblem exists:

$$\overline{\lambda}_k \widehat{\boldsymbol{W}}_k^\dagger = \mathcal{J}^\dagger \widehat{\boldsymbol{W}}_k^\dagger, \tag{II.49}$$

where $\widehat{\boldsymbol{W}}_k^\dagger$ is the adjoint eigenfunction. Solving (II.49) is achieved similarly to problem (II.45): the eigenpairs of $\mathcal{J}^T$ are computed using the shift-and-invert transformation, and the resulting eigenvectors are pre-multiplied by $\mathcal{Q}^{-1}$ to get the adjoint global modes.

In fluid mechanics, $\mathcal{J}$ is often non-normal[4] and due to the nonorthogonality of eigenmodes, the expansion of $\boldsymbol{W}'$ over the direct modal basis $(\widehat{\boldsymbol{W}}_k, k \geq 1)$ can't be achieved easily [22]. Such a difficulty can be overcome by introducing the dual basis made of the adjoint eigenmodes $(\widehat{\boldsymbol{W}}_k^\dagger, k \geq 1)$, which can be normalized such that $\langle \widehat{\boldsymbol{W}}_k^\dagger, \widehat{\boldsymbol{W}}_l \rangle_{\mathcal{Q}} = \delta_{kl}$. The direct and adjoint modal bases form a bi-orthogonal basis over which $\boldsymbol{W}'$ can be uniquely expressed:

$$\boldsymbol{W}' = \sum_{k \geq 1} \langle \widehat{\boldsymbol{W}}_k^\dagger, \boldsymbol{W}' \rangle_{\mathcal{Q}} \widehat{\boldsymbol{W}}_k. \tag{II.50}$$

The adjoint mode provides the receptivity of the direct mode [32] to a given forcing or variation of an initial condition for instance. Additionally, the adjoint mode is the exact initial perturbation that triggers the direct mode in the flow.

[4] verifying $\mathcal{J}\mathcal{J}^T = \mathcal{J}^T\mathcal{J}$.

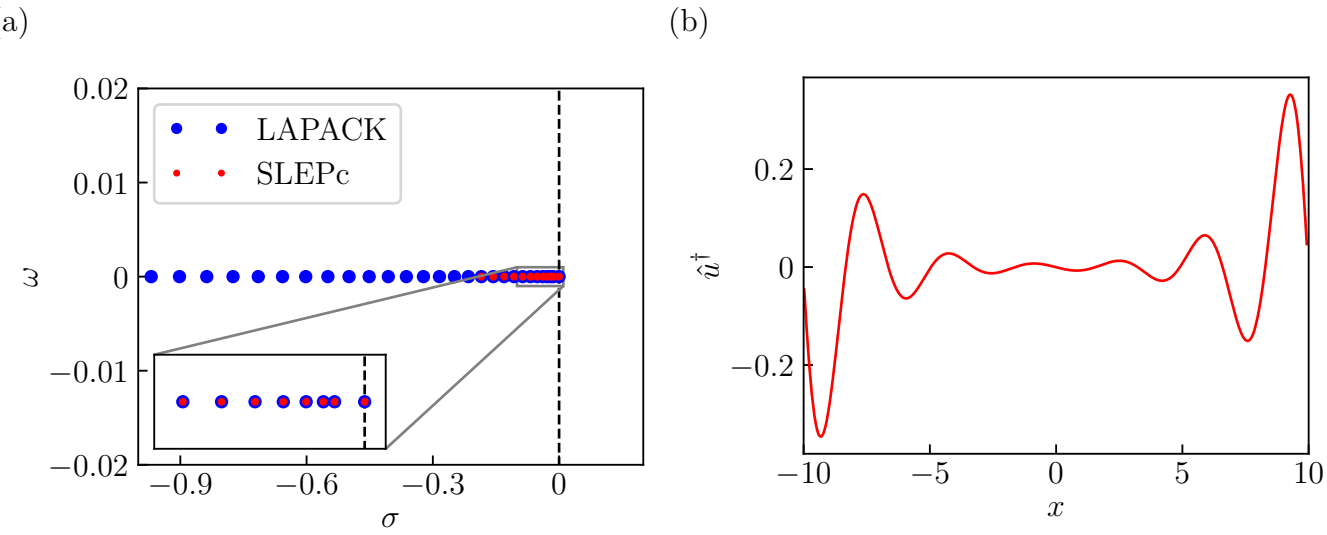


Figure II.5: Spectrum of $\mathcal{J}^\dagger$ (a) computed by SLEPc and LAPACK and real part of eigenmode calculated by SLEPc (b) corresponding to the least stable eigenvalue $\lambda = -4.55 \times 10^{-6}$.

In finite volumes, the scalar product is usually chosen to correspond to the L^2-norm. On a domain Ω, the latter is defined as $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\mathcal{Q}_\Omega} = \int_\Omega \boldsymbol{A}^H \boldsymbol{B} \, \mathrm{d}\Omega = \boldsymbol{A}^H \mathcal{Q}_\Omega \boldsymbol{B}$, where $\mathcal{Q}_\Omega$ is a real diagonal matrix whose coefficients are the volumes $V(\Omega_k)$ of each cell. A numerical experiment on the Burgers equation in the same conditions as for the direct eigenproblem has been carried out, and the resulting spectra are showed in fig. II.5a. Again, a good match between LAPACK and SLEPc for the least stable eigenvalues is noticed. The least stable adjoint eigenfunction computed with SLEPc is presented in fig. II.5b. The largest parts of the adjoint global mode is mainly located at the limits of the domain.

II.5.3 Resolvent analysis

Based on the theoretical framework introduced in I.2, the discrete version of the resolvent analysis, introduced by Trefethen & al. [36], is depicted hereafter. The goal is to analyse the transient behaviour of a stable base flow under the effect of an external forcing $\boldsymbol{\Phi}'$. Adding such a term in equation (II.44) leads to:

$$\frac{\partial \boldsymbol{W}'}{\partial t} = \mathcal{J}\boldsymbol{W}' + \boldsymbol{\Phi}'. \tag{II.51}$$

The Fourier transform of equation (II.51) gives:

$$i\omega \widehat{\boldsymbol{W}} = \mathcal{J}\widehat{\boldsymbol{W}} + \hat{\boldsymbol{\Phi}}. \tag{II.52}$$

Introducing the harmonic response $\widehat{\boldsymbol{W}}$ and forcing $\hat{\boldsymbol{\Phi}}$ at the frequency $\omega \in \mathbb{R}$, such that $\boldsymbol{W}' = \widehat{\boldsymbol{W}}(\boldsymbol{x})e^{i\omega t}$ and $\boldsymbol{\Phi}' = \hat{\boldsymbol{\Phi}}(\boldsymbol{x})e^{i\omega t}$, equation (II.51) can be recast as:

$$\widehat{\boldsymbol{W}} = \mathcal{R}(\omega)\hat{\boldsymbol{\Phi}}, \tag{II.53}$$

with $\mathcal{R}(\omega) = (i\omega\mathcal{I} - \mathcal{J})^{-1}$ the resolvent operator, a transfert function between $\widehat{\boldsymbol{W}}$ and $\hat{\boldsymbol{\Phi}}$ that is a selective noise-amplifier. For the sake of conciseness, the notation $\mathcal{R}$ will be used to denote $\mathcal{R}(\omega)$ in the following. The forcing responsible for the maximum energetic response at the frequency ω

in the flow is computed by maximizing the energetic gain $\eta(\omega) = \|\widehat{\boldsymbol{W}}\|_{\mathcal{Q}_1}/\|\hat{\boldsymbol{\Phi}}\|_{\mathcal{Q}_2}$ associated to $\mathcal{R}$, where the norms are respectively defined relatively to the scalar products $\mathcal{Q}_1$ and $\mathcal{Q}_2$. As for the adjoint discrete problem, $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are non-singular symmetric real matrices. In practice, $\mathcal{Q}_1 = \mathcal{Q}_\Omega$ and $\mathcal{Q}_2$ is associated to kinetic or acoustic energy. In the computations carried out with elsA in this work, $\mathcal{Q}_2$ is associated with the Chu energy [83] (see equation (III.110)). Considering $i\omega$ not to be an eigenvalue of $\mathcal{R}$, meaning $\mathcal{R}$ and $\mathcal{R}^H$ are defined[5], the quotient $\eta(\omega)$ can be expanded as:

$$\eta = \frac{\langle\widehat{\boldsymbol{W}},\widehat{\boldsymbol{W}}\rangle_{\mathcal{Q}_1}}{\langle\hat{\boldsymbol{\Phi}},\hat{\boldsymbol{\Phi}}\rangle_{\mathcal{Q}_2}} = \frac{\langle\mathcal{R}\hat{\boldsymbol{\Phi}},\mathcal{R}\hat{\boldsymbol{\Phi}}\rangle_{\mathcal{Q}_1}}{\langle\hat{\boldsymbol{\Phi}},\hat{\boldsymbol{\Phi}}\rangle_{\mathcal{Q}_2}} = \frac{\hat{\boldsymbol{\Phi}}^H\mathcal{R}^H\mathcal{Q}_1\mathcal{R}\hat{\boldsymbol{\Phi}}}{\hat{\boldsymbol{\Phi}}^H\mathcal{Q}_2\hat{\boldsymbol{\Phi}}}, \tag{II.54}$$

with $\mathcal{R}^H = (-i\omega\mathcal{I}_N - \mathcal{J}^T)^{-1}$. Since $\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$ is Hermitian (i.e. $\mathcal{R}^H\mathcal{Q}_1\mathcal{R} = (\mathcal{R}^H\mathcal{Q}_1\mathcal{R})^H$), such a relation corresponds to a Rayleigh quotient and the problem can be recast in the following way:

$$\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}\hat{\boldsymbol{\Phi}} = \eta^2\hat{\boldsymbol{\Phi}}. \tag{II.55}$$

The solutions of this eigenproblem are the eigenpairs of the Hermitian operator $\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$. Finding the forcing that maximizes η is thus equivalent to calculate the eigenvector associated to the largest eigenvalue of $\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$. The gain η can alternatively be expanded with the harmonic response, leading to the following eigenproblem:

$$\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\widehat{\boldsymbol{W}} = \eta^2\widehat{\boldsymbol{W}}. \tag{II.56}$$

Both operators $\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$ and $\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$ share the same real spectrum, and their eigenvectors respectively form the basis of optimal forcing ($\hat{\boldsymbol{\Phi}}_k, k \geq 1$) and response ($\widehat{\boldsymbol{W}}_k, k \geq 1$). The response $\widehat{\boldsymbol{W}}_k$ associated with the k-th gain η_k is linked with the forcing $\hat{\boldsymbol{\Phi}}_k$ by the relation:

$$\eta_k\widehat{\boldsymbol{W}}_k = \mathcal{R}\hat{\boldsymbol{\phi}}_k, \tag{II.57}$$

and the harmonic response can be decomposed in the following way:

$$\widehat{\boldsymbol{W}} = \sum_{k\geq 1}\eta_k\langle\hat{\boldsymbol{\Phi}}_k,\hat{\boldsymbol{\Phi}}\rangle\widehat{\boldsymbol{W}}_k. \tag{II.58}$$

The vectors composing the forcing and response space respectively are right and left singular vectors of $\mathcal{R}$, and the gains are its singular values. The curve of $\eta(\omega)$ is established by solving one of the eigenproblems (II.55) or (II.56). In this work, we focus on the problem (II.56) as to both benefit from the optimized Jacobian-vector products and avoid inversions when recovering the responses from the forcings. For a given frequency ω, $\hat{\boldsymbol{\Phi}}_1$ is the optimal forcing, i.e. the forcing providing the most energetic response in the flow. In general, $\eta_1 \gg \eta_{k\geq 2}$ and the gains $\eta_{k\geq 2}$ are the sub-optimal ones.

The KS algorithm is suitable to solve the eigenproblem as it provides the eigenvalues of largest magnitudes of $\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$. One may notice that this operator is composed of the inverse matrices $\mathcal{R} = (i\omega\mathcal{I}_N - \mathcal{J})^{-1}$ and $\mathcal{R}^H = (-i\omega\mathcal{I}_N - \mathcal{J}^T)^{-1}$, meaning its application to a given vector requires two inversions of complex linear systems. The computation of the KS basis (see II.6.2) is hence much more expansive than for the modal analysis. However, as the optimal gain is much larger than sub-optimal ones, the KS process usually requires less iteration to accurately converge eigenpairs, as shown for instance in chapter IV. A resolvent analysis has been carried out on the Burgers equation with $\mathcal{Q}_1 = \mathcal{Q}_2 = \mathcal{Q}_\Omega$. The eigenproblem is solved with SLEPc and inversions are managed with LU

[5] when $\mathcal{R}$ is not defined, $i\omega$ is a marginally stable eigenvalue of $\mathcal{J}$.

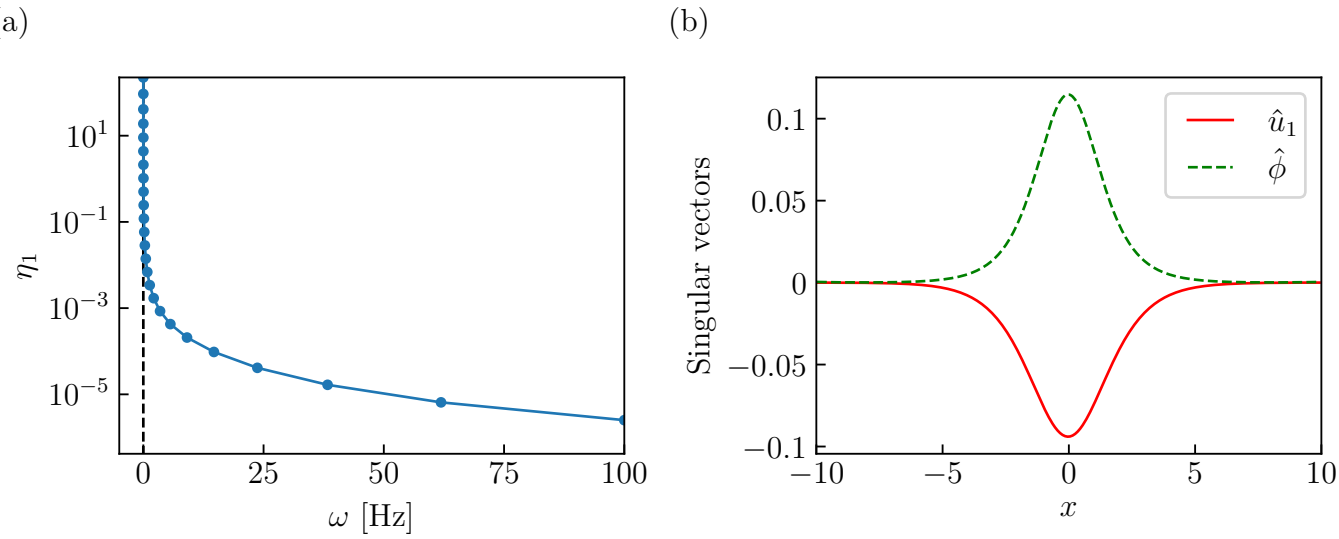


Figure II.6: Optimal gain $\eta_1(\omega)$ (a) and real parts of corresponding optimal forcing and response (b) at $\omega = 10^{-3}$ Hz.

factorization. The gain curve in fig. II.6a indicates a peak at the zero frequency, corresponding to a constant forcing that leads to a constant response. The real parts of optimal forcing and response at $\omega = 10^{-3}$ Hz are presented in fig. II.6b.

This section was devoted to the derivation of the global stability tools in their discrete forms. As mentioned several times, both modal and non modal analyses require the resolution of complex-valued eigenproblems, usually addressed by Arnoldi-like methods like the KS algorithm. The details of the KS algorithm are exposed hereafter.

II.6 Krylov-Schur algorithm

The KS algorithm, introduced by Stewart in [46], is an Arnoldi-based method that projects the eigenvectors of a given complex operator onto a Krylov space. This iterative process provides approximate eigenpairs, called Ritz pairs, such that the eigenvalues are of largest magnitude. In this section, some concepts about the Krylov decomposition through Arnoldi process, as well as the mechanics of the KS algorithm are exposed.

II.6.1 Krylov subspace and Arnoldi algorithm

The Krylov subspace $K_m(\mathcal{C}, \boldsymbol{v})$ of order m, of an operator $\mathcal{C} \in \mathbb{C}^N \times \mathbb{C}^N$ associated to a given vector $\boldsymbol{v} \in \mathbb{C}^N$, is defined as follows:

$$K_m(\mathcal{C}, \boldsymbol{v}) = \text{span}\left\{\boldsymbol{v}, \mathcal{C}\boldsymbol{v}, \mathcal{C}^2\boldsymbol{v}, \ldots, \mathcal{C}^{m-1}\boldsymbol{v}\right\}. \tag{II.59}$$

The principle of Arnoldi-based methods, as shown in [47], is to project $\mathcal{C}$ onto K_m and extract eigenpairs of this projection, which are approximations of the effective eigenpairs of $\mathcal{C}$. To do this, an orthonormal basis of K_m is produced using the Arnoldi process (presented in Algorithm 1), based on the modified Gram-Schmidt algorithm [84], and the matrix $\mathcal{C}$ is reduced to an upper Hessenberg m-by-m matrix $\mathcal{H}_m$. The output of Arnoldi process is a particular case of Krylov

Algorithm 1 Arnoldi process

Input: $\mathcal{C}$, $\boldsymbol{v}_1$ unitary (i.e. $\|\boldsymbol{v}\|_2 = 1$) and $m \in \mathbb{N}$
Output: $\mathcal{V}_m$, $\mathcal{H}_m$, $\boldsymbol{v}_{m+1}$ and $h_{m+1,m}$

```
for k ∈ [[1, m]] do
    v_{k+1} ← C v_k
    for l ∈ [[1, k]] do
        h_{kl} ← ⟨C v_k, v_l⟩_2
        v_{k+1} ← v_{k+1} − v_l h_{kl}
    end for
    h_{k+1,k} ← ‖v_{k+1}‖_2
    if h_{k+1,k} = 0 then
        stop
    else
        v_{k+1} ← v_{k+1} / h_{k+1,k}
    end if
end for
```

decomposition:

$$\mathcal{C}\mathcal{V}_m = \mathcal{V}_m\mathcal{H}_m + \boldsymbol{v}_{m+1}\hat{\boldsymbol{h}}_{m+1,m}^H, \tag{II.60}$$

where $\mathcal{V}_m$ is an N-by-m matrix whose columns are an orthonormal basis of K_m, and $\hat{\boldsymbol{h}}_{m+1,m}^H = h_{m+1,m}\boldsymbol{e}_m^H$.

II.6.2 General Krylov decomposition

Let us now consider an order-m general Krylov decomposition that reduces the matrix $\mathcal{C}$ to a matrix $\mathcal{B}_m$ with an orthonormal basis $\mathcal{U}_m$:

$$\mathcal{C}\mathcal{U}_m = \mathcal{U}_m\mathcal{B}_m + \boldsymbol{u}_{m+1}\hat{\boldsymbol{b}}_m^H. \tag{II.61}$$

Using the orthonormality of the columns of $\mathcal{U}_m$ and pre-multiplying with $\mathcal{U}_m^H$ provides:

$$\mathcal{U}_m^H\mathcal{C}\mathcal{U}_m = \mathcal{B}_m, \tag{II.62}$$

where $\mathcal{B}_m$ is an approximate space whose spectrum is composed of the Ritz values $\lambda_k \in \mathbb{C}$. The Ritz vectors define the projection of the eigenspace of $\mathcal{C}$ onto $\mathcal{B}_m$. The k-th Ritz vector is defined as $\mathcal{U}_m\boldsymbol{y}_k$, with $\boldsymbol{y}_k \in \mathbb{C}^m$ the k-th eigenvector of $\mathcal{B}_m$. The decomposition (II.61) can be post-multiplied by $\boldsymbol{y}_k$ to get:

$$\mathcal{C}\mathcal{U}_m\boldsymbol{y}_k = \mathcal{U}_m\mathcal{B}_m\boldsymbol{y}_k + \boldsymbol{u}_{m+1}\hat{\boldsymbol{b}}_m^H\boldsymbol{y}_k. \tag{II.63}$$

After recasting, taking the L^2-norm and due to the unitarity of $\boldsymbol{u}_{m+1}$, the exact error commited on the k-th Ritz vector is:

$$\|\mathcal{C}\mathcal{U}_m\boldsymbol{y}_k - \lambda\mathcal{U}_m\boldsymbol{y}_k\|_2 = |\hat{\boldsymbol{b}}_m^H\boldsymbol{y}_k|. \tag{II.64}$$

The term $|\hat{\boldsymbol{b}}^H\hat{\boldsymbol{y}}_k|$ is relatively cheap to compute as m is several orders of magnitude smaller than N. When $|\hat{\boldsymbol{b}}^H\hat{\boldsymbol{y}}_k|$ is sufficiently small, the Ritz pair $(\lambda_k, \mathcal{U}_m\boldsymbol{y}_k)$ is considered to be an approximate eigenpair of $\mathcal{C}$:

$$\mathcal{C}\mathcal{U}_m\boldsymbol{y}_k \simeq \lambda\mathcal{U}_m\boldsymbol{y}_k, \tag{II.65}$$

It can additionally be shown that an orthogonal transformation of $\mathcal{B}_m$ produces a new Krylov decomposition equivalent to (II.61). We introduce a m-by-m unitary matrix $\mathcal{W}$[6] that defines the unitarily equivalent transformation of $\mathcal{B}_m = \mathcal{W}^H \mathcal{T}_m \mathcal{W}$, with $\mathcal{T}_m \in \mathbb{C}^m \times \mathbb{C}^m$. Both $\mathcal{B}_m$ and $\mathcal{T}_m$ have the same Ritz values and equation (II.62) can be recast as:

$$\mathcal{W}^H \mathcal{U}_m^H \mathcal{C} \mathcal{U}_m \mathcal{W} = \mathcal{T}_m, \tag{II.66}$$

which means that $\mathcal{T}_m$ is also an approximate space. It can further generate the following new Krylov decomposition:

$$\mathcal{C}\mathcal{U}_m\mathcal{W} = \mathcal{U}_m\mathcal{W}\mathcal{T}_m + \boldsymbol{u}_{m+1}\hat{\boldsymbol{b}}_m^H\mathcal{W}, \tag{II.67}$$

with $\mathcal{U}_m\mathcal{W}$ the new Krylov space. This illustrates the possibility of transforming a Krylov decomposition without modifying the spectral properties of the initial approximate space. In the KS algorithm, this helpful property is used to sort the targeted portion of the spectrum of $\mathcal{C}$ following a provided criterion as described hereafter.

II.6.3 Krylov-Schur process

The KS process is based on the iterative expansion and contraction of a Krylov decomposition, by keeping the spectral informations of interest and enriching them at each iteration. More details about this method, as well as its variations (like spectral KS) and the various possible implementations, can be found in [46, 85, 86].

Let us first produce an order-m Krylov decomposition of operator $\mathcal{C}$ using the Arnoldi process:

$$\mathcal{C}\mathcal{V}_m = \mathcal{V}_m\mathcal{H}_m + \boldsymbol{v}_{m+1}\hat{\boldsymbol{h}}_{m+1,m}^H. \tag{II.68}$$

With the QR algorithm [44], a Schur matrix $\mathcal{S}_m$ containing the Ritz values on its diagonal can be calculated from the approximate space $\mathcal{H}_m$, resulting in the relation $\mathcal{H}_m = \mathcal{W}_1\mathcal{S}_m\mathcal{W}_1^H$, with $\mathcal{W}_1$ a unitary matrix. A sorting represented by the unitary transformation $\mathcal{W}_2$ can be applied without modifying the spectral content of $\mathcal{H}_m$. In practice, the KS algorithm performs best at computing the eigenvalues of largest magnitude. To do so, $\mathcal{Q}_2$ is chosen so that the Ritz values are arranged in descending magnitude from the top left corner to the bottom right one of $\mathcal{S}_m$, leading to the sorted Schur matrix $\mathcal{T}_m$. From the results of II.6.2, by introducing the KS basis $\widetilde{\mathcal{V}}_m = \mathcal{V}_m\mathcal{W}_1\mathcal{W}_2$ and $\hat{\boldsymbol{b}}_m^H = \hat{\boldsymbol{h}}_{m+1,m}^H\mathcal{W}_1\mathcal{W}_2$, the new decomposition

$$\mathcal{C}\widetilde{\mathcal{V}}_m = \widetilde{\mathcal{V}}_m\mathcal{T}_m + \boldsymbol{v}_{m+1}\hat{\boldsymbol{b}}_m^H \tag{II.69}$$

is a Krylov decomposition equivalent to (II.68). This relation can also be writen as:

$$\mathcal{C}\widetilde{\mathcal{V}}_m = \widetilde{\mathcal{V}}_m \begin{bmatrix} \mathcal{T}_p & \overset{\star}{} \\ 0 & \mathcal{T}_{m-p} \end{bmatrix} + \boldsymbol{v}_{m+1} \begin{bmatrix} \hat{\boldsymbol{b}}_p^H & \hat{\boldsymbol{b}}_{m-p}^H \end{bmatrix}, \tag{II.70}$$

where $\mathcal{T}_p$ contains the spectral information relative to the p wanted Ritz vectors. The upper triangular shape of $\mathcal{T}_m$ allows the decomposition to be truncated at order p:

$$\mathcal{C}\widetilde{\mathcal{V}}_p = \widetilde{\mathcal{V}}_p\mathcal{T}_p + \boldsymbol{v}_{p+1}\hat{\boldsymbol{b}}_p^H, \tag{II.71}$$

with $\boldsymbol{v}_{p+1} = \boldsymbol{v}_{m+1}$. This step of truncation purges the unwanted part of the spectrum before a re-extension at order m through $m-p$ steps of the Arnoldi process is performed at next iteration.

[6] verifying $\mathcal{W}\mathcal{W}^H = \mathcal{I}_m$.

From the second iteration, the approximate space is not a Hessenberg matrix anymore. For instance, if $m = 6$ and $p = 3$, the unsorted new approximate space has the following shape:

$$\begin{pmatrix} s & s & s & h & h & h \\ 0 & s & s & h & h & h \\ 0 & 0 & s & h & h & h \\ b & b & b & h & h & h \\ 0 & 0 & 0 & h & h & h \\ 0 & 0 & 0 & 0 & h & h \end{pmatrix}. \tag{II.72}$$

II.6.4 Deflation of Ritz values

This process of successive extension and contraction is repeated until the desired number of Ritz values are converged. Given a Ritz value λ_k and based on equation (II.73), the error associated to the Ritz vector $\widetilde{\mathcal{V}}_m \boldsymbol{y}_k$ is easily calculated:

$$\|\mathcal{C}\widetilde{\mathcal{V}}_m \boldsymbol{y}_k - \lambda_k \widetilde{\mathcal{V}}_m \boldsymbol{y}_k\|_2 = |\hat{\boldsymbol{b}}_m^H \boldsymbol{y}_k|. \tag{II.73}$$

In this work, the relative error is rather used to test the eigenpairs:

$$\epsilon_{\text{rel}}^{\text{KS}} = \frac{\|\mathcal{C}\widetilde{\mathcal{V}}_m \boldsymbol{y}_k - \lambda_k \widetilde{\mathcal{V}}_m \boldsymbol{y}_k\|_2}{|\lambda_k| \|\widetilde{\mathcal{V}}_m \boldsymbol{y}_k\|_2} = \frac{|\hat{\boldsymbol{b}}_m^H \boldsymbol{y}_k|}{|\lambda_k| \|\widetilde{\mathcal{V}}_m \boldsymbol{y}_k\|_2}. \tag{II.74}$$

Algorithm 2 KS algorithm

Input: $\mathcal{C}$, $\boldsymbol{v}_1$ unitary, m and $l \leq m$ the number of desired eigenpairs
Output: $\mathcal{V}_j$, $\mathcal{H}_j$ such that $\mathcal{C}\mathcal{V}_j = \mathcal{V}_j \mathcal{H}_j$, with $j \leq l$

initialization of $\mathcal{V}_m \leftarrow \begin{bmatrix} \boldsymbol{v}_1 & 0 \end{bmatrix}$, $k \leftarrow 0$, $p \leftarrow 0$
while $j \leq l$ **do**
  $m - p$ Arnoldi steps with deflation:

$$\mathcal{C}\widetilde{\mathcal{V}}_m = \widetilde{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{v}_{m+1} \hat{\boldsymbol{b}}_m^H$$

  Schur decomposition with sorting $\mathcal{H}_m \leftarrow \mathcal{W}_2^H \mathcal{W}_1^H \mathcal{H}_m \mathcal{W}_1 \mathcal{W}_2$
  $\mathcal{V}_m \leftarrow \mathcal{V}_m \mathcal{W}_1 \mathcal{W}_2$
  for $k \in [\![j, l]\!]$ **do**
    compute $(\lambda_k, \boldsymbol{y}_k)$ verifying $\mathcal{H}_m \boldsymbol{y}_k = \lambda_k \boldsymbol{y}_k$ and set $\epsilon_{\text{rel}}^{\text{KS}} = |\hat{\boldsymbol{b}}_m^H \boldsymbol{y}_k| / (|\lambda_k| \|\widehat{\mathcal{V}}_m \boldsymbol{y}_k\|_2)$
    if $\epsilon_{\text{rel}}^{\text{KS}} <$ `tol` **then**
      lock $(\lambda_k, \boldsymbol{y}_k)$ and continue
    else
      break
    end if
  end for
  $j \leftarrow k$
  choose p, truncate and set $\boldsymbol{v}_{p+1} \leftarrow \boldsymbol{v}_{m+1}$
  compute $\boldsymbol{b}_p$ and insert it into $\mathcal{H}_p$
end while

The Ritz value is considered as converged if $\epsilon_{\text{rel}}^{\text{KS}}$ is lower than the user-defined tolerance.

When a Ritz value is converged, it should no longer participate in the search for other eigenvalues. This is the goal of deflation. Supposing that λ_1 is the Ritz value with the largest magnitude of $\mathcal{T}_m$, the decomposition (II.69) can also be writen as:

$$\mathcal{C}\begin{bmatrix}\tilde{\boldsymbol{v}}_1 & \widetilde{\mathcal{V}}_{m-1}\end{bmatrix} = \begin{bmatrix}\tilde{\boldsymbol{v}}_1 & \widetilde{\mathcal{V}}_{m-1}\end{bmatrix}\begin{bmatrix}\lambda_1 & \star \\ 0 & \mathcal{T}_{m-1}\end{bmatrix} + \boldsymbol{v}_{m+1}\begin{bmatrix}b_1 & \hat{\boldsymbol{b}}_{m-1}^H\end{bmatrix}. \tag{II.75}$$

If λ_1 is converged, the coefficient b_1 can be set to 0, and all subsequent operations can be managed exclusively on the order-$(m-1)$ decomposition

$$\mathcal{C}\widetilde{\mathcal{V}}_{m-1} = \widetilde{\mathcal{V}}_{m-1}\mathcal{T}_{m-1} + \boldsymbol{v}_{m+1}\hat{\boldsymbol{b}}_{m-1}^H. \tag{II.76}$$

The next Ritz values are then tested and when the test fails, the loop breaks and the decomposition is truncated. Finally, if l Ritz values are converged, we get $\mathcal{C}\widetilde{\mathcal{V}}_j = \widetilde{\mathcal{V}}_j\mathcal{T}_j$, where the approximate eigenvalues are on the diagonal of $\mathcal{T}_m$, and the Ritz pairs of $\mathcal{T}_m$ are approximate eigenpairs of $\mathcal{C}$. Algorithm 2 gives a synthesized implementation of the KS method.

II.6.5 Shift-and-invert transformation

As stated earlier, the Schur matrix resulting from the KS basis expansion is sorted so that the Ritz values of largest magnitude are located at its top left corner. Other critera such as sorting by lowest magnitudes could have also been chosen, but the KS algorithm performs poorly when computing eigenvalues located in the interior of the spectrum [86]. In global modal stability, the KS algorithm is generally used in conjunction with the spectral shift-and-invert transformation [25, 20] already mentioned in previous sections. A shift $s \in \mathbb{C}$ is selected and the transformed eigenvalue problem involving the shit-and-invert operator $(\mathcal{C} - s\mathcal{I}_N)^{-1}$ is solved. Both operators $\mathcal{C}$ and $(\mathcal{C}-s\mathcal{I}_N)^{-1}$ have the same eigenvectors but the eigenvalues of largest magnitude of $(\mathcal{C}-s\mathcal{I}_N)^{-1}$ are directly linked to the eigenvalues of $\mathcal{C}$ closest to s. The concept is schematically illustrated in fig. II.7, where the spectrum of a fictive operator $\mathcal{C}$ is compared with that of the shift-and-invert operator. Only one unstable eigenvalue, equal to 1, is identified. The transformation, carried out with $s = 0.9$, clearly shows that the eigenvalues have been separated and the unstable eigenvalue has become the one with the largest magnitude. The counterpart of such a technique is the many

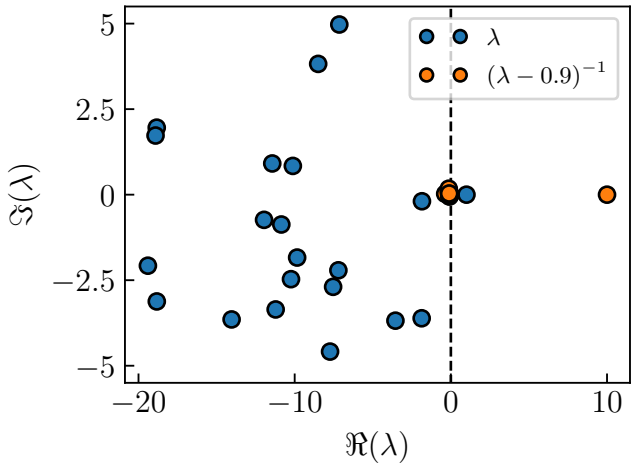


Figure II.7: Illustration of shift-and-invert transformation.

inversions of complex linear systems needed during the formation of the KS basis, because of the application of $(\mathcal{C} - s\mathcal{I}_N)^{-1}$ on the successive Krylov vectors.

II.7 Conclusion

This chapter provides the mathematical background for both the numerical discretization of operators in compressible CFD and the overall global stability analysis process. The first-order implicit backward Euler method is used to compute a fixed point of the Burgers equation with high precision, matching the theoretical solution. The stability analysis is then applied to the fixed point, which is shown to be unconditionally stable by modal analysis. The optimal gain computed with resolvent analysis shows a peak at zero frequency, resulting in a steady optimal forcing. The KS algorithm is also introduced in its complex arithmetic version for the sake of introduction.

The main challenge in scaling up global stability analysis for industrial configurations is dealing with extremely large linear systems that are too big to be factored using LU factorization due to memory constraints. The next chapter presents a strategy for conducting global stability analysis on large-scale systems using elsA. Based on previous work by Guilbert [51], inversions in elsA will be managed using the iterative GMRES algorithm [58]. To maintain HPC optimizations, the linear systems of interest will be reformulated in the real-equivalent K formulation.

Chapter III

Scaling up of the method to industrial cases

Contents

III.1 Introduction

Chapter II was devoted to the presentation of the general concepts of global stability analysis and gave an overview of the proposed method used in this thesis. Burgers equation has been

used to illustrate fixed point computation as well as linear, adjoint, and resolvent global stability analysis. This chapter gives details about the proposed method to deal with large CFD problems (formulated with Euler, NS, or RANS equations), as implemented in elsA software. Because of the number of degrees of freedom (DOF) of such configurations, the linear systems involved in the analyses may become too large to solve with direct methods. Iterative methods must be used instead, and we focus in particular on the Generalized Minimal Residual (GMRES) algorithm introduced by Saad in [58].

III.1.1 Scale up issues

We consider a 3-D RANS case ($d = 6$) with $N_c = 10 \times 10^6$ cells, corresponding to $N = 60 \times 10^6$ complex-valued DOF for the global stability problems. The computation is run on a computer cluster composed of CPUs like Cascade Lake processors (representative of ONERA's clusters), organized in nodes, as shown in fig. III.1. In this case, each node has 28 CPUs, with 4.57 GB/CPU and a total memory of 128 GB. Global stability problems require the inversion of complex operators, which have the same size as the Jacobian matrix, and the number of entries can be estimated as $\mathrm{NE}(\mathcal{J}) \simeq \text{stencil} \times d^2 \times N_c$. Using a second-order finite volume discretization[7], storage of the Jacobian matrix with complex floating-point numbers in double precision requires about 720 GB for the representative case.

We first focus on the memory footprint of the LU factorization of $\mathcal{J}$, written $\mathrm{LU}_{\mathcal{J}}$. Based on the work of L'Excellent [50], the number of entries of $\mathrm{LU}_{\mathcal{J}}$ may be estimated with the following relation:

$$\mathrm{NE}\left(\mathrm{LU}_{\mathcal{J}}\right) \simeq \left(\frac{\mathrm{NE}(\mathcal{J})}{7}\right)^{4/3}. \tag{III.1}$$

For the considered case, this results in about 500 TB of memory and would requires more than 100×10^3 CPU. Most of the existing calculators do not have this amount of memory available during the computation unless resorting to an overly large number of CPUs in parallel. As an indication, the maximum number of CPUs allowed at ONERA for one job is 4788, representing 171 nodes. Instead of such direct methods, the inversions might be addressed using preconditioned iterative Krylov methods (KM) [52]. With such approach, preconditioners can be based on ILU(k) or local LU factorization built from the Jacobian matrix partitioned on each CPU. If n_{KM} is the

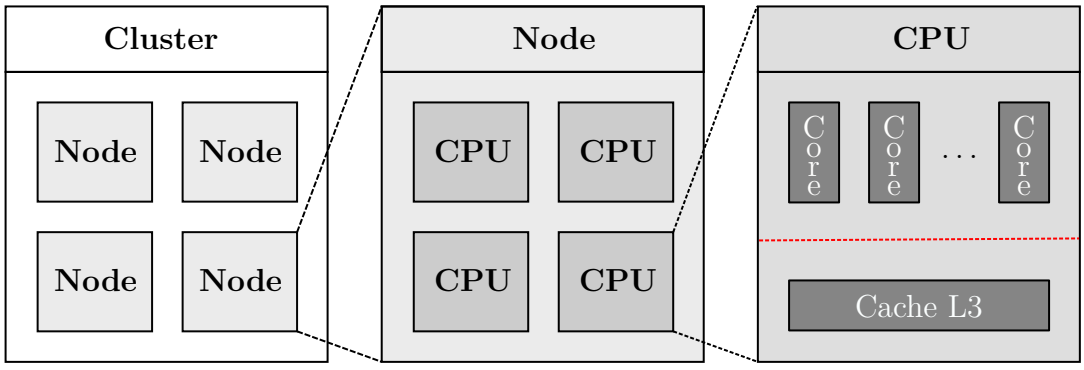


Figure III.1: Schematic representation of computing components.

[7] for which the approximate size of the numerical stencil is stencil $= 5^3$ in elsA.

Method	Memory	N_{CPU}
LU	$\backsim$ 500 TB	$>$ 100000
KM-ILU(k)	$\backsim$ 2.5 TB	$\backsim$ 500
KM-local LU	$\backsim$ 30 TB	$\backsim$ 6000

Table III.1: Comparison of memory costs between direct and preconditioned iterative methods for the considered case ($N = 60 \times 10^6$).

number of Krylov vectors involved in the computation and $\gamma(\text{k})$ a factor depending on the number of fill-in[8], the number of entry of the KM-ILU(k) method is approximated by:

$$\text{NE(KM-ILU(k))} \simeq \gamma(\text{k}) \times \text{NE}(\mathcal{J}) + n_{\text{KM}} \times N. \tag{III.2}$$

For the considered case, 2.5 TB of memory is needed and the computation can be launched on 500 CPUs. In this case, additional storages might be required if the convergence has to be improved using Restricted Additive Schwarz (RAS) methods [51]. Nevertheless, such a cost is much more affordable than that of direct methods. For particularly stiff cases, another approach based on local LU factorization can be used. The number of entries in KM-local LU method is approximated by:

$$\text{NE(KM-local LU)} \simeq N_{\text{CPU}}^{-1/3} \times \left(\frac{\text{NE}(\mathcal{J})}{7}\right)^{4/3} + n_{\text{KM}} \times N. \tag{III.3}$$

This represents about 30 TB of memory and would requires 6000 processors for the computation, considering the representative case. This is still being too large for ONERA's clusters, which is why hybrid preconditioners like Hybrid Local LU (HLU) factorization have been developed (see III.3.4 for more details). Table III.1 summarizes the costs of each method. Based on these observations, the preconditioned KM suit well to circumvent memory issues resulting from direct factorizations.

Another important issue that must be addressed is the complexity of the arithmetic used to construct and invert the operators of global stability, which is poorly vectorizable compared to real arithmetic. This prevents optimal use of HPC optimizations. Moreover, Guilbert's previous work [51] introduced powerful preconditioning tools to efficiently solve real linear systems arising in the fixed point computation (see II.31) in the context of compressible CFD. Taking these considerations into account, we choose to reformulate the non-vectorizable complex linear systems of global stability into a real-equivalent formulation that preserves HPC optimizations and reuses the existing preconditioning tools developed in [51]. This is achieved in this thesis by using the K formulation introduced by Day and Heroux [60] that preserves the sparsity pattern of the original systems. Some of its properties and the K-formulated KS algorithm that has been implemented in elsA are detailed in this chapter.

III.1.2 Numerical set-up of laminar cylinder flow

Each concept will be illustrated using the canonical example of laminar cylinder flow. This example is used to validate the proposed algorithms as it has been widely documented (see [29, 30, 87]). The numerical domain, shown in Fig. III.2, is composed of 34560 unstructured hexahedral cells in a C-grid. The Mach number, $\text{M} = \|\boldsymbol{u}\|_2/a$, is introduced, where a is the speed of sound, and the physical parameters are given in Table III.2. The problem is dimensionless using the diameter

[8] a typical three levels of fill-in, resulting roughly in $\gamma(3) = 3$ and $n_{\text{KM}} = 120$ consitute good parameters.

Free-stream Mach number	$\mathrm{M}_\infty = 0.1$
Free-stream stagnation temperature	$T_\infty = 300$ K
Free-stream velocity	$u_\infty = 34.7\ \mathrm{m\,s^{-1}}$
Free-stream density	$\rho_\infty = 1.177\ \mathrm{kg\,m^{-3}}$
Reynolds number	$\mathrm{Re} \in [40, 80]$

Table III.2: Flow parameters of the 2-D laminar cylinder.

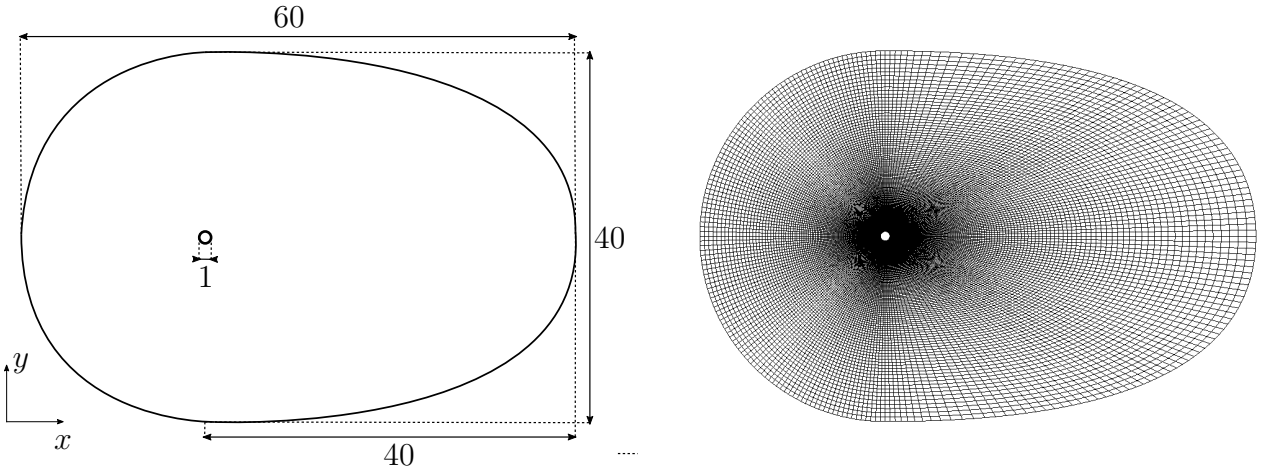


Figure III.2: Schematic view of the cylinder mesh (left) and computational grid (right).

of the cylinder as the reference length, L, and the free-stream density, velocity, and temperature. The other physical quantities are calculated using the following expressions:

$$a_\infty = \frac{1}{\mathrm{M}_\infty}, \tag{III.4}$$

$$p_\infty = \frac{1}{\gamma \mathrm{M}_\infty}, \tag{III.5}$$

$$e_\infty = \frac{1}{\gamma(\gamma-1)\mathrm{M}_\infty^2} = c_v, \tag{III.6}$$

$$R = \frac{1}{\gamma \mathrm{M}_\infty^2}, \tag{III.7}$$

$$\mu_\infty = \frac{1}{\mathrm{Re}_\infty}. \tag{III.8}$$

A second-order Roe scheme with MUSCL reconstruction and no limiter is used for convective fluxes, while diffusive fluxes are computed with a $5p_{\mathrm{cor}}$ scheme (see. III.3.2). The boundary conditions are no-slip at the cylinder wall, non-reflective at the farfield, and symmetry conditions on each side of the domain to produce a 2-D-like computation. Despite being inherently 2-D, this case will be treated as a 3-D computation by the elsA code.

III.2 Equations of compressible CFD

The equations solved in elsA are based on the compressible NS equations, which describe the behavior of a fluid flow in terms of mass, momentum, and energy conservation. This section

provides the mathematical formulation of these equations in a general form for both laminar and turbulent flows.

III.2.1 Navier-Stokes equations

A reference frame where the space-time coordinate of each point is identified by (x, y, z, t) is considered. The state vector of the conservative variables of the flow, $\boldsymbol{w} = \begin{pmatrix} \rho & \rho u_x & \rho u_y & \rho u_z & \rho E \end{pmatrix}^T$, involves the density field ρ, the absolute velocity field $\boldsymbol{u} = \begin{pmatrix} u_x & u_y & u_z \end{pmatrix}^T$ and the total energy per unit of mass E, which is the sum of the internal energy e and the kinetic energy $\|\boldsymbol{u}\|_2^2/2$. The set of NS equations governs the evolution in space and time of $\boldsymbol{w}$ and writes:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \rho \boldsymbol{u} = 0 \\ \dfrac{\partial \rho \boldsymbol{u}}{\partial t} + \overline{\overline{\nabla}} \left(\rho \boldsymbol{u} \otimes \boldsymbol{u} + p \mathcal{I}_3 - \overline{\overline{\tau}} \right) = 0 \\ \dfrac{\partial \rho E}{\partial t} + \boldsymbol{\nabla} \cdot \left(\rho E \boldsymbol{u} + p \boldsymbol{u} - \overline{\overline{\tau}} \cdot \boldsymbol{u} + \boldsymbol{q} \right) = 0. \end{cases} \tag{III.9}$$

with p, $\overline{\overline{\tau}}$ and $\boldsymbol{q}$ respectively standing for the static pressure, the viscous stress tensor and the heat flux vector. Using the notation of II.2.1, the convective and diffusive parts of (III.9) are the following:

$$\boldsymbol{f}_c = \begin{pmatrix} \rho u_x & \rho u_2 & \rho u_3 \\ \rho u_x^2 + p & \rho u_x u_y & \rho u_x u_z \\ \rho u_x u_y & \rho u_y^2 + p & \rho u_y u_z \\ \rho u_x u_z & \rho u_y u_z & \rho u_z^2 + p \\ \rho u_x \left(E + p \right) & \rho u_y \left(E + p \right) & \rho u_z \left(E + p \right) \end{pmatrix}, \tag{III.10}$$

$$\boldsymbol{f}_d = \begin{pmatrix} 0 & 0 & 0 \\ \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \\ u_x \tau_{xx} + u_y \tau_{xy} + u_z \tau_{xz} - q_1 & u_x \tau_{xy} + u_y \tau_{yy} + u_z \tau_{yz} - q_2 & u_x \tau_{xz} + u_y \tau_{yz} + u_z \tau_{zz} - q_3 \end{pmatrix}. \tag{III.11}$$

Expressions of $\overline{\overline{\tau}}$ and $\boldsymbol{q}$ have to be explicited with behavior laws. The fluid is assumed to be Newtonian, meaning that the tensor $\overline{\overline{\tau}}$ which models the internal frictions in the fluid can be expressed as:

$$\overline{\overline{\tau}} = \mu \left[2 \left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T \right) - \frac{2}{3} \boldsymbol{\nabla} \cdot \boldsymbol{u} \mathcal{I}_3 \right], \tag{III.12}$$

with μ is the dynamic viscosity of the fluid. Fourier law is used to describe the heat flux vector:

$$\boldsymbol{q} = -\kappa \boldsymbol{\nabla} T, \tag{III.13}$$

with κ the thermal conductivity coefficient and T the absolute static temperature.

From now, we have a system of five equations for the seven unknowns ρ, u_x, u_y, u_z, E, p and T. This means that two additional relations must be provided to close the problem. In this work, only perfect gas in thermodynamic equilibrium, not chemically reacting and with constant isochoric and

isobaric specific heats c_v and c_p, are considered. The specific heat ratio is $\gamma = c_p/c_v = 1.4$. Such assumptions lead to the following expressions for e and p:

$$e = c_v T \tag{III.14}$$

$$p = \rho R T, \tag{III.15}$$

with R the ratio of the perfect gas constant to the molar mass of the fluid, set at 287.053 $\mathrm{J\,kg^{-1}\,K^{-1}}$ for air.

For perfect gas, the viscosity coefficient μ is considered to be independent of the pressure but strongly depends on the temperature. To this end, the Sutherland law [88] is used:

$$\mu = \mu_S \sqrt{\frac{T}{T_{\mathrm{ref}}}} \frac{1 + T_S/T_{\mathrm{ref}}}{1 + T_S/T}, \tag{III.16}$$

with $\mu_S = 1.711 \times 10^{-5}$ $\mathrm{N\,m^{-2}\,s}$ the dynamic viscosity coefficient of air at temperature $T_{\mathrm{ref}} = 273$ K, and $T_S = 110.4$ K the Sutherland temperature. As we mainly deal with gas, the thermal coefficient κ is proportional to μ according to the law:

$$\kappa = c_p \frac{\mu}{\mathrm{Pr}}, \tag{III.17}$$

where Pr is the Prandtl number [89], assumed to be constant and equal to 0.72.

III.2.2 Reynolds Averaged Navier-Stokes equations

In this work, turbulence is modeled using a statistical approach in which each field f is decomposed into a mean field $\widehat{f}$, and a fluctuating one f', so that $f(\boldsymbol{x}, t) = \widehat{f} + f'$. When f slowly varies in time, it is assumed to be ergodic [90], and its average value can be obtained from the temporal average on a certain period of time T:

$$\widehat{f} = \frac{1}{T} \int_{-T/2}^{T/2} f(\boldsymbol{x}, t + t') \mathrm{d}t'. \tag{III.18}$$

When f has a variable density, the use of Favre average [91] is prefered, giving the decomposition $f(\boldsymbol{x}, t) = \widetilde{f} + f''$ in which the mean field writes:

$$\widetilde{f} = \frac{\widehat{\rho f}}{\widehat{\rho}}. \tag{III.19}$$

In NS equations, the temporal average is used for variables ρ, p, $\overline{\overline{\tau}}$ and $\boldsymbol{q}$ while the Favre average is used for $\boldsymbol{u}$ and E. This results into the following set of RANS equations:

$$\begin{cases} \dfrac{\partial \widehat{\rho}}{\partial t} + \boldsymbol{\nabla} \cdot \widehat{\rho} \widetilde{\boldsymbol{u}} = 0 \\ \dfrac{\partial \widehat{\rho} \widetilde{\boldsymbol{u}}}{\partial t} + \overline{\overline{\nabla}} \left(\widehat{\rho} \widetilde{\boldsymbol{u}} \otimes \widetilde{\boldsymbol{u}} + \widehat{p} \mathcal{I}_3 - \widehat{\overline{\overline{\tau}}} - \overline{\overline{\tau}}_r \right) = 0 \\ \dfrac{\partial \widehat{\rho} \widetilde{E}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\widehat{\rho} \left(\widetilde{E} + k \right) \widetilde{\boldsymbol{u}} + \widehat{p} \widetilde{\boldsymbol{u}} - \widehat{\overline{\overline{\tau}}} \cdot \widetilde{\boldsymbol{u}} + \widehat{\boldsymbol{q}} + \boldsymbol{q}_t \right) = 0. \end{cases} \tag{III.20}$$

Three new terms appear in this last system: the kinetic energy of the turbulent movement $k = \widehat{\rho \boldsymbol{u}''^2}/2\widehat{\rho}$, the Reynolds tensor $\overline{\overline{\tau}}_r = -\widehat{\rho \boldsymbol{u}'' \otimes \boldsymbol{u}''}$ and the turbulent heat flux $\boldsymbol{q}_t = \widehat{\rho \boldsymbol{u}'' h''}$.

The problem is closed providing a turbulence model, which we choose to follow the Boussinesq hypothesis. It states, by analogy with momentum transfers resulting in molecular viscosity μ, that momentum transfers resulting from turbulent eddies are modeled with an eddy viscosity μ_t. The Reynolds stress tensor and the turbulent heat flux vector can thus be recast as:

$$\widetilde{\overline{\tau}}_r = -\frac{2}{3}\left(\widehat{\rho}k + \mu_t \boldsymbol{\nabla}\cdot\widetilde{\boldsymbol{u}}\right)\mathcal{I}_3 + 2\mu_t\left(\nabla\widetilde{\boldsymbol{u}} + \nabla\widetilde{\boldsymbol{u}}^T\right) \tag{III.21}$$

$$\boldsymbol{q}_t = -\frac{c_p\mu_t}{\mathrm{Pr}_t}\boldsymbol{\nabla}T, \tag{III.22}$$

where Pr_t is the turbulent Prandtl number, set at 0.9 for air. With such hypothesis, we can therefore introduce the following starred quantities:

$$\begin{cases} E^\star = \widehat{E} + k \\ p^\star = \widehat{p} + \dfrac{2}{3}\widehat{\rho}k \\ \widetilde{\overline{\tau}}^\star = \left(1 + \dfrac{\mu_t}{\mu}\right)\widetilde{\overline{\tau}} \\ \boldsymbol{q}^\star = \left(1 + \dfrac{\mu_t}{\mu}\dfrac{\mathrm{Pr}}{\mathrm{Pr}_t}\right)\widehat{\boldsymbol{q}} \end{cases} \tag{III.23}$$

and replace in equation (III.20), their corresponding non-starred quantities, leading to the RANS equations in a similar form as the NS equations.

The one-equation Spalart-Allmaras (SA) model [92] is commonly used in compressible RANS simulations for aerodynamic. It has been designed for aerodynamic applications involving wall-bounded flows. It solves a transport equation for a kinematic-like viscosity coefficient $\widetilde{\nu}$ that is linked to the turbulent dynamic viscosity *via*:

$$\mu_t = \widehat{\rho}\widetilde{\nu}f_{v1}, \tag{III.24}$$

where f_{v1} is the following damping function:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}}, \tag{III.25}$$

with $\chi = \widetilde{\nu}/\nu$, ν being the molecular kinematic viscosity ($\nu = \mu/\rho$), and $c_{v1} = 7.1$. The conservative form of the added transport equation governs the evolution of the quantity $\widehat{\rho}\widetilde{\nu}$ and writes:

$$\frac{\partial\widehat{\rho}\widetilde{\nu}}{\partial t} + \boldsymbol{\nabla}\cdot\left(\widehat{\rho}\widetilde{\nu}\widetilde{\boldsymbol{u}} - \frac{1}{\sigma_{SA}}\widehat{\rho}\left(\nu + \widetilde{\nu}\right)\nabla\widetilde{\nu}\right) = \underbrace{c_{b1}\left(1 - f_{t2}\right)\widetilde{S}\widehat{\rho}\widetilde{\nu} + \frac{c_{b2}}{\sigma_{SA}}\nabla\widetilde{\nu}\cdot\nabla\widehat{\rho}\widetilde{\nu}}_{\text{production}} - \underbrace{\widehat{\rho}\left(c_{w1}f_w - \frac{c_{b1}}{\kappa_{SA}^2}f_{t2}\right)\left(\frac{\widetilde{\nu}}{d_w}\right)^2}_{\text{destruction}}. \tag{III.26}$$

Unlike the other considered equations, this one has a source term on its right-hand side representing a balance between a production and a destruction terms. The various quantities of the model are

then provided by:

$$f_{t2} = c_{t3} e^{-c_{t4}\chi^2}, \tag{III.27}$$

$$f_w = g\left(\frac{1+c_{w3}^6}{g^6+c_{w3}^6}\right)^{1/6}, \tag{III.28}$$

$$g = r + c_{w2}\left(r^6 - r\right), \qquad r = \min\left(\frac{\widetilde{\nu}}{\widetilde{S}\kappa_{SA}^2 d_w^2}, 10\right) \tag{III.29}$$

$$\widetilde{S} = S + \frac{\widetilde{\nu}}{\kappa_{SA}^2 d_w^2} f_{v2}, \qquad S = \sqrt{2\overline{\overline{\Omega}} : \overline{\overline{\Omega}}}, \qquad \overline{\overline{\Omega}} = \frac{1}{2}\left(\nabla\widetilde{\boldsymbol{u}} - \nabla\widetilde{\boldsymbol{u}}^T\right), \tag{III.30}$$

$$f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \tag{III.31}$$

where d_w stands for the distance from the field point to the nearest wall. The constants of the model are:

$$\sigma_{SA} = \frac{2}{3}, \qquad \kappa_{SA} = 0.41 \tag{III.32}$$

$$c_{b1} = 0.1355, \qquad c_{b2} = 0.622, \tag{III.33}$$

$$c_{w1} = \frac{c_{b1}}{\kappa_{SA}^2} + \frac{1+c_{b2}}{\sigma_{SA}}, \tag{III.34}$$

$$c_{w2} = 0.3, \qquad c_{w3} = 2, \tag{III.35}$$

$$c_{t3} = 1.2, \qquad c_{t4} = 0.5. \tag{III.36}$$

$$\tag{III.37}$$

This original SA model requires $\widetilde{\nu}$ to be always positive, which is ensured by taking its absolute value. Since this function cann't be differentiated at zero, we have preferred to use the negative SA model [93] which is a slight modification of the standard model. When $\widetilde{\nu} \geq 0$, equation (III.26) is solved. However, when it becomes negative, the following equation is solved instead:

$$\frac{\partial\widehat{\rho}\widetilde{\nu}}{\partial t} + \boldsymbol{\nabla}\cdot\left(\widehat{\rho}\widetilde{\nu}\widetilde{\boldsymbol{u}} - \frac{1}{\sigma_{SA}}\widehat{\rho}\left(\nu+\widetilde{\nu}f_n\right)\nabla\widetilde{\nu}\right) = c_{b1}\left(1-c_{t3}\right)S\widehat{\rho}\widetilde{\nu} + \frac{c_{b2}}{\sigma_{SA}}\nabla\widetilde{\nu}\cdot\nabla\widehat{\rho}\widetilde{\nu} + \widehat{\rho}c_{w1}f_w\left(\frac{\widetilde{\nu}}{d_w}\right)^2, \tag{III.38}$$

with

$$f_n = \frac{c_{n1}+\chi^3}{c_{n+1}-\chi^3}, \qquad c_{n1} = 16. \tag{III.39}$$

Next section first provides a brief overview of the numerical discretization techniques utilized in elsA. It then describes the two fixed point computation methods used and offers insight on preconditioning and multicoloring.

III.3 elsA CFD solver

The numerical methods used in elsA for computing steady solutions of the discretized NS equations are described below. The focus is on implicit methods that solve (approximately or exactly) the implicit backward Euler (see equation (II.31)), which are preferred over explicit methods due to their unconditional stability and ability to handle larger time-steps. However, these methods require solving linear systems, which can be computationally expensive.

In some cases, such as when the initial field is uniform, using an exact inversion method at each step can be counterproductive, especially during the first iterations. To address this issue, less efficient but faster methods, like matrix-free ones [94] or the so-called LU-Symmetric Successive Overrelaxation Scheme (LU-SSOR) [95] that solves an approximation of the linear system can be used to refine the initial uniform field towards a fixed point, reducing the need for costly matrix factorizations during the early iterations. After this, a more accurate computation is performed using the GMRES [58] algorithm to invert the exact linear systems and decrease the residual below 10×10^{-10}. These methods will be described in detail in the following section.

III.3.1 Unstructured computation

In this thesis, only unstructured computational domain are considered. The mesh is contained in a CGNS file (`https://www.cgns.org`) and, as discussed by Bourasseau in [96], elsA, for an unstructured block, performs the calculations by iterating through all the faces of the mesh. The faces are ordered by starting with the internal faces and ending with the boundary faces, grouping both the connection faces and the physical boundary faces to which a physical boundary condition is imposed. However, since the boundary faces are not ordered by physical or connection conditions, elsA generates two lists grouping, for one, all physical boundary faces and, for the other, all connection faces. Each of these lists contains lists whose length is equal to the number of faces of the boundary condition it represents. It is therefore necessary to create, for each list, an indirection table to go from the local numbering of the face corresponding to its position in the list to the global numbering of that face. The correspondence between the faces and the cells is handled by building two indirection tables to determine for each face its right and left cells, knowing that the definition of right and left is imposed by the normal to the face, itself imposed by the ordering of the nodes that make up this face.

III.3.2 Numerical discretization

To describe the numerical discretization as employed in elsA, we rely on the general form of the governing equations of compressible CFD and their discretization with finite volumes detailed in II.12. We work on a flow domain Ω that is partitioned into an unstructured grid containing N_c cells, as illustrated in fig. III.3. The system (III.9) is integrated over each cell Ω_k, and the finite volume method is used to compute the solution as in equation (II.12). Approximate Riemann solver like Roe's (see equation (II.13)) or HLLC [63] are used to calculate the function $\boldsymbol{h}_c$ at each interface.

As for equation (II.20), the diffusive fluxes are computed with a discretization centered at the faces. The main difficulty results from the computation of the gradients of the primitive variables. The $5p$ method (for five-point stencil) is a simple arithmetic mean of gradients computed at cell centers of both sides of the face I_j. Referring to the scheme of fig. III.3, it writes:

$$\nabla \boldsymbol{p}_{I_j}^{5p} = \frac{1}{2} \left(\nabla \boldsymbol{p}|_k + \nabla \boldsymbol{p}|_l \right). \tag{III.40}$$

However, this expression is modified to get a so-called $5p_\mathrm{cor}$ method to improve the discretization quality and avoid oscillations. In the latter, the gradient is computed as if the control volume was centered on I_j and composed of cells Ω_k and Ω_l, providing the following new expression:

$$\nabla \boldsymbol{p}_{I_j}^{5p_\mathrm{cor}} = \nabla \boldsymbol{p}_{I_j}^{5p} - \left(\nabla \boldsymbol{p}_{I_j}^{5p} \cdot \frac{\boldsymbol{x}_k - \boldsymbol{x}_l}{\|\boldsymbol{x}_k - \boldsymbol{x}_l\|_2} - \frac{\boldsymbol{p}_k - \boldsymbol{p}_l}{\|\boldsymbol{x}_k - \boldsymbol{x}_l\|_2} \right) \frac{\boldsymbol{x}_k - \boldsymbol{x}_l}{\|\boldsymbol{x}_k - \boldsymbol{x}_l\|_2}. \tag{III.41}$$

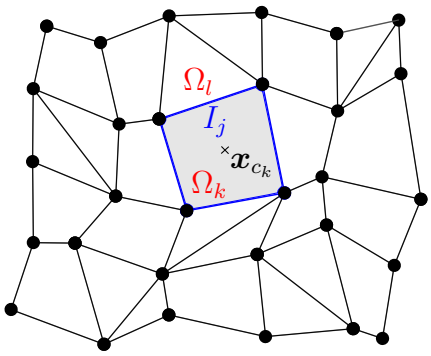


Figure III.3: Example of unstructured mesh. Grey zone highlights the cell Ω_k of center $\boldsymbol{x}_{c_k}$. Interface I_j is the fronter between Ω_k and Ω_l.

For RANS configurations, the turbulent source terms are discretized using a second-order integration method.

III.3.3 LU-SGS method for fixed point computation

Iterative methods such as matrix-free, LU-SGS, or LU-SSOR [94, 97] are derived by approximating the implicit operator $\mathcal{A} = \Delta t^{-1}\mathcal{I}_N + \mathcal{J}$ to solve the system in equation (II.31). This system is constructed by discretizing the convective component using the Rusanov flux and finding the spectral radius of the resulting operator, which leads to a simple tridiagonal operator $\widetilde{\mathcal{A}}$. During each nonlinear iteration, a sub-iteration process is used to calculate $\Delta \boldsymbol{W}^n$ and the system is solved using the SSOR method [52, 95]. As pointed out in [98], the best results are achieved when the relaxation parameter is set to 1, leading to the symmetric Gauss-Seidel (SGS) method. Considering $\mathcal{L}_l$, $\mathcal{D}_l$ and $\mathcal{U}_l$ being the strictly lower part, the diagonal part, and the strictly upper part of $\widetilde{\mathcal{A}}(\boldsymbol{W}^{n,l})$, the LU-SGS method is obtained by factorizing $\widetilde{\mathcal{A}}(\boldsymbol{W}^{n,l})$ as:

$$\widetilde{\mathcal{A}}(\boldsymbol{W}^{n,l}) = \mathcal{L}_l + \mathcal{D}_l + \mathcal{U}_l \simeq (\mathcal{L}_l + \mathcal{D}_l)\,\mathcal{D}_l^{-1}\,(\mathcal{U}_l + \mathcal{D}_l)\,, \tag{III.42}$$

where $\mathcal{L}_l\mathcal{D}_l^{-1}\mathcal{U}_l$ is the error committed relatively to the original factorization. The resolution is carried out by the three following steps:

$$\begin{cases} (\mathcal{U}_l + \mathcal{D}_l)\,\Delta \boldsymbol{W}^{l+1/3} = \widetilde{\boldsymbol{R}}(\boldsymbol{W}^{n,l}) \\ \mathcal{D}_l \Delta \boldsymbol{W}^{n,l+2/3} = \Delta \boldsymbol{W}^{n,l+1/3} \\ (\mathcal{L}_l + \mathcal{D}_l)\,\Delta \boldsymbol{W}^{n,l+1} = \Delta \boldsymbol{W}^{n,l+2/3}, \end{cases} \tag{III.43}$$

where $\widetilde{\boldsymbol{R}}$ stands for the residual associated with the approximated implicit system.

Inversions of the approximate tridiagonal operator at each nonlinear iteration is much less expansive than that of the exact operator, resulting in a much faster computation. It is especially fast for low CFL numbers up to CFL = 100, when the goal is to filter coarse unsteadiness, i.e. decreasing the residual $\|\boldsymbol{R}(\boldsymbol{W}^n)\|_2$ to below 10^{-3} to 10^{-4} (see [51]). However, when the goal is to reach convergence around machine precision, the CFL number must increase to recover the Newton method that achieves a quadratic convergence, and the exact linear system must be solved.

III.3.4 GMRES algorithm for fixed point computation

To increase the accuracy of a computed a fixed point of the governing equations, the implicit backward Euler scheme (II.31) with adaptative CFL must be solved at each nonlinear iteration.

This requires the multiple iversions of $\mathcal{A}$ during the computation. To this end, given the sizes of 3-D configurations, the use of iterative methods instead of direct ones is mandatory, due to memory limitations. Additionally, as noted by Guilbert in [51], due to the non-symmetrical, non-normal, and potentially non-diagonal dominant nature of the operator, the focus is placed on the GMRES algorithm introduced by Saad & Schultz in [58].

Derivation of GMRES algorithm. Considering a non-singular matrix $\mathcal{A} \in \mathbb{R}^N \times \mathbb{R}^N$, and a right-hand side vector $\boldsymbol{b} \in \mathbb{R}^N$, we focus on the resolution of the following real-valued linear system:

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \tag{III.44}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the unknown. The GMRES algorithm computes a solution to equation (III.44) by projecting $\boldsymbol{x}$ onto a given Krylov space. Until convergence, the Krylov basis is continually improved. The steps of the GMRES algorithm are described hereafter.

Given an initial approximation $\boldsymbol{x}_0$ of $\boldsymbol{x}$, the initial residual $\boldsymbol{r}_0$ is defined as $\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{A}\boldsymbol{x}_0$. The first step is to produce an orthonormal basis of the Krylov space $K_m(\mathcal{A}, \boldsymbol{r}_0)$ associated with $\boldsymbol{r}_0$, denoted as K_m in the following. As in II.6.1, this is achieved with the Arnoldi algorithm 1 which produces the following Krylov decomposition:

$$\mathcal{A}\mathcal{V}_m = \mathcal{V}_m \mathcal{H}_m + \boldsymbol{v}_{m+1} \widehat{h}_{m,m+1}^T, \tag{III.45}$$

which can be written in a more compact form as:

$$\mathcal{A}\mathcal{V}_m = \mathcal{V}_{m+1} \widetilde{\mathcal{H}}_m. \tag{III.46}$$

Here, $\mathcal{V}_{m+1}$ and $\widetilde{\mathcal{H}}_m$ are N-by-$(m+1)$ and $(m+1)$-by-m real matrices, respectively, and $\beta = \|\boldsymbol{r}_0\|_2$. The Arnoldi algorithm is initialized with the unitary vector $\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$. The matrix-vector products required to build the decomposition are computed using AD, which is optimal to benefit from all HPC optimizations (see II.4).

The next step is to look for a vector $\boldsymbol{x}_m \in \boldsymbol{x}_0 + K_m$ that minimizes $\|\boldsymbol{r}_m\|_2$. As $(\boldsymbol{v}_k, 0 \leq k \leq m-1)$ is an orthonormal basis of K_m, there exists a vector $\boldsymbol{y}_m \in \mathbb{R}^m$ verifying $\boldsymbol{x}_m = \boldsymbol{x}_0 + \mathcal{V}_m \boldsymbol{y}_m$. The residual $\boldsymbol{r}_m = \boldsymbol{b} - \mathcal{A}\boldsymbol{x}_m$ can thus be recast as:

$$\begin{aligned} \boldsymbol{b} - \mathcal{A}\boldsymbol{x}_m &= \boldsymbol{r}_0 - \mathcal{A}\mathcal{V}_m \boldsymbol{y}_m \\ &= \boldsymbol{r}_0 - \mathcal{V}_{m+1} \widetilde{\mathcal{H}}_m \boldsymbol{y}_m \\ &= \mathcal{V}_{m+1} \left(\mathcal{V}_{m+1}^T \beta \boldsymbol{v}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}_m \right), \end{aligned} \tag{III.47}$$

which becomes, after taking the L^2-norm and using the orthogonality of the Krylov vectors:

$$\|\boldsymbol{b} - \mathcal{A}\boldsymbol{x}_m\|_2 = \|\beta \boldsymbol{e}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}_m\|_2. \tag{III.48}$$

Minimizing the left-hand side of the latter equality is hence equivalent to solve following size m least-square problem:

$$\boldsymbol{x}_m = \text{argmin}_{\boldsymbol{y}} \|\beta \boldsymbol{e}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}\|_2. \tag{III.49}$$

This can be efficiently achieved using the Givens rotations [99]. This technique has the additional advantage of computing the relative residual $\epsilon_{\text{rel}}^{\text{GMRES}} = \|\boldsymbol{r}_m\|/\beta$ implicitly, i.e. without requiring an explicit calculation of the approximate solution $\boldsymbol{x}_m$. The approximation is computed once when the relative residual becomes lower than the user-specified tolerance.

Algorithm 3 Basic GMRES

Input: $\mathcal{A}$, $\boldsymbol{b}$, $\boldsymbol{x}_0$, `tol` and m
Output: $\boldsymbol{x}_m$ minimizing $\|\boldsymbol{r}_m\|_2$ with actualized m

$\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{A}\boldsymbol{w}_0$
$\beta = \|\boldsymbol{r}_0\|_2$
$\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$
for $k \in [\![1, m]\!]$ **do**
  $\boldsymbol{v}_{k+1} \leftarrow \mathcal{A}\boldsymbol{v}_k$
  for $l \in [\![1, k]\!]$ **do**
    $h_{lk} \leftarrow \langle \mathcal{A}\boldsymbol{v}_k, \boldsymbol{v}_l \rangle$
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{w}_k - \boldsymbol{v}_l h_{lk}$
  end for
  $h_{k+1,k} \leftarrow \|\boldsymbol{v}_{k+1}\|_2$
  compute the relative residual $\epsilon_{\text{rel}}^{\text{GMRES}}$ with Givens rotations
  if $h_{k+1,k} = 0$ or $\epsilon_{\text{rel}}^{\text{GMRES}} <$ `tol` **then**
    $m \leftarrow k$
    break main loop
  else
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{v}_{k+1}/h_{k+1,k}$
  end if
end for
$\boldsymbol{y}_m = \text{argmin}_{\boldsymbol{y}} \|\beta \boldsymbol{e}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}\|_2$
$\boldsymbol{x}_m = \boldsymbol{x}_0 + \mathcal{V}_m \boldsymbol{y}_m$

When using the GMRES algorithm to compute $\Delta \boldsymbol{W}^n$, it can be counterproductive to set the tolerance too low, as it may prevent the adaptive CFL from increasing as it should. Through trial and error, a tolerance of 10^{-4} has proven to be a good compromise. In practice, this algorithm, in its current state, is not expected to converge well, as the operator $\mathcal{A}$ is generally ill-conditioned. In fact, the solvability of system (III.44) depends on the condition number of $\mathcal{A}$, which, given a convenient norm $\|\cdot\|$, is defined as:

$$\kappa(\mathcal{A}) = \|\mathcal{A}^{-1}\|\|\mathcal{A}\|. \tag{III.50}$$

In our systems of interest, the non-normality of $\mathcal{A}$ is responsible for increasing $\kappa(\mathcal{A})$, and the larger it is, the more complicated the resolution is. This is why we need to slightly modify this system to incorporate operators called preconditioners, which aim at lowering $\kappa(\mathcal{A})$.

Preconditioned GMRES. Preconditioning is a modification of a linear system that makes it easier to solve using an iterative method. Although not guaranteed, the resulting system may need fewer steps to converge. Concretely, the goal is to find an operator $\mathcal{M}$ such that $\mathcal{M} \simeq \mathcal{A}$ and apply it on the left (equation (III.51)) or right (equation (III.52)) of the system (III.44):

$$\begin{cases} \mathcal{M}^{-1}\mathcal{A}\boldsymbol{x} = \mathcal{M}^{-1}\boldsymbol{b} & \text{(III.51)} \\ \mathcal{A}\mathcal{M}^{-1}\mathcal{M}\boldsymbol{x} = \boldsymbol{b}. & \text{(III.52)} \end{cases}$$

As discussed in [100], the asymptotic behavior of an iterative method is the same whether we use left or right-preconditioning. However, in the left-handed case, the preconditioning also applies

Algorithm 4 Right-preconditioned GMRES

Input: $\mathcal{A}$, $\mathcal{M}$, $\boldsymbol{b}$, $\boldsymbol{x}_0$, `tol` and m
Output: $\boldsymbol{x}_m$ minimizing $\|\boldsymbol{r}_m\|_2$ with actualized m

$\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{A}\boldsymbol{w}_0$
$\beta = \|\boldsymbol{r}_0\|_2$
$\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$
for $k \in [\![1, m]\!]$ **do**
  $\boldsymbol{v}_{k+1} \leftarrow \mathcal{A}\mathcal{M}^{-1}\boldsymbol{v}_k$
  for $l \in [\![1, k]\!]$ **do**
    $h_{lk} \leftarrow \langle \mathcal{A}\boldsymbol{v}_k, \boldsymbol{v}_l \rangle$
    $\boldsymbol{v}_{j+1} \leftarrow \boldsymbol{w}_j - \boldsymbol{v}_k h_{kj}$
  end for
  $h_{k+1,k} \leftarrow \|\boldsymbol{v}_{k+1}\|_2$
  compute the relative residual $\epsilon_{\text{rel}}^{\text{GMRES}}$ with Givens rotations
  if $h_{k+1,k} = 0$ or $\epsilon_{\text{rel}}^{\text{GMRES}} <$ `tol` **then**
    $m \leftarrow k$
    break main loop
  else
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{v}_{k+1}/h_{k+1,k}$
  end if
end for
$\boldsymbol{y}_m = \text{argmin}_{\boldsymbol{y}} \|\beta \boldsymbol{e}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}\|_2$
$\boldsymbol{x}_m = \boldsymbol{x}_0 + \mathcal{M}^{-1}\mathcal{V}_m \boldsymbol{y}_m$

to the residual making it different from that of the original problem. This might result in significant differences between the original residual and the left-preconditioned one, as the calculation proceeds. Instead, we choose to work with right-preconditioning as it leaves the residual vector unchanged and often shows better convergence:

$$\boldsymbol{r} = \boldsymbol{b} - \mathcal{A}\boldsymbol{x} = \boldsymbol{b} - \mathcal{A}\mathcal{M}^{-1}\mathcal{M}\boldsymbol{x}. \tag{III.53}$$

With right-preconditioning, algorithm 3 becomes algorithm 4. In this version of GMRES, the preconditioner remains the same at each iteration, and the solution is recovered at the end of the process by pre-multiplying $\mathcal{V}_m \boldsymbol{y}_m$ with $\mathcal{M}^{-1}$. Another version that is popular is CFD is the so-called Flexible GMRES (FGMRES) algorithm [101] in which the preconditioner is updated at each step. To this end, two GMRES are nested one into the other. At each step of the external GMRES, an internal one computes the intermediate vector $\boldsymbol{u}_{k+1}$ with:

$$\boldsymbol{u}_{k+1} = \mathcal{M}_k^{-1}\boldsymbol{v}_k, \tag{III.54}$$

and the new participant to the external Krylov basis writes $\boldsymbol{v}_{j+1} = \mathcal{A}\boldsymbol{u}_{j+1}$. The set of internal Krylov vectors are stored in the basis $\mathcal{U}_m$ to recover the solution at the end of the main loop:

$$\boldsymbol{x}_m = \boldsymbol{x}_0 + \mathcal{U}_m \boldsymbol{y}_m. \tag{III.55}$$

Algorithm 5 presents an implementation of FGMRES. This variation usually shows significant improvement in terms of convergence speed compared to the basic right-preconditioned GMRES. However, it requires the storage of a second Krylov basis. The dimensions of both internal and

Algorithm 5 FGMRES

Input: $\mathcal{A}$, $\mathcal{M}$, $\boldsymbol{b}$, $\boldsymbol{x}_0$, `tol` and m
Output: $\boldsymbol{x}_m$ minimizing $\|\boldsymbol{r}_m\|_2$ with actualized m

$\boldsymbol{r}_0 = \boldsymbol{b} - \mathcal{A}\boldsymbol{w}_0$
$\beta = \|\boldsymbol{r}_0\|_2$
$\boldsymbol{v}_1 = \boldsymbol{r}_0/\beta$
$\boldsymbol{u}_1 = \mathcal{M}^{-1}\boldsymbol{v}_1$
for $k \in [\![1, m]\!]$ **do**
  $\boldsymbol{u}_{k+1} \leftarrow \mathcal{M}_j^{-1}\boldsymbol{v}_k$
  $\boldsymbol{v}_{k+1} \leftarrow \mathcal{A}\boldsymbol{u}_{k+1}$
  for $l \in [\![1, k]\!]$ **do**
    $h_{lk} \leftarrow \langle \mathcal{A}\boldsymbol{v}_k, \boldsymbol{v}_l \rangle$
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{w}_k - \boldsymbol{v}_l h_{lk}$
  end for
  $h_{k+1,k} \leftarrow \|\boldsymbol{v}_{k+1}\|_2$
  compute the relative residual $\epsilon_{\text{rel}}^{\text{GMRES}}$ with Givens rotations
  if $h_{k+1,k} = 0$ or $\epsilon_{\text{rel}}^{\text{GMRES}} <$ `tol` **then**
    $m \leftarrow k$
    break main loop
  else
    $\boldsymbol{v}_{k+1} \leftarrow \boldsymbol{v}_{k+1}/h_{k+1,k}$
  end if
end for
$\boldsymbol{y}_m = \text{argmin}_{\boldsymbol{y}} \|\beta \boldsymbol{e}_1 - \widetilde{\mathcal{H}}_m \boldsymbol{y}\|_2$
$\boldsymbol{x}_m = \boldsymbol{x}_0 + \mathcal{U}_m \boldsymbol{y}_m$

external Krylov bases must thus be carefully chosen to be compliant with memory limits. In each case, the preconditioning is initially built upon the Jacobian matrix. In the next paragraph, the procedure employed in elsA to compute $\mathcal{J}$ is described.

Formation and storage of $\mathcal{J}$. A review of existing preconditioners is exposed in [52]. Simple ones arise from classical iterative techniques like Jacobi or Gauss-Seidel. However, they are known to show a poor convergence rate when used in the context of compressible CFD. The so-called matrix-free approaches are really cheap in terms of both memory and CPU usage, but the most efficient among them requires *a minima* the formation of a good approximation of $\mathcal{A}$. When addressing compressible CFD problems, Guilbert [51] showed that the most convenient way is to precondition from the exact matrix. We recall that the matrices involved in CFD problems, like $\mathcal{J}$, usually are sparse matrices, i.e. matrices in which most of the elements are zero, as shown in fig. III.4.

The first step of preconditioning is therefore to compute the exact operator $\mathcal{A}$, and in particular the Jacobian matrix $\mathcal{J}$. In chapter I, the columns of $\mathcal{J}$ are calculated by evaluating successive matrix-vector products with AD on each vector of the canonical base. However, when dealing with larger 2-D or 3-D configurations, meshes could reach millions of cells, and such a strategy becomes unaffordable. In elsA, this issue is addressed by using a multicoloring technique. Full details about this approach can be found in [51]. To summarize, the idea is first to identify structurally orthogonal columns, i.e. columns that have no common element in the same line, and then to

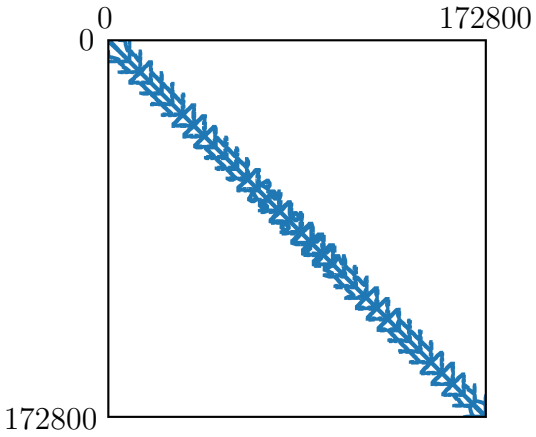


Figure III.4: Jacobian pattern of the laminar cylinder (0.9% of nonzero elements).

produce a probing vector through which a Jacobian-vector product returns the identified columns. Multicoloring can be illustrated with the following schematic sparse matrix whose structurally orthogonal columns has been colored the same way:

$$\mathcal{A} = \begin{pmatrix} \star & \star & & & \star & \\ & & \star & \star & \star & \\ \star & & & \star & & \star \\ \star & & & \star & \star & \\ & \star & \star & & & \star \end{pmatrix} \xrightarrow[\text{coloring}]{} \begin{pmatrix} \star & \star & & & \star & \\ & & \star & \star & \star & \\ \star & & & \star & & \star \\ \star & & & \star & \star & \\ & \star & \star & & & \star \end{pmatrix}. \tag{III.56}$$

Introducing the three following probing vectors:

$$\boldsymbol{t}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \boldsymbol{t}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{t}_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \tag{III.57}$$

the entire matrix can be recovered evaluating $\mathcal{A}\boldsymbol{t}_1$, $\mathcal{A}\boldsymbol{t}_2$ and $\mathcal{A}\boldsymbol{t}_3$. The matrix being built by blocks, the final number of matrix-vector products to compute is $N_{\text{color}} \times d$, which also depends on both the discretization order and the mesh topology.

Several storage format are available to store sparse matrices and most of them are presented in [52]. In elsA, $\mathcal{A}$ is stored in the Block compressed Sparse Row (BSR) format, that is the block variation of the Compressed Sparse Row (CSR) format. The CSR storage of a matrix $\mathcal{A}$ is composed of three tables $AA(\mathcal{A})$, $JA(\mathcal{A})$ and $IA(\mathcal{A})$. The first one contains every nonzero coefficient, the second their column index, and the third provides, for a given index p, the position in $AA(\mathcal{A})$ of the first nonzero coefficient of line p. An example of CSR storage for a 3-by-3 matrix is provided hereafter:

$$\mathcal{A} = \begin{pmatrix} a_{11} & 0 & a_{13} \\ 0 & a_{22} & 0 \\ 0 & a_{32} & a_{33} \end{pmatrix} \xrightarrow[\text{CSR}]{} \begin{aligned} AA(\mathcal{A}) &= \boxed{a_{11}\ a_{13}\ a_{22}\ a_{32}\ a_{33}} \\ JA(\mathcal{A}) &= \boxed{1\ 3\ 2\ 2\ 3} \\ IA(\mathcal{A}) &= \boxed{1\ 3\ 4} \end{aligned}. \tag{III.58}$$

The BSR format is identical except that the coefficients a_{qr} of $\mathcal{A}$ are d-by-d blocks instead of scalar quantities.

Preconditioners. This thesis builds upon the previous developments made by Guilbert in elsA, which give access to a large variety of preconditioning tools based on domain decomposition. Incomplete LU factorizations [52] like ILU(0) and ILU(k), where k is the level of fill-in, coupled with FGMRES, are sufficient for a large variety of CFD problems. The quality of those preconditioners may be enhanced by distributed computing extension *via* the Restricted Additive Schwarz (RAS) methods. When the problem is particularly stiff, like in RANS computations, more sophisticated preconditioning tools must be used. Guilbert [51] developed an hybrid method based on the mesh partitioning, that identifies the critical zones through the "LU-ILU distance" criterion, and decides whether ILU(0), ILU(k) or local LU factorization (see `https://www.pardiso-project.org/`) are performed. In the most critical zones, the preconditioner is a local LU factorization, and user-defined incomplete factorizations in the remaining zones. By allowing each domain to be preconditioned differently, this hybrid approach may yield important gain over a classical local LU decomposition. This type of preconditioning will be denoted HLU factorization in the following. Finding a good preconditioner for large sparse systems in CFD remains a challenging task and is still an important research topic at ONERA with for instance the work of Jang [102].

The different variations of preconditioning developed by Guilbert in elsA have been tested on laminar cylinder case. To test and compare the convergence of FGMRES with the various available preconditioners, the focus was made on the resolution of the system $\mathcal{J}\boldsymbol{z} = \boldsymbol{b}$, where $\mathcal{J}$ is the Jacobian matrix of a fixed point calculated at $\mathrm{Re} = 47$, and $\boldsymbol{b}$ is some random vector. This system is a particular case of equation (III.44) when $\mathrm{CFL} \to +\infty$, i.e. when the fixed point is reached and the time term vanishes. At the beginning of a fixed point computation, this term is large and increases the diagonal dominance of $\mathcal{A}$, which makes the system easier to solve. On the contrary, the closer the fixed point is, the stiffer the system becomes, resulting in a more challenging resolution. In the experiments, both internal and external Krylov basis were chosen to be composed of 60 vectors, and $\epsilon_{\mathrm{rel}}^{\mathrm{GMRES}}$ has been set to 10^{-14} while the inner GMRES tolerance is set to 10^{-3}. Fig. III.5 presents the convergence results using ILU(1), ILU(2), ILU(3), and HLU and table III.3 shows the required memory and elapsed time for each computation. HLU is the best in term of CPU time but also has the highest memory requirement. This is why in practice,

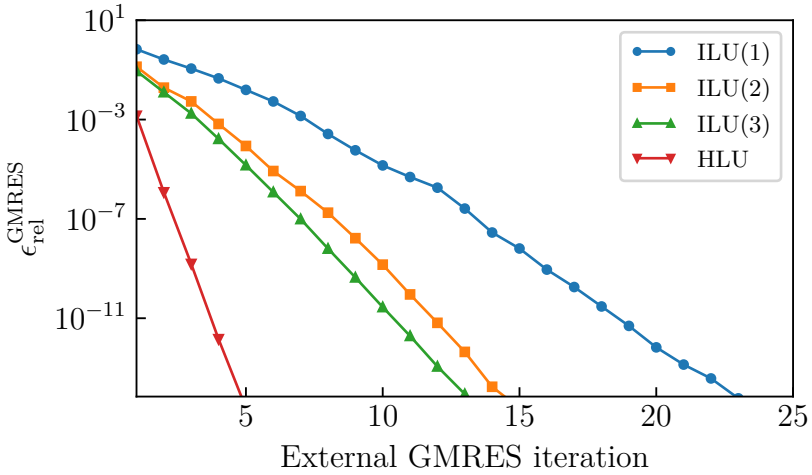


Figure III.5: Comparison of preconditioned GMRES

Preconditioner	ILU(1)	ILU(2)	ILU(3)	HLU
Allocated memory for preconditioning	0.88 GB	0.96 GB	1.04 GB	1.61 GB
Allocated memory for computation	1.14 GB	1.21 GB	1.29 GB	1.87 GB
CPU time	46.5 s	36.84 s	37 s	26.2 s

Table III.3: Memory consumption and computation duration for various preconditioners on the laminar cylinder case at Re = 47.

we prefer to use ILU(3) preconditioning which provides good performances in both CPU time and memory requirement. HLU will be used for highly stiff problems.

III.3.5 Fixed point calculation of the laminar cylinder flow

Fixed point computation on the laminar cylinder case have been performed by solving the implicit backward Euler scheme presented in II.3.2 with adaptative CFL [69] and using FGMRES-ILU(3) for inversions. The calculations were started with a uniform field. Considering Δl_k the characteristic length of a cell Ω_k, the local time-step in elsA is defined as follows:

$$\Delta t_k = \mathrm{CFL} \min\left(\frac{\Delta l_k}{\|\boldsymbol{U}_k\|_2 + a}, \frac{\rho \mathrm{Pr} \Delta l_k^2}{2\gamma\mu_k}\right) \tag{III.59}$$

Internal and external Krylov basis are both composed of 60 vectors. The internal GMRES stops when the relative error drops below 10^{-3} while the tolerance of the external GMRES is set at 10^{-4}. The nonlinear computation stops when the nonlinear residual on the density $\|\rho\|_2$ drops below 10^{-14}.

Fig. III.6a shows the convergence evolution over the computation at Re = 47. A satisfying level of convergence of 10^{-15} is obtained within 18 iterations. Fig. III.6b shows the streamlines of the corresponding base flow. A large recirculation bubble that extends over $x = 3.3$ composed

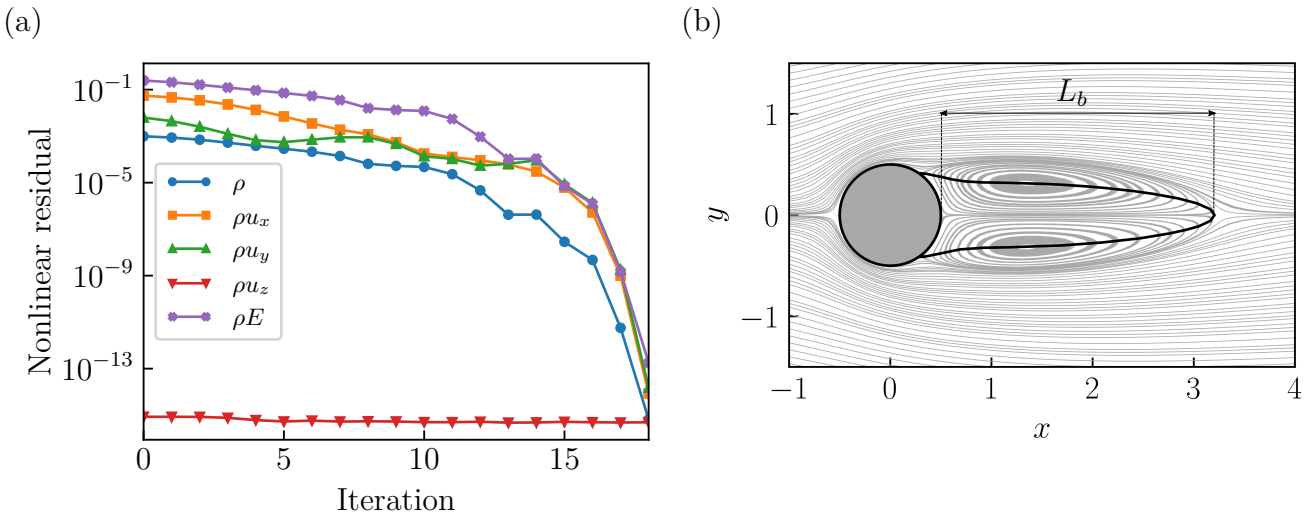


Figure III.6: Convergence curve (a) and streamlines (b) for the Re = 47 case. The solid black line indicates the separated region.

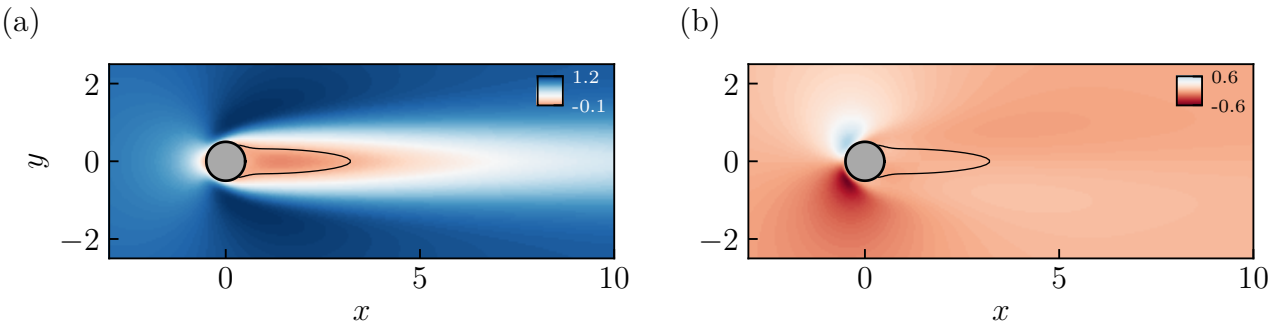


Figure III.7: Streamwise (a) and cross-stream (b) velocities of the baseflow computed at Re = 47. The solid line indicates the recirculation bubble.

of a pair of counter clockwise rotating vortices is observed immediately downstream the cylinder body, and the solid line highlights the iso-contour of zero streamwise velocity. The bubble length L_b is measured from the rear stagnation point to the rightmost point of this line. For the sake of exhaustivity, streamwise and cross-stream velocities are presented in fig. III.7.

The computations at various Reynolds number are validated by comparing the evolution of L_b when Re increases which, according to Gianetti & Luchini [30], should be linear. The bubble length evolution resulting from elsA computations is compared to the results of Gianetti & Lucchini in fig. III.8, where a good match is observed.

This section was dedicated to the presentation of the techniques selected in this thesis to accurately compute fixed points with elsA. The GMRES algorithm as well as the preconditioning techniques, as implemented to address inversions of the real-valued linear systems involved in fixed point computation, were detailed. The next natural step is to wonder how these tools could be adapted to solve the complex linear systems involved in global stability analyses. Next section presents the strategy developed in elsA to do so, and focus especially on the use of the K formulation [60].

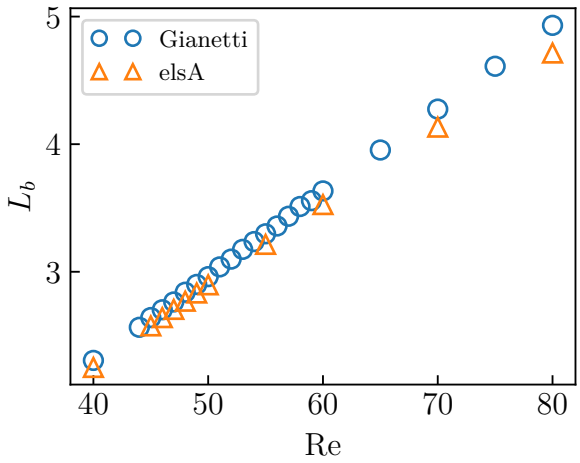


Figure III.8: Evolution of recirculation bubble length with Reynolds number.

III.4 K formulation and some of its properties

The tools of fixed point computation developed by Guilbert operate in real arithmetics, which differs from the global stability problems that are complex-valued. The code must therefore be thoroughly rewritten to be adapted to solve complex linear systems. Moreover, one of the main component of the HPC optimizations implemented for the fixed point computation is the vectorization of loop patterns. Under some conditions like the independence of each operation within a loop (see [103] for more details), this provides a significant speed-up during the computation [57]. For instance, if a given loop satisfies the vectorization conditions, and the optimization packs the indices five by five, this means that all subsequent operations are managed for the five indices at the same time. In practice, a speed-up factor of three is approximately observed when using vectorization. Another mandatory condition for the vectorization is the memory contiguity of the floats composing the input vector. However, in a complex vector, each component consist of a tuple of real and imaginary parts. From a memory point of view, the complex vector is stored in an alternance of real and imaginary components, and the contiguity is lost[9]. The use of complex arithmetics thus breaks the vectorization and consequently the HPC optimizations. To avoid such issue, we choose in this work to reformulate the complex systems involved in global stability in a real-equivalent version. Among all available formulations, the K formulation introduced by Day & Heroux [60] has been selected. The justification of this choice, as well as some of its mathematical properties are provided in this section.

III.4.1 Real equivalent formulation of a complex problem

For the sake of clarity, the focus is made on the general complex system under the following form:

$$(\delta\mathcal{B} + \gamma\mathcal{I}_N)\boldsymbol{z} = \boldsymbol{t}, \tag{III.60}$$

with $\boldsymbol{z}, \boldsymbol{t} \in \mathbb{C}^N$, $\gamma \in \mathbb{C}$, $\delta = \pm 1$, and where $\mathcal{B}$ stands either for $\mathcal{J}$ or $\mathcal{J}^T$. We introduce the sparse matrix $\mathcal{C} = (\delta\mathcal{B} + \gamma\mathcal{I}_N)$. By decomposing each vector and matrix in their real and imaginary part, and by respectively introducing $\mathcal{D}$ and $\mathcal{E}$ as the real and imaginary parts of $\mathcal{C}$, the system (III.60) becomes:

$$(\mathcal{D} + i\mathcal{E})(\boldsymbol{z}_\Re + i\boldsymbol{z}_\Im) = (\boldsymbol{t}_\Re + i\boldsymbol{t}_\Im),$$

which can be recast in the following form:

$$\underbrace{\begin{pmatrix} \mathcal{D} & -\mathcal{E} \\ \mathcal{E} & \mathcal{D} \end{pmatrix}}_{=\mathcal{C}^{\mathrm{K1}}} \begin{pmatrix} \boldsymbol{z}_\Re \\ \boldsymbol{z}_\Im \end{pmatrix} = \begin{pmatrix} \boldsymbol{t}_\Re \\ \boldsymbol{t}_\Im \end{pmatrix}. \tag{III.61}$$

This so-called K1 formulation provides an equivalent real alternative to the complex system where the operator $\mathcal{C} \in \mathbb{C}^N \times \mathbb{C}^N$ has been reformulated into operator $\mathcal{C}^{\mathrm{K1}} \in \mathbb{C}^{2N} \times \mathbb{C}^{2N}$. With such a modification, the vectorization is preserved however, the sparsity pattern of the original system is not. This is problematic since the preconditioners developed by Guilbert [51] are built from the sparsity pattern of the considered operator.

For this reason, we choose instead to focus on the equivalent K formulation [60], in which each coefficient $c_{qr} = d_{qr} + ie_{qr}$ is replaced by a 2-by-2 block entry c^{K}_{qr}:

$$c^{\mathrm{K}}_{qr} = \begin{pmatrix} d_{qr} & -e_{qr} \\ e_{qr} & d_{qr} \end{pmatrix}. \tag{III.62}$$

[9]The contiguity is achieved when for a given $\boldsymbol{v} \in \mathbb{C}^N$, each component of $\boldsymbol{v}_\Re$ and $\boldsymbol{v}_\Im$ are aligned in memory.

The K-formulated matrix $\mathcal{C}$ in the K formulation will be noted $\mathcal{C}^{\mathrm{K}}$. This mapping is an homomorphism [60], i.e. it satisfies the following condition:

$$\text{(i) } \mathcal{I}_N^{\mathrm{K}} = \mathcal{I}_{2N} \tag{III.63}$$

$$\text{(ii) } (\mathcal{X}\mathcal{Y})^{\mathrm{K}} = \mathcal{X}^{\mathrm{K}}\mathcal{Y}^{\mathrm{K}},\ \forall \mathcal{X}, \mathcal{Y} \in \mathbb{C}^N \times \mathbb{C}^N. \tag{III.64}$$

For example, the K formulation of the matrix

$$\mathcal{C} = \begin{pmatrix} c_{11} & 0 & c_{13} \\ 0 & c_{22} & 0 \\ 0 & c_{32} & c_{33} \end{pmatrix} \xrightarrow[\text{CSR}]{} \begin{aligned} AA(\mathcal{C}) &= \boxed{c_{11}\ c_{13}\ c_{22}\ c_{32}\ c_{33}} \\ JA(\mathcal{C}) &= \boxed{1\ 3\ 2\ 2\ 3} \\ IA(\mathcal{C}) &= \boxed{1\ 3\ 4} \end{aligned} \tag{III.65}$$

is the matrix

$$\mathcal{C}^{\mathrm{K}} = \left(\begin{array}{cc|cc|cc} d_{11} & -e_{11} & 0 & 0 & d_{13} & -e_{13} \\ e_{11} & d_{11} & 0 & 0 & e_{13} & d_{13} \\ \hline 0 & 0 & d_{22} & -e_{22} & 0 & 0 \\ 0 & 0 & e_{22} & d_{22} & 0 & 0 \\ \hline 0 & 0 & d_{32} & -e_{32} & d_{33} & -e_{33} \\ 0 & 0 & e_{32} & d_{32} & e_{33} & d_{33} \end{array}\right) \xrightarrow[\text{BSR}]{} \begin{aligned} AA(\mathcal{C}^{\mathrm{K}}) &= \boxed{c_{11}^{\mathrm{K}}\ c_{13}^{\mathrm{K}}\ c_{22}^{\mathrm{K}}\ c_{32}^{\mathrm{K}}\ c_{33}^{\mathrm{K}}} \\ JA(\mathcal{C}^{\mathrm{K}}) &= \boxed{1\ 3\ 2\ 2\ 3} \\ IA(\mathcal{C}^{\mathrm{K}}) &= \boxed{1\ 3\ 4} \end{aligned} \tag{III.66}$$

The block sparsity pattern of $\mathcal{C}^{\mathrm{K}}$ exactly matches the point sparsity pattern of $\mathcal{C}$, and the CSR storage of $\mathcal{C}$ has the exact same structure as the BSR storage of $\mathcal{C}^{\mathrm{K}}$. Consequently, applying any pattern-based block preconditioning technique like those implemented in the elsA software (ILU(k), local LU, etc.) to $\mathcal{C}^{\mathrm{K}}$ corresponds to applying the equivalent point preconditioning to $\mathcal{C}$. This statement stays true if c_{qr} are d-by-d block entries, as in elsA, and the block size of the K-formulated operator becomes $2d$-by-$2d$. This is shown in the case of the cylinder in fig. III.9, on which we note that the patterns of $\mathcal{J}$ and $\mathcal{J}^{\mathrm{K}}$ are identical.

As detailed in II.6, the KS algorithm operates in complex arithmetics. To benefit from the vectorization during the whole eigenproblem resolution, the real-equivalent formulation needs to be conserved within each step of the KS algorithm, and not only for the inversions of linear systems. To this end, the original eigenproblems are reformulated and the new problematic is to be able to link the eigenpairs of the K-formulated operators to those of the original ones.

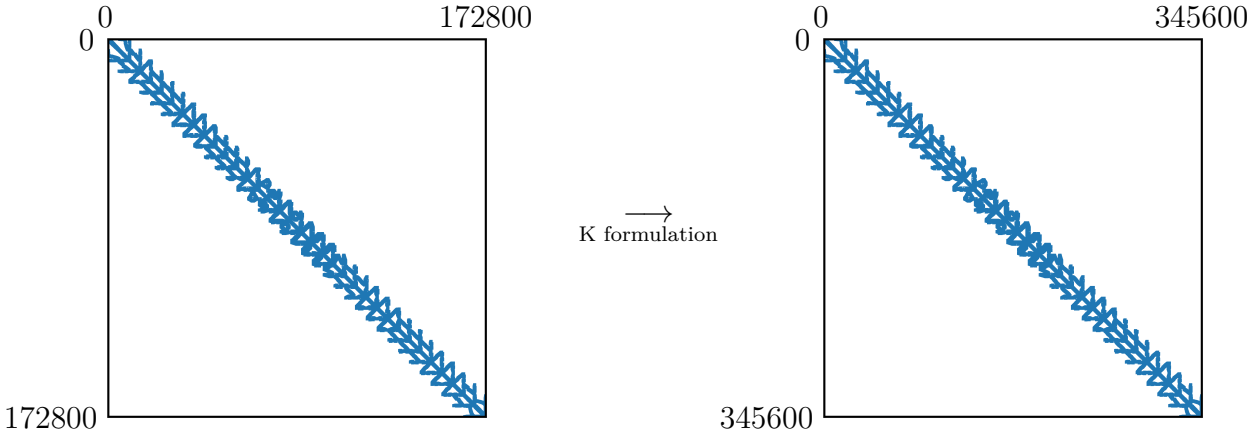


Figure III.9: Patterns of $\mathcal{J}$ (left) and $\mathcal{J}^{\mathrm{K}}$ (right) for the laminar cylinder.

III.4.2 Eigenvalues of the K-formulated operator

Considering the eigenvalues $\lambda_1, \ldots, \lambda_N$ of $\mathcal{C}$ in any prescribed order, the goal is to explicit the spectrum $\mathrm{Sp}(\mathcal{C}^{\mathrm{K}})$. Provided that $\mathcal{C}$ is a complex squared matrix, there exists a so-called complex Schur decomposition verifying:

$$\mathcal{C} = \mathcal{X}\mathcal{S}\mathcal{X}^H, \tag{III.67}$$

where $\mathcal{X} \in \mathbb{C}^N \times \mathbb{C}^N$ is a unitary matrix[10], and $\mathcal{S} \in \mathbb{C}^N \times \mathbb{C}^N$ is an upper triangular matrix whose diagonal entries are the eigenvalues of $\mathcal{C}$:

$$\mathcal{S} = \begin{pmatrix} \lambda_1 & & \star \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix}. \tag{III.68}$$

From property (III.64), the following relation holds:

$$\mathcal{C}^{\mathrm{K}} = \mathcal{X}^{\mathrm{K}}\mathcal{S}^{\mathrm{K}}\mathcal{X}^{H^{\mathrm{K}}}, \tag{III.69}$$

where $\mathcal{S}^{\mathrm{K}}$ reads:

$$\mathcal{S}^{\mathrm{K}} = \begin{pmatrix} \Lambda_1 & & \star \\ & \ddots & \\ 0 & & \Lambda_N \end{pmatrix}, \text{ with } \Lambda_q = \begin{pmatrix} \Re(\lambda_q) & -\Im(\lambda_q) \\ \Im(\lambda_q) & \Re(\lambda_q) \end{pmatrix}, \text{ for all } q \in [\![1, N]\!]. \tag{III.70}$$

Proposition 1 *$Sp(\mathcal{C}^K) = Sp(\mathcal{C}) \cup Sp(\overline{\mathcal{C}})$.*

Proof Let us first focus on the attributes of $\mathcal{X}^{\mathrm{K}}$. From properties (III.63) and (III.64), we have:

$$\mathcal{X}^{\mathrm{K}}\mathcal{X}^{H^{\mathrm{K}}} = (\mathcal{X}\mathcal{X}^H)^{\mathrm{K}} = \mathcal{I}_N^{\mathrm{K}} = \mathcal{I}_{2N}. \tag{III.71}$$

Moreover, if the general term of $\mathcal{X}$ is $(x_{qr})_{1\leq q,r\leq N}$, then that of $\mathcal{X}^H$ writes $(\overline{x}_{rq})_{1\leq r,q\leq N}$ and the general terms of $\mathcal{X}^{\mathrm{K}}$ and $\mathcal{X}^{H^{\mathrm{K}}}$ are, for a given $(q, r) \in [\![1, N]\!] \times [\![1, N]\!]$:

$$x_{qr}^{\mathrm{K}} = \begin{pmatrix} \Re(x_{qr}) & -\Im(x_{qr}) \\ \Im(x_{qr}) & \Re(x_{qr}) \end{pmatrix} \quad \text{and} \quad \overline{x}_{rq}^{\mathrm{K}} = \begin{pmatrix} \Re(x_{qr}) & \Im(x_{qr}) \\ -\Im(x_{qr}) & \Re(x_{qr}) \end{pmatrix}. \tag{III.72}$$

Equation (III.72) highlights the relation $\mathcal{X}^{H^{\mathrm{K}}} = \mathcal{X}^{\mathrm{K}^T}$, which combined with (III.71), yields:

$$\mathcal{X}^{\mathrm{K}}\mathcal{X}^{\mathrm{K}^T} = \mathcal{I}_{2N}. \tag{III.73}$$

We can therefore state that the K formulation $\mathcal{X}^{\mathrm{K}}$ of the unitary matrix $\mathcal{X}$ is an orthogonal matrix, and the decomposition (III.69) can be rewritten under the following form:

$$\mathcal{C}^{\mathrm{K}} = \mathcal{X}^{\mathrm{K}}\mathcal{S}^{\mathrm{K}}\mathcal{X}^{\mathrm{K}^T}. \tag{III.74}$$

Considering the shape of $\mathcal{S}^{\mathrm{K}}$, we can stated, from [104], that the relation (III.74) is a real Schur decomposition. In this real form, the eigenvalues are contained in 2-by-2 diagonal blocks (or

[10] verifying $\mathcal{X}\mathcal{X}^H = \mathcal{I}_N$.

scalar for real eigenvalues) instead of complex scalars. The consequence of this result is that $\mathrm{Sp}(\mathcal{C}^{\mathrm{K}}) = \mathrm{Sp}(\mathcal{S}^{\mathrm{K}})$ and the eigenvalues of $\mathcal{S}^{\mathrm{K}}$ are those of all of its 2-by-2 diagonal blocks, i.e.:

$$\mathrm{Sp}(\mathcal{S}^{\mathrm{K}}) = \bigcup_{q=1}^{N} \mathrm{Sp}(\Lambda_q) = \bigcup_{q=1}^{N} \{\lambda_q, \overline{\lambda}_q\}, \tag{III.75}$$

which demonstrates proposition 1:

$$\boxed{\mathrm{Sp}(\mathcal{C}^{\mathrm{K}}) = \mathrm{Sp}(\mathcal{C}) \cup \mathrm{Sp}(\overline{\mathcal{C}})}. \tag{III.76}$$

The operator $\mathcal{C}^{\mathrm{K}}$ thus contains the spectral informations of both $\mathcal{C}$ and $\overline{\mathcal{C}}$. The next objective is to recover the eigenvectors of $\mathcal{C}$ or $\overline{\mathcal{C}}$ from those of $\mathcal{C}^{\mathrm{K}}$. This point is addressed in the following.

III.4.3 Eigenvectors of the K-formulated operator

As stated in [60], $\mathcal{C}^{\mathrm{K}}$ and $\mathcal{C}^{\mathrm{K1}}$ are orthogonally similar matrices linked through the relation:

$$\mathcal{C}^{\mathrm{K}} = \mathcal{P}\mathcal{C}^{\mathrm{K1}}\mathcal{P}^{T}, \tag{III.77}$$

where $\mathcal{P}$ is a permutation matrix that, for a given $(q, r) \in [\![1, 2N]\!] \times [\![1, 2N]\!]$, is of general term:

$$p_{qr} = \begin{cases} 1 \text{ if } q \text{ is odd and } r = (q+1)/2, \\ 1 \text{ if } q \text{ is even and } r = q/2 + N, \\ 0 \text{ otherwise.} \end{cases} \tag{III.78}$$

The main consequence of relation (III.77) is that $\mathrm{Sp}(\mathcal{C}^{\mathrm{K}}) = \mathrm{Sp}(\mathcal{C}^{\mathrm{K1}})$. Considering a complex eigenvector $\boldsymbol{v} = (v)_{1 \leq q \leq 2N} \in \mathbb{C}^{2N}$ of $\mathcal{C}^{\mathrm{K}}$ associated with the eigenvalue $\lambda = \alpha + i\beta$, we hence know that $\mathcal{P}\boldsymbol{v}$ is an eigenvector of $\mathcal{C}^{\mathrm{K1}}$ verifying:

$$\mathcal{C}^{\mathrm{K1}}\mathcal{P}\boldsymbol{v} = \lambda\mathcal{P}\boldsymbol{v}, \tag{III.79}$$

and that λ is also an eigenvalue of $\mathcal{C}^{\mathrm{K1}}$. Introducing the two vectors $\boldsymbol{v}^o = (v_{2q})_{1 \leq q < N} \in \mathbb{C}^N$ and $\boldsymbol{v}^e = (v_{2q+1})_{1 \leq q < N} \in \mathbb{C}^N$ (respectively composed of the odd and even components of $\boldsymbol{v}$), and based on the definition (III.78) of $\mathcal{P}$:

$$\mathcal{P}\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}^o \\ \boldsymbol{v}^e \end{pmatrix}. \tag{III.80}$$

In the case $N = 3$, this leads to:

$$\mathcal{P}\boldsymbol{v} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_3 \\ v_5 \\ v_2 \\ v_4 \\ v_6 \end{pmatrix}. \tag{III.81}$$

These relations are introduced in order to simplify the next demonstration. The following demonstrated propositions clarify the shape of $\boldsymbol{v}$ depending on whether λ belongs to $\mathrm{Sp}(\mathcal{C})$, to $\mathrm{Sp}(\overline{\mathcal{C}})$, or to $\mathrm{Sp}(\mathcal{C}) \cap \mathrm{Sp}(\overline{\mathcal{C}})$ (which is included in $\mathbb{R}$), and show how an eigenvector of $\mathcal{C}$ can be recovered from $\boldsymbol{v}$.

Proposition 2 *$\boldsymbol{v}^o$ and $\boldsymbol{v}^e$ are eigenvectors of $\mathcal{C}$ associated with $\lambda \in \mathbb{C} \Leftrightarrow \boldsymbol{v}^o = -i\boldsymbol{v}^e$.*

Proof This proposition is an equivalence that is demonstrated showing both implications. We first focus on the left-to-right implication. Considering that $\boldsymbol{v}^o$ and $\boldsymbol{v}^e$ are eigenvectors of $\mathcal{C}$ associated with the eigenvalue λ, we have $\mathcal{C}\boldsymbol{v}^o = \lambda\boldsymbol{v}^o$ and $\mathcal{C}\boldsymbol{v}^e = \lambda\boldsymbol{v}^e$. Expanding these two relations leads to the following set of equations:

$$\begin{cases} \mathcal{D}\boldsymbol{v}^o_{\Re} - \mathcal{E}\boldsymbol{v}^o_{\Im} = \alpha\boldsymbol{v}^o_{\Re} - \beta\boldsymbol{v}^o_{\Im} & \text{(III.82)} \\ \mathcal{E}\boldsymbol{v}^o_{\Re} + \mathcal{D}\boldsymbol{v}^o_{\Im} = \beta\boldsymbol{v}^o_{\Re} + \alpha\boldsymbol{v}^o_{\Im} & \text{(III.83)} \\ \mathcal{D}\boldsymbol{v}^e_{\Re} - \mathcal{E}\boldsymbol{v}^e_{\Im} = \alpha\boldsymbol{v}^e_{\Re} - \beta\boldsymbol{v}^e_{\Im} & \text{(III.84)} \\ \mathcal{E}\boldsymbol{v}^e_{\Re} + \mathcal{D}\boldsymbol{v}^e_{\Im} = \beta\boldsymbol{v}^e_{\Re} + \alpha\boldsymbol{v}^e_{\Im}. & \text{(III.85)} \end{cases}$$

$\mathcal{P}\boldsymbol{v}$ being an eigenvector of $\mathcal{C}^{\mathrm{K1}}$, equation (III.79) can be recast as:

$$\begin{pmatrix} \mathcal{D} & -\mathcal{E} \\ -\mathcal{E} & \mathcal{D} \end{pmatrix} \begin{pmatrix} \boldsymbol{v}^o_{\Re} + i\boldsymbol{v}^o_{\Im} \\ \boldsymbol{v}^e_{\Re} + i\boldsymbol{v}^e_{\Im} \end{pmatrix} = (\alpha + i\beta) \begin{pmatrix} \boldsymbol{v}^o_{\Re} + i\boldsymbol{v}^o_{\Im} \\ \boldsymbol{v}^e_{\Re} + i\boldsymbol{v}^e_{\Im} \end{pmatrix}. \tag{III.86}$$

This system is equivalent to the following new set of equations:

$$\begin{cases} \mathcal{D}\boldsymbol{v}^o_{\Re} - \mathcal{E}\boldsymbol{v}^e_{\Re} = \alpha\boldsymbol{v}^o_{\Re} - \beta\boldsymbol{v}^o_{\Im} & \text{(III.87)} \\ \mathcal{E}\boldsymbol{v}^o_{\Re} + \mathcal{D}\boldsymbol{v}^e_{\Re} = \alpha\boldsymbol{v}^e_{\Re} - \beta\boldsymbol{v}^e_{\Im} & \text{(III.88)} \\ \mathcal{D}\boldsymbol{v}^o_{\Im} - \mathcal{E}\boldsymbol{v}^e_{\Im} = \beta\boldsymbol{v}^o_{\Re} + \alpha\boldsymbol{v}^o_{\Im} & \text{(III.89)} \\ \mathcal{E}\boldsymbol{v}^o_{\Im} + \mathcal{D}\boldsymbol{v}^e_{\Im} = \beta\boldsymbol{v}^e_{\Re} + \alpha\boldsymbol{v}^e_{\Im}. & \text{(III.90)} \end{cases}$$

After summing equation (III.82) and (III.87), and rearranging, a new relation is obtained:

$$\mathcal{D}\boldsymbol{v}^o_{\Re} - \frac{1}{2}\mathcal{E}(\boldsymbol{v}^o_{\Im} + \boldsymbol{v}^o_{\Re}) = \alpha\boldsymbol{v}^o_{\Re} - \beta\boldsymbol{v}^o_{\Im}, \tag{III.91}$$

which, referring to equation (III.82), is accurate only if:

$$\frac{1}{2}(\boldsymbol{v}^o_{\Im} + \boldsymbol{v}^e_{\Re}) = \boldsymbol{v}^e_{\Re}, \tag{III.92}$$

i.e. $\boldsymbol{v}^o_{\Im} = \boldsymbol{v}^e_{\Re}$. Such a result allows the replacement of $\boldsymbol{v}^e_{\Re}$ by $\boldsymbol{v}^o_{\Im}$ in equation (III.88), making the left-hand side equal to that of equation (III.83). Equaling both right-hand sides provides:

$$\beta\boldsymbol{v}^o_{\Re} + \alpha\boldsymbol{v}^o_{\Im} = \alpha\boldsymbol{v}^o_{\Im} - \beta\boldsymbol{v}^e_{\Im}. \tag{III.93}$$

Assuming $\lambda \notin \mathrm{Sp}(\mathcal{C}) \cap \mathrm{Sp}(\overline{\mathcal{C}})$, and thus $\lambda \notin \mathbb{R}$, then $\beta \neq 0$, and finally $\boldsymbol{v}^o_{\Re} = -\boldsymbol{v}^e_{\Im}$, which completes the demonstration of $\boxed{\boldsymbol{v}^o = -i\boldsymbol{v}^e}$.

We now focus on the right-to-left implication by considering that $\boldsymbol{v}^o = -i\boldsymbol{v}^e$, i.e. $\boldsymbol{v}^o_{\Re} = -i\boldsymbol{v}^e_{\Im}$ and $\boldsymbol{v}^o_{\Im} = \boldsymbol{v}^e_{\Re}$. $\mathcal{P}\boldsymbol{v}$ is an eigenvector of $\mathcal{C}^{\mathrm{K1}}$ so the set of equations (III.87) to (III.90) holds. Replacing $\boldsymbol{v}^e_{\Re}$ by $\boldsymbol{v}^o_{\Im}$ and $\boldsymbol{v}^e_{\Im}$ by $-\boldsymbol{v}^o_{\Re}$ in equation (III.87) and (III.88) leads to equation (III.82) and (III.83). And replacing $\boldsymbol{v}^o_{\Re}$ by $-\boldsymbol{v}^e_{\Im}$ and $\boldsymbol{v}^o_{\Im}$ by $\boldsymbol{v}^e_{\Re}$ in equation (III.89) and (III.90) leads to equation (III.84) and (III.85). Finally, the whole system (III.82) to (III.85) is satisfied, which means $\boxed{\boldsymbol{v}^o \text{ and } \boldsymbol{v}^e \text{ are eigenvectors of } \mathcal{C}}$.

Proposition 3 *$\boldsymbol{v}^o$ and $\boldsymbol{v}^e$ are eigenvectors of $\overline{C}$ associated with $\lambda \in \mathbb{C} \Leftrightarrow \boldsymbol{v}^o = i\boldsymbol{v}^e$.*

PROOF The proof is similar to that of proposition 2, except that the system (III.82) to (III.85) is replaced by:

$$\begin{cases} \mathcal{D}\boldsymbol{v}^o_{\Re} + \mathcal{E}\boldsymbol{v}^o_{\Im} = \alpha\boldsymbol{v}^o_{\Re} - \beta\boldsymbol{v}^o_{\Im} & \text{(III.94)} \\ -\mathcal{E}\boldsymbol{v}^o_{\Re} + \mathcal{D}\boldsymbol{v}^o_{\Im} = \beta\boldsymbol{v}^o_{\Re} + \alpha\boldsymbol{v}^o_{\Im} & \text{(III.95)} \\ \mathcal{D}\boldsymbol{v}^e_{\Re} + \mathcal{E}\boldsymbol{v}^e_{\Im} = \alpha\boldsymbol{v}^e_{\Re} - \beta\boldsymbol{v}^e_{\Im} & \text{(III.96)} \\ -\mathcal{E}\boldsymbol{v}^e_{\Re} + \mathcal{D}\boldsymbol{v}^e_{\Im} = \beta\boldsymbol{v}^e_{\Re} + \alpha\boldsymbol{v}^e_{\Im}. & \text{(III.97)} \end{cases}$$

Proposition 4 *If $\lambda \in Sp(\mathcal{C}) \cap Sp(\overline{\mathcal{C}})$, i.e. $\lambda \in \mathbb{R}$, then $\boldsymbol{v}^o_{\Re} + i\boldsymbol{v}^e_{\Re}$ and $\boldsymbol{v}^o_{\Im} + i\boldsymbol{v}^e_{\Im}$ are eigenvectors of $\mathcal{C}$; and $\boldsymbol{v}^o_{\Re} - i\boldsymbol{v}^e_{\Re}$ and $\boldsymbol{v}^o_{\Im} + i\boldsymbol{v}^e_{\Im}$ are eigenvectors of $\overline{C}$.*

PROOF λ being real, the system (III.87) to (III.90) simplifies as:

$$\begin{cases} \mathcal{D}\boldsymbol{v}^o_{\Re} - \mathcal{E}\boldsymbol{v}^e_{\Re} = \alpha\boldsymbol{v}^o_{\Re} & \text{(III.98)} \\ \mathcal{E}\boldsymbol{v}^o_{\Re} + \mathcal{D}\boldsymbol{v}^e_{\Re} = \alpha\boldsymbol{v}^e_{\Re} & \text{(III.99)} \\ \mathcal{D}\boldsymbol{v}^o_{\Im} - \mathcal{E}\boldsymbol{v}^e_{\Im} = \alpha\boldsymbol{v}^o_{\Im} & \text{(III.100)} \\ \mathcal{E}\boldsymbol{v}^o_{\Im} + \mathcal{D}\boldsymbol{v}^e_{\Im} = \alpha\boldsymbol{v}^e_{\Im}, & \text{(III.101)} \end{cases}$$

which shows that $\boldsymbol{v}^o_{\Re} + i\boldsymbol{v}^e_{\Re}$ and $\boldsymbol{v}^o_{\Im} + i\boldsymbol{v}^e_{\Im}$ are eigenvectors of $\mathcal{C}$, and $\boldsymbol{v}^o_{\Re} - i\boldsymbol{v}^e_{\Re}$ and $\boldsymbol{v}^o_{\Im} - i\boldsymbol{v}^e_{\Im}$ are eigenvectors of $\overline{C}$.

For a given eigenpair of $\mathcal{C}^{\mathrm{K}}$, whether the eigenvalue is complex or real, we are hence able to recover an eigenpair of $\mathcal{C}$. We now have all the required informations to adapt the KS algorithm to K formulation. The next section provides the implementation details of the K formulation applied to both GMRES and KS algorithms.

III.5 K-formulated global stability analysis method

To get a complete working implementation of global stability analysis in K formulation, each component of the eigenproblem resolution must be adapted. This section first presents the adaptation of the Jacobian-vector products of Tapenade, which is one of the main elementary component of the GMRES. The preconditioning tools are also adapted and some convergence results of the GMRES in K formulation are provided for the laminar cylinder case.

III.5.1 K-formulated matrix-vector product

Producing differentiated code with Tapenade in K formulation is not only doubling the size of each input vectors. In fact, as mentioned earlier, the contiguity of the real and imaginary components of the input K-formulated vector must be conserved. To this end, the input vector must be split and Jacobian-vector products must be managed on the real and imaginary parts separately. The description of the process is described below.

For the sake of illustration and without loss of generalities, we consider here that $\mathcal{B} = \mathcal{J}$ and

$\delta = 1$. The product between $\mathcal{C} = \mathcal{J} + s\mathcal{I}_N$ and a given complex vector $\boldsymbol{v} \in \mathbb{C}^N$ writes:

$$\begin{aligned}\mathcal{C}\boldsymbol{v} &= \left[\mathcal{J} + (s_\Re + i s_\Im)\,\mathcal{I}_N\right](\boldsymbol{v}_\Re + i\boldsymbol{v}_\Im) \\ &= \mathcal{J}\boldsymbol{v}_\Re + s_\Re\boldsymbol{v}_\Re - s_\Im\boldsymbol{v}_\Im + i\left(\mathcal{J}\boldsymbol{v}_\Im + s_\Re\boldsymbol{v}_\Im + s_\Im\boldsymbol{v}_\Re\right) \\ &= \underbrace{\begin{pmatrix} (\mathcal{J}\boldsymbol{v}_\Re)_1 + s_\Re v_{\Re_1} - s_\Im v_{\Im_1} \\ \vdots \\ (\mathcal{J}\boldsymbol{v}_\Re)_N + s_\Re v_{\Re_N} - s_\Im v_{\Im_N} \end{pmatrix}}_{=\Re(\mathcal{C}\boldsymbol{v})} + i \underbrace{\begin{pmatrix} (\mathcal{J}\boldsymbol{v}_\Im)_1 + s_\Re v_{\Im_1} - s_\Im v_{\Re_1} \\ \vdots \\ (\mathcal{J}\boldsymbol{v}_\Im)_N + s_\Re v_{\Im_N} - s_\Im v_{\Re_N} \end{pmatrix}}_{=\Im(\mathcal{C}\boldsymbol{v})}.\end{aligned} \tag{III.102}$$

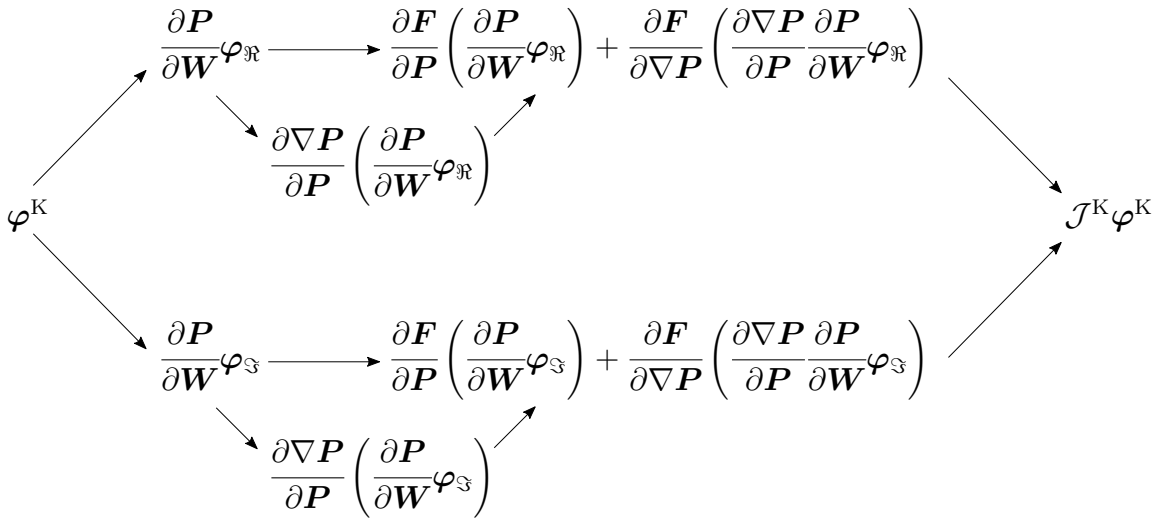
The general term of the K formulation $\mathcal{C}^{\mathrm{K}} = (c^{\mathrm{K}}_{qr})_{1\leq q,r\leq N}$ of $\mathcal{C}$ is expressed as follows:

$$c^{\mathrm{K}}_{qr} = \begin{cases} \begin{pmatrix} \mathcal{J}_{qr} + s_\Re & -s_\Im \\ s_\Im & \mathcal{J}_{qr} + s_\Re \end{pmatrix} & \text{if } q = r, \\ \begin{pmatrix} \mathcal{J}_{qr} & 0 \\ 0 & \mathcal{J}_{qr} \end{pmatrix} & \text{otherwise.} \end{cases} \tag{III.103}$$

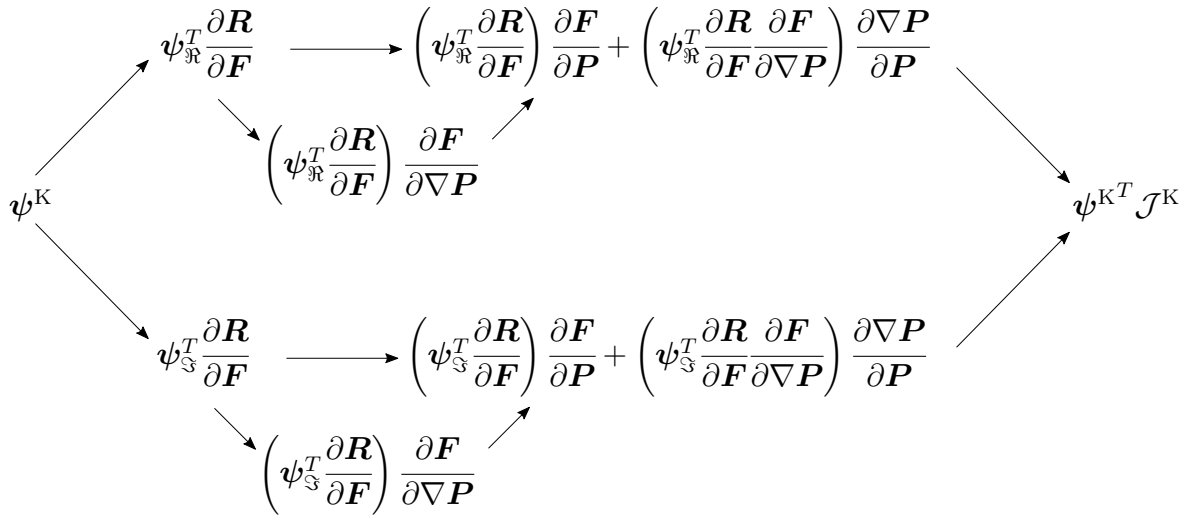
Now, reformulating $\boldsymbol{v}$ in $\boldsymbol{v}^{\mathrm{K}} = \begin{pmatrix} v_{\Re_1} & v_{\Im_1} & \cdots & \cdots & v_{\Re_N} & v_{\Im_N} \end{pmatrix}^T \in \mathbb{R}^{2N}$, the K formulation of $\mathcal{C}\boldsymbol{v}$ reads:

$$\mathcal{C}^{\mathrm{K}}\boldsymbol{v}^{\mathrm{K}} = \begin{pmatrix} (\mathcal{J}\boldsymbol{v}_\Re)_1 + s_\Re v_{\Re_1} - s_\Im v_{\Im_1} \\ (\mathcal{J}\boldsymbol{v}_\Im)_1 + s_\Re v_{\Im_1} - s_\Im v_{\Re_1} \\ \vdots \\ \vdots \\ (\mathcal{J}\boldsymbol{v}_\Re)_N + s_\Re v_{\Re_N} - s_\Im v_{\Im_N} \\ (\mathcal{J}\boldsymbol{v}_\Im)_N + s_\Re v_{\Im_N} - s_\Im v_{\Re_N} \end{pmatrix}. \tag{III.104}$$

To achieve the computation of $\mathcal{C}^{\mathrm{K}}\boldsymbol{v}^{\mathrm{K}}$, the vector $\boldsymbol{v}^{\mathrm{K}}$ has first to be split into $\boldsymbol{v}_\Re$ and $\boldsymbol{v}_\Im$. Given the Jacobian-vector products provided by Tapenade (see II.4.1), the two products $\mathcal{J}\boldsymbol{v}_\Re$ and $\mathcal{J}\boldsymbol{v}_\Im$ must then be performed. They are finally assembled in a $2N$-dimensional vector with the remaining terms of $s\boldsymbol{v}$. Using the notation of II.4.2, the graph providing the K-formulated right matrix-vector product, i.e. the mapping $\boldsymbol{\varphi}^{\mathrm{K}} \in \mathbb{R}^{2N} \mapsto \mathcal{J}^{\mathrm{K}}\boldsymbol{\varphi}^{\mathrm{K}} \in \mathbb{R}^{2N}$, reads:



while that of the left matrix-vector product, i.e. the mapping $\boldsymbol{\psi}^{\mathrm{K}} \in \mathbb{R}^{2N} \mapsto \mathcal{J}^{T^{\mathrm{K}}}\boldsymbol{\psi}^{\mathrm{K}} \in \mathbb{R}^{2N}$, reads:



A possible implementation consists in extracting and storing the real and imaginary parts from the input K-formulated vector, successively computing the two Jacobian-vector products, and recombining them into a new K-formulated vector. This strategy requires three copies for each matrix-vector product in K formulation, resulting in a large number of additional copies when thousands of matrix-vector products are needed as in global stability computation. Alternatively, by inlining the product in K formulation, the real and imaginary Jacobian-vector products might be computed simultaneously on a $2N$-dimensional vector conveniently arranged. This strategy avoids the three copies and improves the vectorization and cache-blocking optimization. It has been the object of a collaboration with Hascoët and resulted in a new feature of Tapenade in which the operations on the real and imaginary parts are done directly on the $2N$-dimensional vector. Examples of K-formulated differentiated codes are provided in appendix B.

III.5.2 GMRES algorithm in K formulation

By doubling the size of the Krylov vectors and using the matrix-vector product in K formulation, the whole GMRES implementation described in III.3.4 can be used on the K-formulated problem $\mathcal{C}^{\mathrm{K}}\boldsymbol{z}^{\mathrm{K}} = \boldsymbol{t}^{\mathrm{K}}$. Likewise, the preconditioning routines already implemented in elsA can be reused on the K-formulated operators, doubling the block size, as shown in III.4.1. The jacobian matrix is formed at the beginning of the computation, and the K-formulated operator is obtained by transforming $AA(\mathcal{J})$ to $AA(\mathcal{J}^{\mathrm{K}})$.

The resulting implementation of the GMRES in K formulation has been tested on the K-formulated linear systems associated to the laminar cylinder case. As in III.3.4, both external and internal Krylov basis are composed of 60 vectors, and the solver is the FGMRES-ILU(3). Results are presented in fig. III.10, and table III.4 shows the allocated memory and time of each computation. We first observe that the resolution converges faster with the K-formulated adjoint operator than for the direct one. This agrees with the results of Timme [55] who also noticed such behaviour. On the other hand, when $s = 0$, convergence is more difficult than when $s \neq 0$. This is explained by the reinforcement of the diagonal dominance (see III.3.4) with a nonzero shift. Additionally, when the shift is really close to an eigenvalue, the operator is almost singular resulting in a large condition number and a more difficult inversion. A balance is thus to be found between setting a shift with a large real part, which makes the inversions easy and fast but requires much more eigensolver iterations to converge toward the desired eigenvalue; and setting a shift

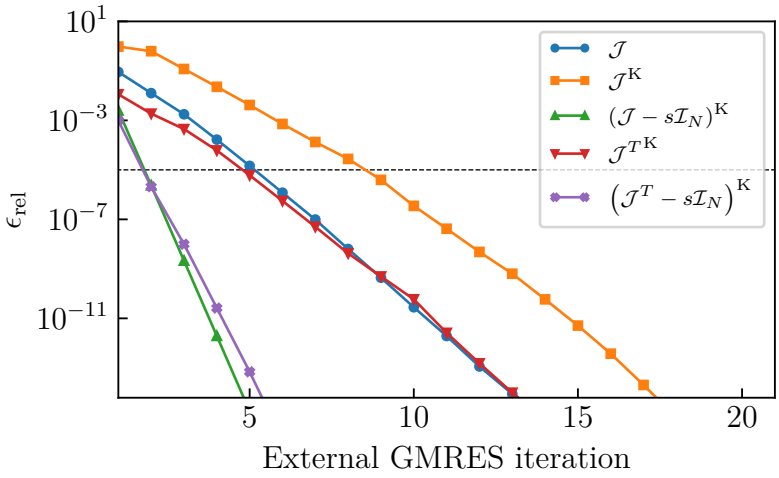


Figure III.10: GMRES convergence curves for various operators on the laminar cylinder case at $\mathrm{Re} = 47$. The dashed line is at 10^{-5} which is the selected tolerance for global stability analysis about cylinder flow. the shift is $s = 1 + 0.9i$.

Operator	$\mathcal{J}$	$\mathcal{J}^{\mathrm{K}}$	$(\mathcal{J} - s\mathcal{I}_N)^{\mathrm{K}}$	$\mathcal{J}^{T\mathrm{K}}$	$\left(\mathcal{J}^T - s\mathcal{I}_N\right)^{\mathrm{K}}$
Allocated memory for preconditioning	1.04 GB	1.61 GB	1.72 GB	1.71 GB	1.84 GB
Allocated memory for computation	1.29 GB	3.01 GB	3.03 GB	3.06 GB	3.05 GB
CPU time	37 s	86.2 s	34 s	65.7 s	34.2 s

Table III.4: Memory consumption and computation duration for various operators on the laminar cylinder case at $\mathrm{Re} = 47$.

very close to the selected eigenvalue, which makes the eigensolver faster but the systems stiffer ant the inversions slower.

III.5.3 Adaptating the Krylov-Schur algorithm to the K formulation

The KS algorithm also have to be adapted to K formulation. It has been implemented in elsA in its real version, in which the complex Schur decomposition in equation (III.67) is replaced by a real one [104], where the eigenvalues are contained in 2-by-2 diagonal blocks (or scalar for real eigenvalues). The size of the Krylov vectors that compose the KS basis has been doubled to comply with the K formulation. For the shift-and-invert transformation, the convergence test has been modified to test the errors associated with eigenvectors of $\mathcal{J}$ instead of $\mathcal{J}^{\mathrm{K}}$. Some details of these implementations are provided below.

Modal analysis. Whether we focus on the direct of adjoint eigenproblem, the operator of interest has the form of a shift-and-invert operator, as shown in II.5. The KS algorithm builds a KS basis from K formulation of operator $\mathcal{C}^{-1} = (\mathcal{B} - s\mathcal{I}_N)^{-1}$. The homomorphism properties (III.63) and (III.64) of the K formulation provide:

$$\mathcal{C}^{-1^{\mathrm{K}}} = \mathcal{C}^{\mathrm{K}^{-1}}. \tag{III.105}$$

The KS basis is generated by successive evaluations of $\mathcal{C}^{\mathrm{K}^{-1}}$ on the Krylov vectors through GMRES algorithm. This basis is successively extended and truncated to find the eigenvalues $\widetilde{\lambda}_l^{\mathrm{K}}$ of $\mathcal{C}^{\mathrm{K}^{-1}}$, which are linked to the eigenvalues λ_l^{K} of $\mathcal{C}^{\mathrm{K}}$ through the relation $\lambda_l^{\mathrm{K}} = \widetilde{\lambda}_l^{\mathrm{K}^{-1}}$. Both $\mathcal{C}^{\mathrm{K}^{-1}}$ and $\mathcal{C}^{\mathrm{K}}$ have the same eigenvectors. The eigenvalues λ_l of $\mathcal{B}$ can then directly be recovered from the λ_l^{K} by using the K formulation properties and the fact that $\overline{\mathcal{C}^{-1}} = \overline{\mathcal{C}}^{-1}$:

$$\lambda_l = \begin{cases} \lambda_l^{\mathrm{K}} + s & \text{if } \lambda_l^{\mathrm{K}} \in \mathrm{Sp}(\mathcal{C}), \\ \lambda_l^{\mathrm{K}} + \overline{s} & \text{otherwise.} \end{cases} \tag{III.106}$$

To discriminate one of the two possibles values, a test must be carried out. By using the results of III.4.3 and provided the eigenvector $\boldsymbol{v}_l \in \mathbb{C}^{2N}$ associated to λ_l^{K}, a test that compares the quantities $\|\boldsymbol{v}_l^o + i\boldsymbol{v}_l^e\|_2$ and $\|\boldsymbol{v}_l^o - i\boldsymbol{v}_l^e\|_2$ may be implemented. However, based on the author's experience, the relations $\boldsymbol{v}_l^o = -i\boldsymbol{v}_l^e$ and $\boldsymbol{v}_l^o = i\boldsymbol{v}_l^e$ do not necessarily hold when the eigenvector is not perfectly converged. We choose instead to focus on the residual associated to an eigenvector of $\mathcal{B}$ extracted from $\boldsymbol{v}_l$. To do so, let us consider $\boldsymbol{v}_l^o$ the eigenvector to test and the quantities $\epsilon_1 = \|\mathcal{B}\boldsymbol{v}_l^o - (\lambda + s)\boldsymbol{v}_l^o\|_2$ and $\epsilon_2 = \|\mathcal{B}\boldsymbol{v}_l^o - (\lambda + \overline{s})\boldsymbol{v}_l^o\|_2$. The lowest value between ϵ_1 and ϵ_2 indicates the correct value of λ_l in equation (III.106) and gives the residual associated with the operator $\mathcal{B}$. Hence, the KS algorithm is entirely implemented in K-formulated shift-and-invert,

Algorithm 6 K-formulated KS algorithm for modal analysis

Input: $\mathcal{B}$, $s \in \mathbb{C}$, $\boldsymbol{v}_1 \in \mathbb{R}^{2N}$ unitary, m and $l \leq m$ the number of desired eigenpairs
Output: j eigenpairs of $\mathcal{B}$, with $j \leq l$

initialization of $\mathcal{V}_m \leftarrow \begin{bmatrix} \boldsymbol{v}_1 & 0 \end{bmatrix}$, $j \leftarrow 0$, $p \leftarrow 0$
while $j \leq l$ until convergence **do**
  $m - p$ Arnoldi steps with deflation using preconditioned GMRES:

$$(\mathcal{B} - s\mathcal{I}_N)^{\mathrm{K}^{-1}} \widetilde{\mathcal{V}}_m = \widetilde{\mathcal{V}}_m \mathcal{H}_m + \boldsymbol{v}_{m+1} \hat{\boldsymbol{b}}_m^T$$

  real Schur decomposition with sorting $\mathcal{H}_m \leftarrow \mathcal{W}_2^T \mathcal{W}_1^T \mathcal{H}_m \mathcal{W}_1 \mathcal{W}_2$
  $\mathcal{V}_m \leftarrow \mathcal{V}_m \mathcal{W}_1 \mathcal{W}_2$
  for $k \in [\![j, l]\!]$ **do**
    compute $(\widetilde{\lambda}_k^{\mathrm{K}}, \boldsymbol{y}_k)$ verifying $\mathcal{H}_m \boldsymbol{y}_k = \widetilde{\lambda}_k^{\mathrm{K}} \boldsymbol{y}_k$ and set $\boldsymbol{v}_k = \widehat{\mathcal{V}}_m \boldsymbol{y}_k$
    compute $\epsilon_1 = \|\mathcal{B}\boldsymbol{v}_k^o - (\widetilde{\lambda}_k^{\mathrm{K}^{-1}} + s)\boldsymbol{v}_k^o\|_2$
    compute $\epsilon_2 = \|\mathcal{B}\boldsymbol{v}_k^o - (\widetilde{\lambda}_k^{\mathrm{K}^{-1}} + \overline{s})\boldsymbol{v}_k^o\|_2$
    set $\epsilon_{\mathrm{rel}}^{\mathrm{KS}} = \min(\epsilon_1, \epsilon_2)/(|\lambda_k| \|\boldsymbol{v}_k^o\|_2)$
    if $\epsilon_{\mathrm{rel}}^{\mathrm{KS}} <$ `tol` **then**
      lock $(\lambda_k, \boldsymbol{y}_k)$ and continue
    else
      break
    end if
  end for
  $j \leftarrow k$
  choose p, truncate and set $\boldsymbol{v}_{p+1} \leftarrow \boldsymbol{v}_{m+1}$
  compute $\boldsymbol{b}_p$ and insert it into $\mathcal{H}_p$
end while

but the convergence test is done on the original operator instead of the K-formulated one. The implementation is given in algorithm 6. It is worth noting that such a test is associated to an eigenpair of $\mathcal{B}$ instead of $\mathcal{C}^{-1}$, which might give too optimistic bounds and provide an eigenvalue of $\mathcal{B}$ that is not really converged [82].

Resolvent analysis. The shift-and-invert transformation is not used here, but the operator of interest still involves inverted operators. As stated in II.5.3, the resolvent analysis can be achieved by solving the eigenproblem associated to one of the operator $\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$ or $\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$. When solving the eigenproblem associated with $\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1\mathcal{R}$, the resulting eigenvectors are the optimal forcings and, as seen with relation (II.57), each optimal response's computation involve an inversion. In this work, the focus is made on the operator $\mathcal{C} = \mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$, whose eigenvectors provide the optimal responses, from which the optimal forcings can be recovered by using a simple matrix-vector product with the operator $(i\omega\mathcal{I}_N - \mathcal{J})$. Based on what we know about the K formulation, the expression of $\mathcal{C}^{\mathrm{K}}$ reads:

$$\mathcal{C}^{\mathrm{K}} = \mathcal{R}^{\mathrm{K}}\mathcal{Q}_2^{K^{-1}}\mathcal{R}^{H^{\mathrm{K}}}\mathcal{Q}_1^{\mathrm{K}} = (-\mathcal{J} - i\omega\mathcal{I})^{\mathrm{K}^{-1}}\mathcal{Q}_2^{\mathrm{K}^{-1}}(-\mathcal{J}^T + i\omega\mathcal{I})^{\mathrm{K}^{-1}}\mathcal{Q}_1^{\mathrm{K}}. \tag{III.107}$$

From [60], this operator is also Hermitian and has the same eigenvalues as $\mathcal{C}$, each with doubled multiplicity[11]. Moreover, we know that a real matrix with real eigenvalues have complex eigenvectors. However, we ensure the eigenvectors are real by operating with real arithmetics,

Algorithm 7 K-formulated KS algorithm for resolvent analysis

Input: $\mathcal{J}$, $\mathcal{J}^T$, $\mathcal{Q}_1$, $\mathcal{Q}_2$, $\omega \in \mathbb{R}$, $\boldsymbol{v}_1 \in \mathbb{R}^{2N}$ unitary, m and $l \leq m$ the number of desired eigenpairs
Output: j eigenpairs of $\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$, with $j \leq l$

initialization of $\mathcal{V}_m \leftarrow \begin{bmatrix}\boldsymbol{v}_1 & 0\end{bmatrix}$, $j \leftarrow 0$, $p \leftarrow 0$
while $k \leq l$ **do**
  $m - p$ Arnoldi steps with deflation using preconditioned GMRES:

$$(-\mathcal{J} - i\omega\mathcal{I})^{\mathrm{K}^{-1}}\mathcal{Q}_2^{K^{-1}}(-\mathcal{J}^T + i\omega\mathcal{I})^{\mathrm{K}^{-1}}\mathcal{Q}_1^{\mathrm{K}}\widetilde{\mathcal{V}}_m = \widetilde{\mathcal{V}}_m\mathcal{H}_m + \boldsymbol{v}_{m+1}\hat{\boldsymbol{b}}_m^T$$

  real Schur decomposition with sorting $\mathcal{H}_m \leftarrow \mathcal{W}_2^T\mathcal{W}_1^T\mathcal{H}_m\mathcal{W}_1\mathcal{W}_2$
  $\mathcal{V}_m \leftarrow \mathcal{V}_m\mathcal{W}_1\mathcal{W}_2$
  for $k \in [\![j, l]\!]$ **do**
    compute $(\widetilde{\eta}_k, \boldsymbol{y}_k)$ verifying $\mathcal{H}_m\boldsymbol{y}_k = \widetilde{\eta}_k\boldsymbol{y}_k$ and set $\epsilon_{\mathrm{rel}}^{\mathrm{KS}} = |\hat{\boldsymbol{b}}_m^T\boldsymbol{y}_k|/(|\widetilde{\eta}_k|\|\widehat{\mathcal{V}}_m\boldsymbol{y}_k\|_2)$
    if $\epsilon_{\mathrm{rel}}^{\mathrm{KS}} <$ `tol` **then**
      lock $(\widetilde{\eta}_k, \boldsymbol{y}_k)$ and continue
    else
      break
    end if
  end for
  $j \leftarrow k$
  choose p, truncate and set $\boldsymbol{v}_{p+1} \leftarrow \boldsymbol{v}_{m+1}$
  compute $\boldsymbol{b}_p$ and insert it into $\mathcal{H}_p$
end while

[11] we recall that $\mathrm{Sp}(\mathcal{C}) \in \mathbb{R}$.

meaning if the eigenvalue is real, the corresponding eigenvector will also be. From proposition 4, by considering an eigenpair $(\boldsymbol{v}_l, \tilde{\eta}_l) \in \mathbb{R}^{2N} \times \mathbb{R}$ of $\mathcal{C}^{\mathrm{K}}$, the vector $\boldsymbol{v}_l^o + i\boldsymbol{v}_l^e$ is selected as an eigenvector of $\mathcal{R}\mathcal{Q}_2^{-1}\mathcal{R}^H\mathcal{Q}_1$ and the resulting optimal forcing is calculated by $\sqrt{\tilde{\eta}_l}^{-1}\mathcal{R}^{-1}(\boldsymbol{v}_l^o + \boldsymbol{v}_l^e)$. No spectral transformation is performed in this resolvent case. The error calculated from the Krylov decomposition (see equation (II.73)) directly provides the residual associated to the K-formulated eigenproblem. The calculation of the residual associated to $\mathcal{C}$ is avoided since it requires two inversions for each eigenpair to test. An error is thus committed because the convergence is based on the K-formulated problem. However, based on the author's experience, the error of an eigenpair of $\mathcal{C}^{\mathrm{K}}$ is usually of the same order of magnitude as the error associated to the extracted eigenpair of $\mathcal{C}$. An implementation of the K-formulated KS algorithm for resolvent analysis is presented in algorithm 7.

These adaptations of the KS algorithm to K formulation complete the description of the method to scale up global stability analysis to industrial configurations, denoted as K-Formulated Global Stability Method (KFGSM). In the next section, the KFGSM is tested to draw an exhaustive picture of the stability of the laminar cylinder flow, and the results are compared to the reference SLEPc eigensolver.

III.6 Global stability analysis of the laminar cylinder flow

The numerical experiments carried out on the laminar cylinder case to validate the KFGSM are described hereafter. The various eigenproblems that arise in the global stability analysis are solved for the base flows discussed in III.3.5, and the results are compared to those obtained from the reference eigensolver SLEPc [82]. The latter uses complex arithmetic, forms the operators explicitly through multicoloring (see III.3.4), and solves the complex linear systems through an LU factorization, while the KFGSM uses real arithmetic and the K formulation, and employs FGMRES-ILU(3) with an external tolerance of 10^{-6}. The relative tolerance for the eigenpairs for all calculations has been set to 10^{-4}. The KS basis includes 20 vectors for direct and adjoint calculations and 10 vectors for resolvent analysis. We also introduce the direct and adjoint perturbation velocities $\widehat{\boldsymbol{U}}$ and $\widehat{\boldsymbol{U}}^\dagger$.

III.6.1 Direct modal analysis

The eigenpairs of $\mathcal{J}$ were calculated by solving the eigenproblem in equation (II.45) using both SLEPc and elsA. The relative tolerance for the two solvers was set to $\epsilon_{\mathrm{rel}}^{\mathrm{KS}} = 10^{-4}$. The amplification rate σ, and Strouhal number $\mathrm{St} = 2\pi/\omega$, obtained for the least stable eigenvalues are shown as a function of Re, in fig. III.11a and III.11b, respectively. These values agree with those obtained by Gianetti and Luchini in [30]. The critical value of Re at which the global mode becomes unstable is $\mathrm{Re_c} = 47$, which is consistent with values found in the literature [29, 22, 30]. These results validate our numerical setup. The spatial structures of the streamwise and cross-stream perturbation velocities, normalized by the L^∞-norm of the perturbation velocity vector, are shown in fig. III.12. Coherent structures, with an anti-symmetry through x-z plane, can be seen to be temporally and spatially amplified with a spatial wavelength of $\Lambda_x \simeq 6.3L$, and advected downstream. The von Kármán vortex street appears when the Reynolds number is increased beyond $\mathrm{Re_c}$, due to this unstable mode, which denotes a linearly amplified mechanism. The mode having a nonzero frequency, it is oscillating and thus denotes a so-called Hopf bifurcation[12]. These

[12] unlike a regular bifurcation in which the mode is steady.

linear computations are confronted to nonlinear unsteady ones in III.7.

Moreover, a good match is observed between the eigenvalues computed with elsA and SLEPc. This result validates the implementation of the K-formulated KS algorithm in elsA using real arithmetic, which returns the same eigenvalues as the complex reference eigensolver SLEPc. Table III.5 also shows that elsA returns the same eigenvalue using 1 or 8 CPUs, demonstrating its ability to operate in a massively parallel environment. However, in this particular case, the SLEPc solver is much faster than elsA. This was expected as our solver, designed for industrial configurations, has a much higher arithmetic intensity than the SLEPc counterpart used here, which relies on LU factorizations. This SLEPc strategy may be used for 2-D cases and small 3-D ones, but is unable to handle 3-D configurations with a high number of DOF, as discussed in III.1.1.

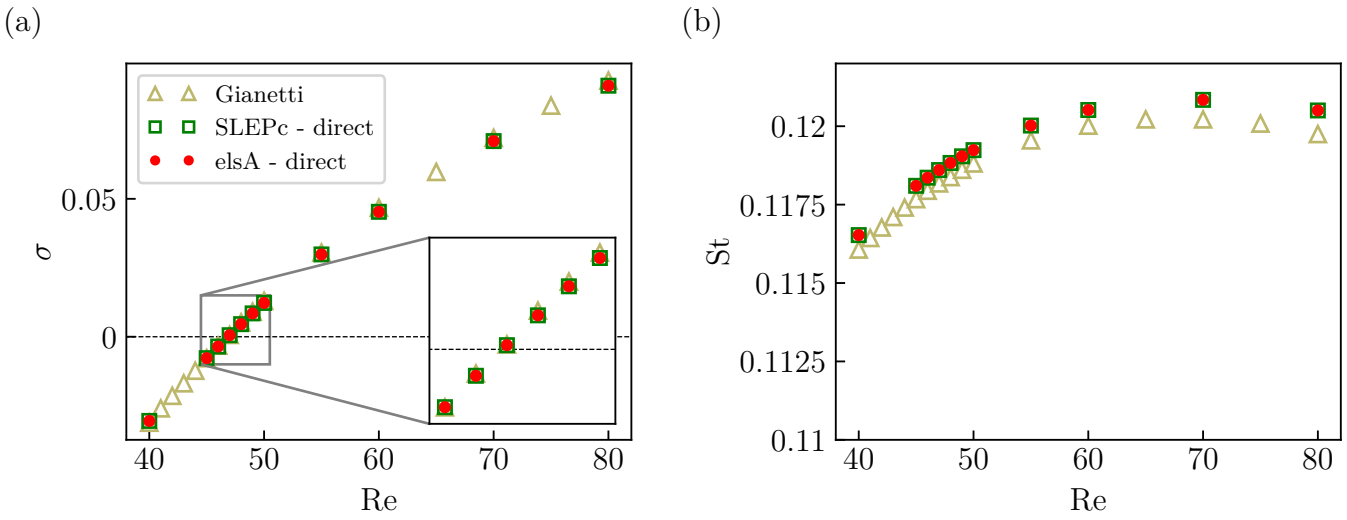


Figure III.11: Evolutions of amplification rate (a) and Strouhal number (b) according to Re for direct modal analysis. The critical Reynolds number is $\mathrm{Re_c} = 47$.

Solver	SLEPc - 1 CPU	elsA - 1 CPU	elsA - 8 CPUs
Allocated memory after computation	3.34 GB	3.55 GB	3.52 GB
CPU time	78 s	6204 s	1895 s
KS iteration	8	10	10
Eigenvalue	$5.62 \times 10^{-4} + 0.745i$	$5.616 \times 10^{-4} + 0.745i$	$5.616 \times 10^{-4} + 0.745i$
$\epsilon_{\mathrm{rel}}^{\mathrm{KS}}$	8.2×10^{-5}	2.1×10^{-5}	2.35×10^{-6}

Table III.5: Comparison between SLEPc an elsA computation for direct global stability analysis, computation at $\mathrm{Re} = 47$.

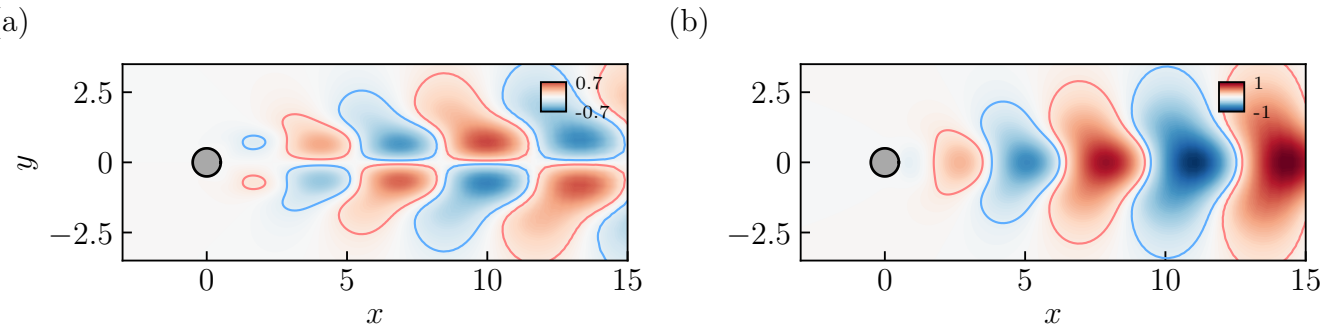


Figure III.12: Real part of streamwise (a) and cross-stream (b) velocity perturbation of the direct unstable mode at $\mathrm{Re} = 47$. Solid lines indicate iso-contours at $\Re(\widehat{u}_x) \pm 0.1$ and $\Re(\widehat{u}_y) \pm 0.1$.

III.6.2 Adjoint modal analysis

The same numerical experiments as in the direct eigenproblem are shown for the adjoint one (see (II.49)). The mass matrix considered is the $5N$-by-$5N$ matrix $\mathcal{Q}_\Omega$ (see II.5.2) of general term:

$$q_\Omega = \begin{pmatrix} V(\Omega) & 0 & 0 & 0 & 0 \\ 0 & V(\Omega) & 0 & 0 & 0 \\ 0 & 0 & V(\Omega) & 0 & 0 \\ 0 & 0 & 0 & V(\Omega) & 0 \\ 0 & 0 & 0 & 0 & V(\Omega) \end{pmatrix}. \tag{III.108}$$

The least stable eigenpairs was computed with both elsA and SLEPc eigensolvers. The evolutions of σ and St as a function of Re are presented in fig. III.13a and III.13b, respectively. A good agreement between direct and adjoint eigenvalues is observed, demonstrating the duality between of the direct and adjoint eigensolvers. The critical Reynolds number is also found to be $\mathrm{Re}_c = 47$. Fig. III.14 displays the spatial structure for the streamwise and cross-stream perturbation velocities, normalized by the L^∞-norm for the least stable mode at Re_c. Both shapes are consistent with previous results from the literature [87]. Unlike the direct mode, the adjoint is advected upstream. As stated by Sipp [22], this is due to the opposite signs of the advection operator in the adjoint and direct perturbation equation. The areas with the brightest colors correspond to the zones where the flow instability is the most sensitive to an external forcing. This information is useful to develop flow control strategy in order to delay the instability onset [22].

As for the direct eigenproblem, a good match is observed between elsA and SLEPc, which validates the K-formulated KS algorithm for the adjoint operator. Table III.6 presents detailed results after various computations. The eigenvalue returned by elsA using 1 or 8 CPUs is the same, further confirming the solver's ability to operate in a massively parallel environment. Additionally, it can be noted that the adjoint eigensolver implemented in elsA converges faster than the direct one, in agreement with Timme [55]. Again, SLEPc is faster than elsA for the reasons mentioned above.

Given both direct and adjoint unstable modes, and by analogy with equation (I.7), the structural sensitivity can be computed in its discrete form to describe the critical zones in the domain responsible for the instability (the so-called wavemaker). From [30, 29], it is defined as:

$$\boldsymbol{S}(\boldsymbol{x}) = \|\widehat{\boldsymbol{U}}(\boldsymbol{x})\|_2 \|\widehat{\boldsymbol{U}}^\dagger(\boldsymbol{x})\|_2, \tag{III.109}$$

where the L^2 norm is complex. The resulting map is shown in fig. III.15 and agrees with the results from [30]. It displays two dark parts around $x = 2.3$, indicating the precise origin of the

instability. This means that modifying the base flow outside these zones will have a marginal effect on the instability mode as long as these sensitive zones remain unaltered. In an industrial context, this information is valuable as it also provides insights into how to distribute meshing points economically while still capturing the instability.

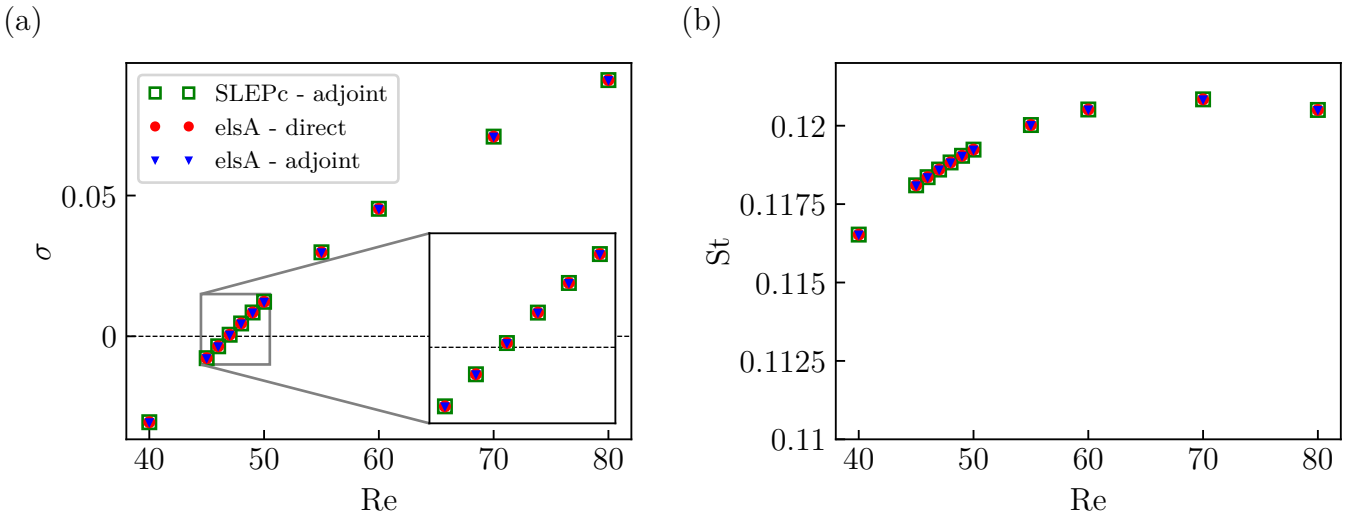


Figure III.13: Evolution of amplification rate (a) and Strouhal number (b) according to Re for adjoint modal analysis. The critical Reynolds number is $\mathrm{Re_c} = 47$.

Solver	SLEPc - 1 CPU	elsA - 1 CPU	elsA - 8 CPUs
Allocated memory after computation	3.34 GB	3.54 GB	3.61 GB
CPU time	74 s	4512 s	1022 s
KS iteration	8	10	10
Eigenvalue	$5.62 \times 10^{-4} + 0.745i$	$5.616 \times 10^{-4} + 0.745i$	$5.616 \times 10^{-4} + 0.745i$
$\epsilon_{\mathrm{rel}}^{\mathrm{KS}}$	6.6×10^{-5}	1.1×10^{-5}	6.2×10^{-5}

Table III.6: Comparison between SLEPc an elsA computation for adjoint global stability analysis computation at $\mathrm{Re} = 47$.

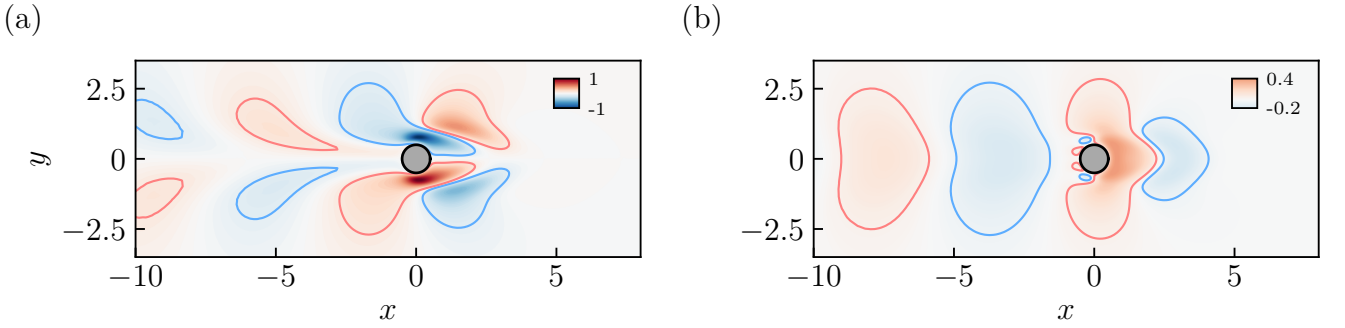


Figure III.14: Real part of streamwise (a) and cross-stream (b) velocity perturbation of the adjoint unstable mode at $\mathrm{Re} = 47$. Solid lines indicate iso-contours at $\Re(\widehat{u}_x) \pm 0.1$ and $\Re(\widehat{u}_y) \pm 0.05$.

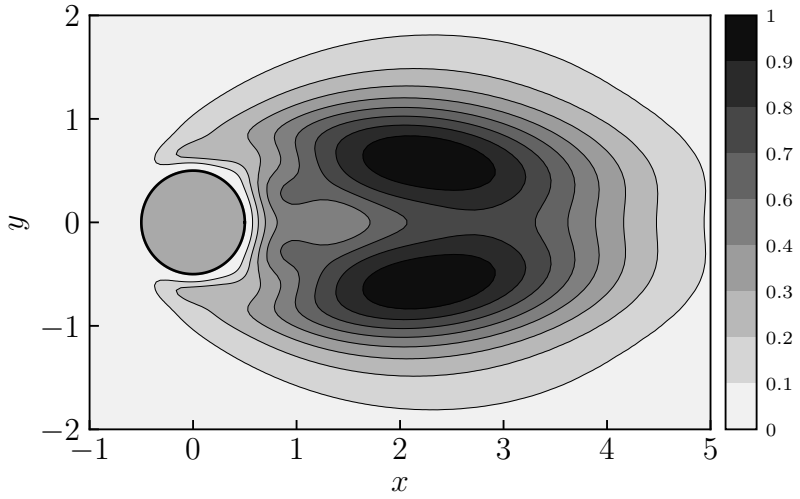


Figure III.15: Structural sensitivity of the laminar cylinder flow at Re = 40. The dark zones indicates the wavemaker.

III.6.3 Resolvent analysis

The KFGSM was ultimately tested on the resolvent analysis for the subcritical Reynolds number Re = 40. The energetic gain was optimized with respect to Chu's energy [83] for which the mass matrix $\mathcal{Q}_{\text{Chu}}$ is of general term:

$$q_{\text{Chu}} = \frac{V(\omega)}{\rho^2} \begin{pmatrix} \|\boldsymbol{u}\|_2^2 & -u_x & -u_y & -u_z & 0 \\ -u_x & 1 & 0 & 0 & 0 \\ -u_y & 0 & 1 & 0 & 0 \\ -u_z & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{III.110}$$

Fig. III.16 shows the two optimal gains obtained at this regime. A peak at St = 0.117 is observed, corresponding to the frequency of the least stable mode at this regime. This most amplified frequency corresponds to that of the vortex shedding observed when the von Kàrmàn instability is triggered. Spatial structures of streamwise and cross-stream velocities for $\widehat{\boldsymbol{\Phi}}_{\text{opt}}$ and $\widehat{\boldsymbol{W}}_{\text{opt}}$ are presented in fig. III.17. As expected, the optimal forcing has the same shape as the adjoint mode of fig. III.14, while the optimal response is similar to the direct mode of fig. III.12.

The eigenvalues calculated with elsA match well with those obtained with SLEPc. Table III.7 shows the results of various numerical experiments at St = 0.1165. Although SLEPc is faster than elsA, it requires more memory due to the need to compute and store LU factorizations of $\mathcal{R}$ and $\mathcal{R}^H$, while elsA only stores their ILU(3) factorizations, which are less expensive. In addition, SLEPc must store the mass matrix, whereas elsA never forms the matrix explicitly, using only matrix-vector products to calculate the actions of $\mathcal{Q}_\Omega^{-1}$ and $\mathcal{Q}_{\text{Chu}}$ on a given vector. The results of the numerical experiments show that elsA can work in a massively parallel environment, with the same eigenvalues being obtained using 1 or 8 CPUs.

To ensure the accuracy of the eigenpairs, it is important to set the external tolerance of GMRES as low as possible, with a maximal value of 10^{-6}. This is to ensure the validity of the product $(\mathcal{R}\mathcal{Q}_\Omega^{-1}\mathcal{R}^H\mathcal{Q}_{\text{Chu}})^{\text{K}}$ with a given vector. The KS algorithm is efficient in finding the largest eigenvalues, which is why a large KS basis and multiple iterations may not be necessary in the resolvent case. Indeed, the largest eigenvalue is generally much higher than the other ones and, based on the

author's experience, the algorithm usually converges it in one or two iterations, using a relatively small KS basis. This contrasts with direct and adjoint calculations, which use shift-and-invert without knowledge *a priori* of the eigenvalue, and may require a larger basis and more iterations

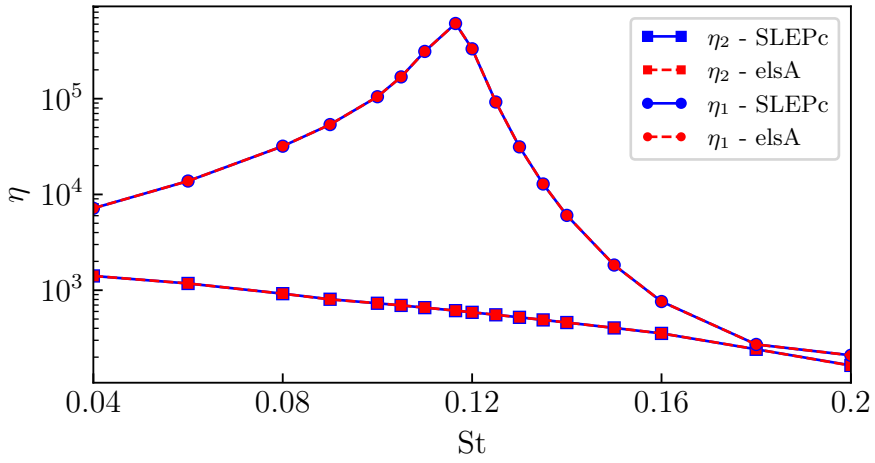


Figure III.16: Optimal gain obtained for the resolvent analysis at Re = 40.

Solver	SLEPc - 1 CPU	elsA - 1 CPU	elsA - 8 CPUs
Allocated memory after computation	7.12 GB	4.59 GB	4.58 GB
CPU time	51s	4705 s	1116 s
KS iteration	1	1	1
Eigenvalue	609481	609482	609482
$\epsilon_{\text{rel}}^{\text{KS}}$	6.7×10^{-15}	8×10^{-23}	8×10^{-23}

Table III.7: Comparison between SLEPc an elsA computation for resolvent analysis computation at Re = 40 and St = 0.1165.

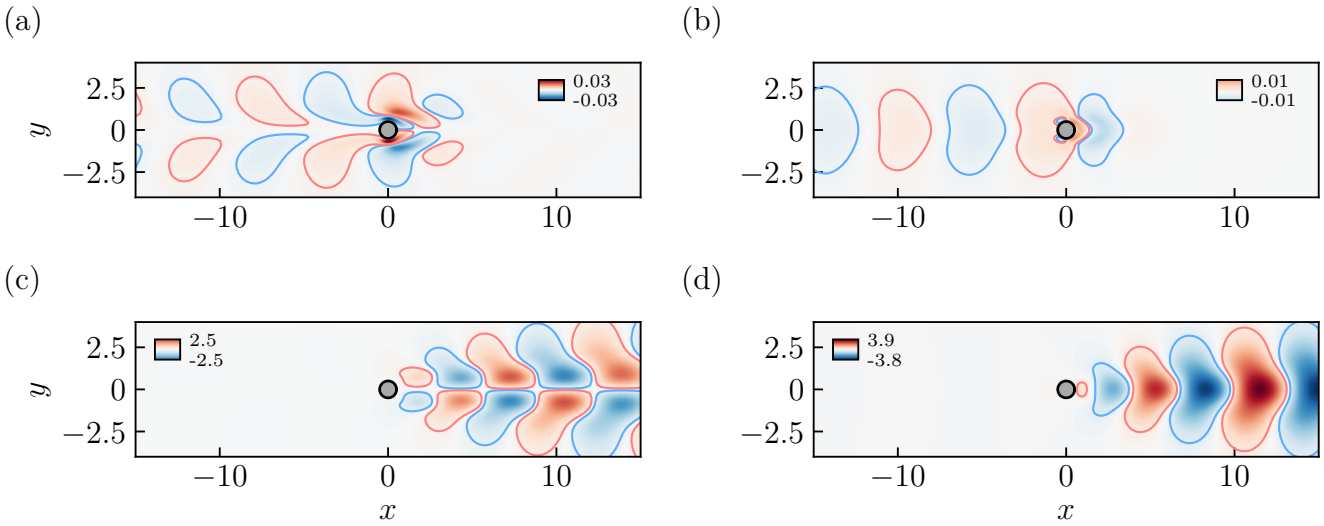


Figure III.17: Real part of streamwise and cross-stream perturbation velocity of the optimal forcing (a-b) and response (c-d) adjoint unstable mode at Re = 40 and St = 0.117.

to converge the eigenpairs.

This section aimed at showing the validation of the KFGSM on both the reference eigensolver SLEPc, and the literature, for modal and nonmodal analyses. A common practice to compare the amplification mechanism in both linear and nonlinear frameworks is to perform nonlinear unsteady computations. In the next section, the second-order Gear method is described and unsteady computations are performed on the laminar cylinder case.

III.7 Unsteady computation

The nonlinear unsteady computation can be performed prior to a global stability analysis in order to determine the frequency of the unsteadiness. This information can then be utilized to choose the shift in a modal analysis or to compute the optimal forcing and response at this frequency through the resolvent analysis. The results of the global stability analysis can also be compared retrospectively with the unsteady computation to verify if the origin of a particular unsteadiness can be attributed to a linearly amplified mechanism. To carry out the temporal integration of the governing equations, the elsA software employs the Gear method, which is explained in more detail below. Unsteady computations have been performed on the laminar cylinder flow and validate the capability of the KFGSM to predict the onset of the von Kármán vortex street.

III.7.1 Gear method

Each unsteady computations in this work are carried out by solving the Gear scheme. The latter, described in [105], is a backward multistep method that is second-order accurate in time. It is associated with the following nonlinear implicit scheme:

$$\mathcal{G}(\boldsymbol{W}^{n+1}) = \frac{V(\Omega)}{\Delta t}\left(\frac{3}{2}\boldsymbol{W}^{n+1} - 2\boldsymbol{W}^{n} + \frac{1}{2}\boldsymbol{W}^{n-1}\right) - \boldsymbol{R}(\boldsymbol{W}^{n+1}) = 0. \tag{III.111}$$

Similarly to II.3.1, this scheme is equivalent to a Newton method:

$$\mathcal{J}_{\mathcal{G}}(\boldsymbol{W}^{n,l})\Delta\boldsymbol{W}^{n,l} = -\mathcal{G}(\boldsymbol{W}^{n,l}) \tag{III.112}$$

where l is the index of the current sub-iteration of the Newton algorithm. As for the LU-SGS method described in III.3.3, the Jacobian matrix is replaced by an approximated operator $\widetilde{\mathcal{J}}$ and the following system is solved at each nonlinear iteration:

$$\left(\frac{3}{2\Delta t}\mathcal{I}_N + \widetilde{\mathcal{J}}(\boldsymbol{W}^{n,l})\right)\Delta\boldsymbol{W}^{n,l} = \boldsymbol{R}(\boldsymbol{W}^{n,l}) \tag{III.113}$$

Note that $\widetilde{\mathcal{J}}$ is updated at each sub-iteration of the Newton resolution. The time-step Δt corresponds to the physical user-defined one.

III.7.2 Nonlinear unsteady computation of the laminar cylinder flow

The physical time-step for these computations has been set to $\Delta t = 10^{-3}$, so that the CFL number at the wall is approximately 10. The initial fields are the base flows presented in III.3.5, and perturbed by least stable mode computed using the direct eigensolver. Fig. III.18 displays

the time evolution of the lift coefficient C_L at $\mathrm{Re} = 55$. It grows exponentially until the nonlinear saturation leads to a limit cycle around $t = 190$, which represents the destabilization of the unstable steady base flow toward the von Kármán vortex street. This can be seen in fig. III.19, which shows isocontours of vorticity magnitude at $t = 300$. A spatial wavelength of $\Lambda_x \simeq 6.5L$ is also observed, which is of the same order as that identified for the direct unstable mode (see Fig. III.12).

The measured signal is analyzed in fig. III.20a and III.20b to extract the corresponding nonlinear amplification rate σ^{NL} and Strouhal number $\mathrm{St}^{\mathrm{NL}}$. Only one peak is identified in the magnitude spectrum, which is associated to the frequency of the vortex shedding. Table III.8 summarizes the results obtained for various Reynolds numbers, where $(\sigma^{\mathrm{L}}, \mathrm{St}^{\mathrm{L}})$ are the values resulting from the linear direct eigensolver. The growth rates from the linear analyses are consistent with those obtained from the linear stage of the nonlinear simulations. Differences in both the amplification rate and Strouhal number can be seen between the linear and nonlinear computations. As stated for instance in [106], while this discrepancy is normal since it is due to nonlinearities, the actual nonlinear frequency may nonetheless be recovered from the linear stability analysis by linearizing the equations about the mean flow instead of the fixed point. This is further discussed for instance in [38]. However, at relatively low Reynolds number, these results support the conclusion that the vortex shedding originates from a linearly amplified mechanism.

The unsteady computation is hence useful in identifying unsteadiness in the flow, but when multiple unsteady phenomena coexist, they interact nonlinearly and it is often difficult to isolate

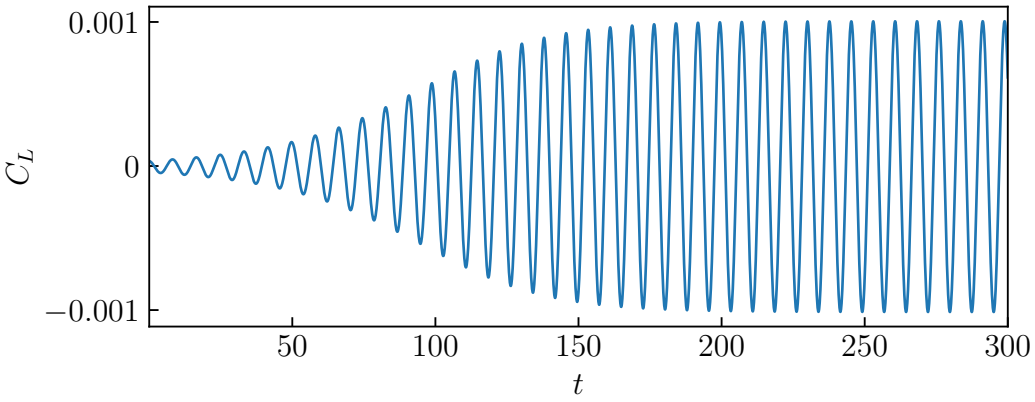


Figure III.18: Time evolution of lift coefficient at $\mathrm{Re} = 55$.

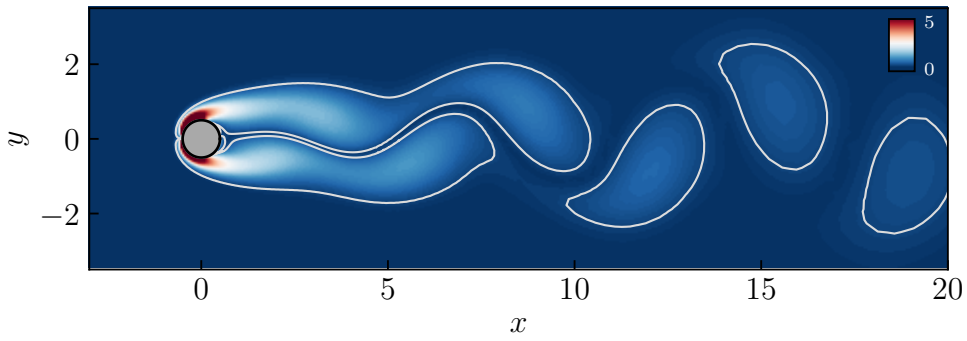


Figure III.19: Vorticity magnitude map at $\mathrm{Re} = 55$ and $t = 300$. The computation was initialized using the baseflow perturbed by the unstable mode. Solid lines indicate iso-contours at $\|\boldsymbol{\nabla} \times \boldsymbol{u}\|_2 = 0.175$.

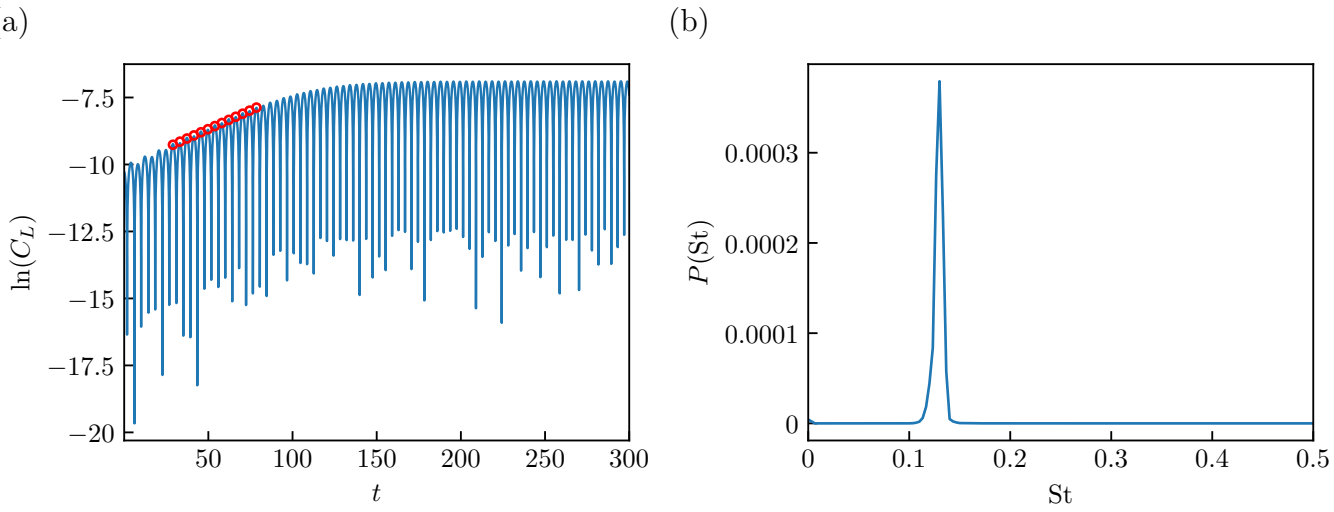


Figure III.20: (a) Lift coefficient trend over the time. The dotted line represents $\sigma^{\mathrm{NL}} = 0.028$. (b) Magnitude spectrum of the signal at Re = 55. A pike at $St^{\mathrm{NL}} = 0.13$ is identified.

Re	$(\boldsymbol{\sigma^{\mathrm{L}}, \mathrm{St}^{\mathrm{L}}})$	$(\boldsymbol{\sigma^{\mathrm{NL}}, \mathrm{St}^{\mathrm{NL}}})$	**ΔSt**	**Δσ**
48	$(0.0046, 0.1188)$	$(0.0043, 0.1205)$	1.3%	6.5%
55	$(0.03, 0.12)$	$(0.028, 0.13)$	7.7%	7.1%
60	$(0.045, 0.1205)$	$(0.04, 0.1366)$	11.8%	12.5%

Table III.8: Comparison between SLEPc an elsA computation for resolvent analysis computation at Re = 40 and St = 0.1165.

them. Additionally, the required computation time to reach the limit cycle can be quite long. This is why global stability analysis is useful, as it allows for the isolation of different mechanisms in a reasonable amount of computation time.

III.8 Conclusion

This chapter first presents the fixed point methods, including preconditioning computation, as implemented in elsA. Then, a mathematical foundation for the KFGSM is established. The code is validated through testing on the laminar cylinder case. In chapter III, the KFGSM is applied to more complex configurations, including the turbulent OAT15A airfoil flow in transonic conditions, which is a challenging RANS case, and the 3-D laminar subsonic sphere flow, which contains a large number of DOF. Bifurcations of axisymmetric and planar-symmetric base flows are successfully characterized.

Chapter IV

Challenging test cases

Contents

IV.1 Introduction

Chapter II aims to introduce general concepts related to compressible CFD and global stability. In chapter III, the KFGSM is described and validated on the canonical case of the laminar cylinder flow to predict the emergence of the von Kármán vortex street. In this chapter, the capability of the KFGSM to handle more challenging cases is demonstrated. The first case studied is the RANS OAT15A airfoil flow, which destabilizes through a Hopf bifurcation resulting in the buffet phenomenon, an interaction between the boundary layer and a shock wave located on the suction side of the airfoil [41]. The linear systems involved in both fixed point computation and global stability analysis are known to be particularly stiff [51].

The second case considered is the 3-D subsonic laminar sphere flow. This flow has two bifurcations: a regular one around $\mathrm{Re} = 210$ that breaks one symmetry, and a second one around

$\mathrm{Re} = 280$ which is a Hopf bifurcation, responsible for the degeneration towards hairpin-like vortex shedding [107]. The computational grid for this case consists of about 10×10^6 cells, resulting in 50×10^6 DOF. For the super-critical Hopf bifurcation of the planar-symmetric base flow, a refined mesh consisting of about 125×10^6 DOF was used. Such sizes are representative of those of industrial configurations that the code was designed for. This demonstrates the capability of the KFGSM to operate on such configurations where direct factorizations are not feasible due to memory limits. As a reminder, the largest known global stability analysis conducted by Timme to characterize the buffet phenomenon on an entire plane was performed on 50×10^6 DOF [55].

IV.2 OAT15A airfoil case

Under given Mach number and angle of attack, the shock on the suction side of a plane wing may interact with the detached boundary layer and gives rise to the so-called unsteady buffet phenomenon [108]. This causes the shock to oscillate through a self-sustained mechanism, and may lead to severe damage, mainly due to fatigue. Crouch first showed in [109] that this mechanism could be predicted on a 2-D airfoil through the use of global stability analysis. Sartor conducted such an analysis on the OAT15A airfoil as described in [41], and Paladini [27] extended the study to 3-D unswept wing, exhibiting buffet cells on the boundary layer. In this section, we focus on the 2-D supercritical OAT15A airfoil to validate the KFGSM on this particularly stiff RANS case [51]. As for the laminar cylinder flow, fixed point computation as well as direct, adjoint, and resolvent analysis are detailed below.

IV.2.1 Numerical set-up

The computational domain presented in fig. IV.1 is an unstructured C-grid composed of 70091 hexaedral cells. The first mesh point in the boundary layer on the profile is always below $y^+ = 0.9$, and considering the chord length c as the characteristic dimension, the grid definition in the shock region is $\Delta x/c = 0.003$. The selected physical parameters are given in table IV.1. To improve the resolution of linear systems during fixed point computations or global stability analysis, the problem is undimensioned using stagnation density $\rho_{i,\infty}$, stagnation temperature $T_{i,\infty}$, and the associated sound-velocity $a_{i,\infty}$, which means that these quantities are all unitary. Such a modification of the original problem is a form of preconditioning that improves the condition

Free-stream Mach number	$\mathrm{M}_\infty = 0.73$
Free-stream stagnation temperature	$T_{i,\infty} = 288$ K
Free-stream stagnation pressure	$p_{i,\infty} = 101325$ Pa
Free-stream stagnation density	$\rho_{i,\infty} = 1.226\ \mathrm{kg\,m^{-3}}$
Reynolds number	3.2×10^6
Angle of attack	$\alpha \in [3.75°, 6°]$

Table IV.1: Flow parameters of the 2-D OAT15A airfoil.

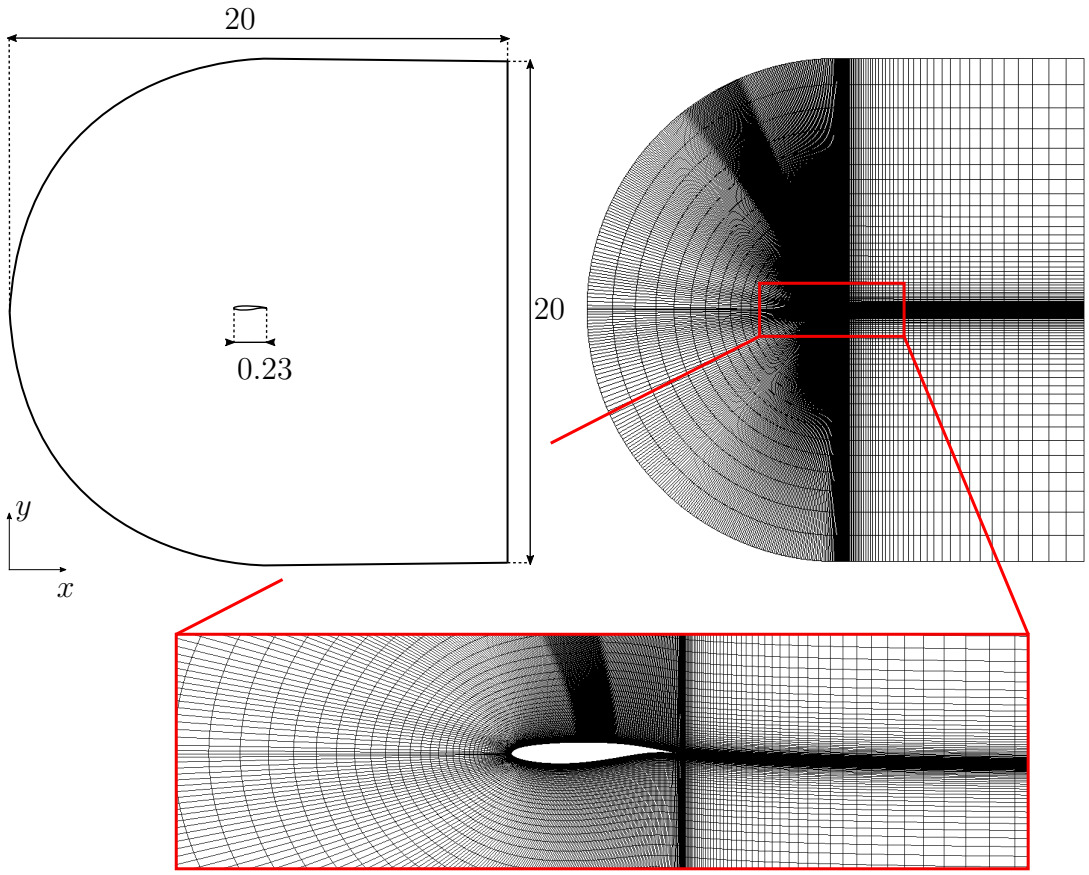


Figure IV.1: Computational domain for the two-dimensional OAT15A airfoil. It expands over $100c$.

number of the linear systems. Other physical quantities are obtained with the following relations:

$$a_\infty = \sqrt{\left(1 + \frac{\gamma - 1}{2} \mathrm{M}_\infty^2\right)^{-1}}, \tag{IV.1}$$

$$p_\infty = \left(1 + \frac{\gamma - 1}{2} \mathrm{M}_\infty^2\right)^{\frac{-\gamma}{\gamma - 1}}, \tag{IV.2}$$

$$\rho_\infty = \left(1 + \frac{\gamma - 1}{2} \mathrm{M}_\infty^2\right)^{\frac{-1}{\gamma - 1}}, \tag{IV.3}$$

$$e_\infty = \frac{1}{\gamma(\gamma - 1)} \left(1 + \frac{\gamma - 1}{2} \mathrm{M}_\infty^2\right)^{-1}, \tag{IV.4}$$

$$R = \frac{1}{\gamma}, \tag{IV.5}$$

$$\mu_\infty = \frac{c \rho_\infty \mathrm{M}_\infty a_\infty}{\mathrm{Re}_\infty}. \tag{IV.6}$$

The angle of attack α is taken in to account by imposing at the farfield boundaries the streamwise $u_{x,\infty} = u_\infty \cos(\alpha)$ and cross-stream $u_{y,\infty} = u_\infty \sin(\alpha)$ velocities.

Concerning the numerical methods, the negative SA scheme presented in III.2.2 is used to model turbulent effects. Convective fluxes are discretized using the approximate Riemann solver HLLC scheme as described in [63], with second-order accuracy without a limiter function. The

diffusive fluxes, as in the case of the laminar cylinder case, are discretized by computing the gradient at faces using the $5p_{\text{cor}}$ method (see III.3.2). As for laminar cylinder flow (see III.1.2), the boundary conditions are set to no-slip velocity at the wall, non-reflective conditions in the farfield, and symmetry conditions on the transverse faces of the domain.

IV.2.2 Fixed point computation

Under unstable regimes, computing a fixed point in this case is particularly difficult. In fact, using the backward Euler scheme with GMRES, as presented in III.3.4, initialized with a uniform field fails and causes the calculation to diverge. We chose instead to first perform a LU-SGS computation (see III.3.3) initialized with a uniform field to get a non-converged field that can then be used to initialize the backward Euler scheme combined with the GMRES. Both simulations were run on 24 CPUs, and the convergence result for the LU-SGS method is presented in fig. IV.2a. The residual $\|\rho\|_2$ was reduced to 7×10^{-7} in 10^4 iterations.

For GMRES computations, the FGMRES-HLU method is employed due to the stiffness of the linear systems of interest (see III.3.4), with both the internal and external Krylov bases consisting of 60 vectors. The internal and external tolerances are set to 10^{-3} and 10^{-4}, respectively. Fig. IV.2b displays the convergence curve obtained using the GMRES algorithm, which was initialized with the field returned by the LU-SGS computation. The final nonlinear residual $\|\rho\|_2$ was reduced to

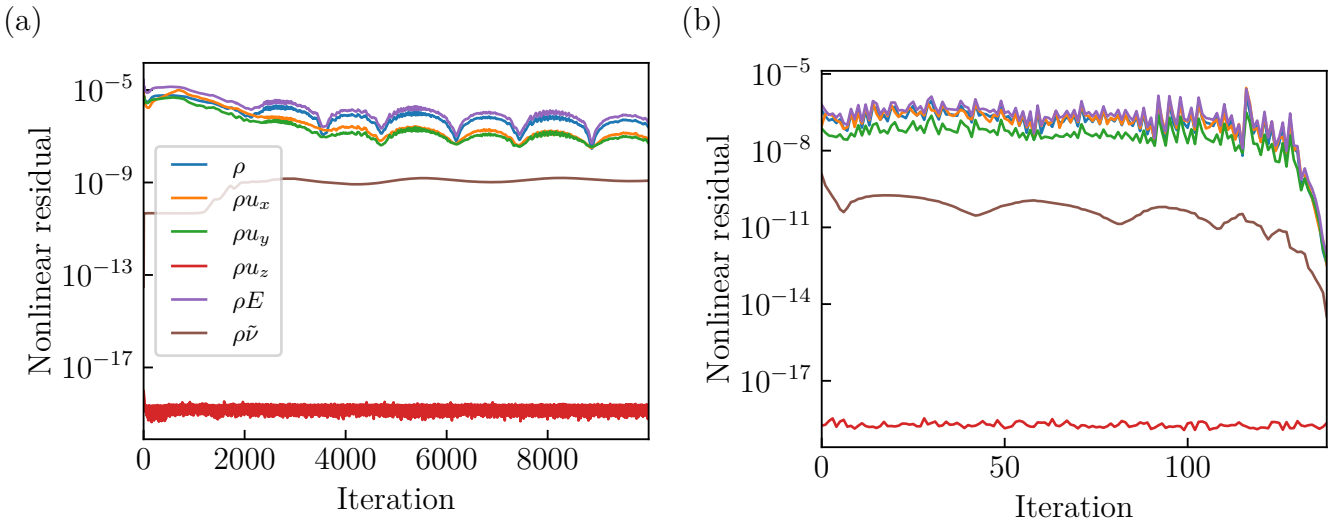


Figure IV.2: Convergence curve of LU-SGS method initialized with a uniform field (a) and FGMRES-ILU(3) method (b) at $\alpha = 4°$.

Solver	LU-SGS	GMRES
nonlinear iterations	10^4	139
Final $\|\rho\|_2$	7×10^{-7}	4×10^{-13}
Allocated memory after computation	0.59 GB	4.14 GB
CPU time	127 s	582 s

Table IV.2: Comparative figures between LU-SGS and FGMRES-ILU(3) method for fixed point computation at $\alpha = 4°$.

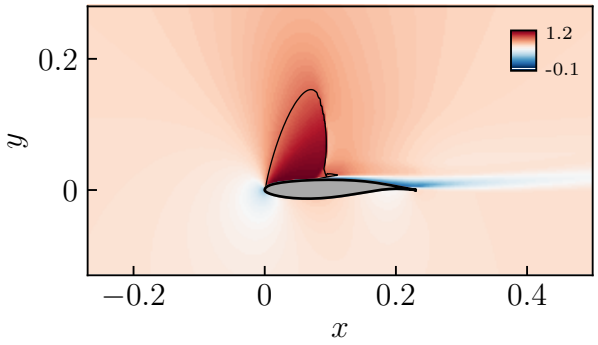


Figure IV.3: Streamwise velocity of the base flow at $\alpha = 4°$ case. Sonic line is indicated by the solid black line.

4×10^{-13} after 139 iterations. The horizontal velocity of the corresponding base flow is illustrated in fig. IV.3, where the sonic line is indicated. The contour map is in good agreement with the results reported by Sartor in [41]. Table IV.2 provides quantitative data to compare the two methods. As expected, the memory requirements for the GMRES method are significantly higher than those of the LU-SGS method, as the GMRES method solves the exact linear systems rather than approximate ones, requiring the formation of an effective preconditioner, which consumes a lot of memory.

Such results demonstrate the complementary nature of both fixed point computation methods and the efficacy of using them successively to compute a fixed point in a stiff case without prior knowledge of its shape. However, in an industrial setting, particularly for turbomachinery computations, the LU-SGS method may not converge at all when initialized with a uniform field. This issue can be addressed by approximating the initial field using the meridian model, as described in [110]. Based on these successful base flow computations, the focus is made in the following on the global stability analyses of this case.

IV.2.3 Direct modal analysis

With the fixed points computed with high accuracy, the KFGSM can now be tested on this configuration. The computations are still performed on 24 CPUs. The KS basis consists of 30 vectors and the relative tolerance for the eigenvalue is set to 10^{-4}. The inversion method used is the FGMRES-HLU with 2-level RAS [51], where both the internal and external Krylov bases consist of 60 vectors. The internal and external tolerances are set to 10^{-2} and 10^{-6}, respectively. For this case, the linear systems are so stiff that the real part of the shift was set to 5 to make the inversions operational. The imaginary part of the shift was set near the expected non-dimensional buffet frequency, which is related to the Strouhal number defined as follows:

$$\mathrm{St} = \frac{fc}{u_\infty}. \tag{IV.7}$$

The evolution of both σ and St with respect to α is presented in fig. IV.4a and IV.4b, respectively. The buffet onset is found to occur at $\alpha_c = 4°$ with the corresponding critical Strouhal number $\mathrm{St}_c = 0.062$, resulting in a buffet onset frequency around 64 Hz. The flow restabilizes between $\alpha = 5.5°$ and $\alpha = 6°$, which indicates the buffet offset. These results are in agreement with the results obtained by Sartor in [41, 111]. The contour maps of both velocity and density

of the direct unstable mode at α_c are shown in fig. IV.5. The most energetic part of the mode is located within the shock wave, and a smaller part is identified in the mixing layer.

As with the laminar cylinder flow, the elsA solver is compared to the SLEPc solver to validate the implementation of the K-formulated KS algorithm. A good match is observed for both σ and St. Table IV.3 provides quantitative figures to compare both solvers at α_c. The elsA computation requires much more memory than the SLEPc solver, which is not surprising since the GMRES

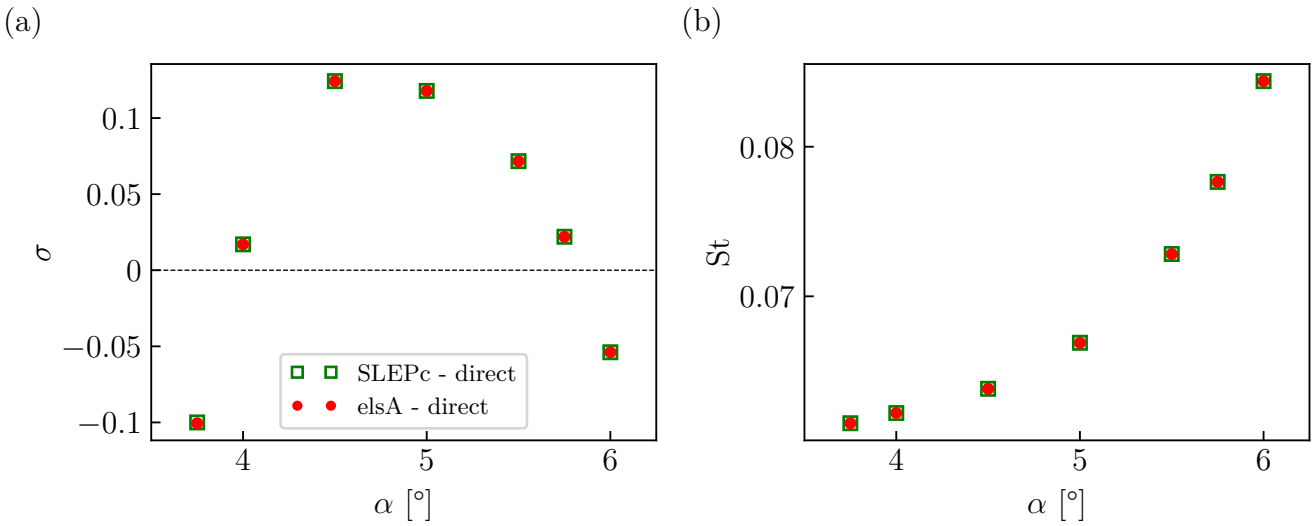


Figure IV.4: Evolutions of amplification rate (a) and Strouhal number (b) according to α for direct modal analysis. Critical angle of attack is found to be $\alpha_c = 4°$.

Solver	SLEPc - 1 CPU	elsA - 24 CPU
Allocated memory after computation	10.05 GB	17.84 GB
CPU time	248 s	18040 s
KS iteration	6	15
Eigenvalue	$0.017 + 1.12i$	$0.017 + 1.12i$
$\boldsymbol{\epsilon}_{\text{rel}}^{\text{KS}}$	4.9×10^{-5}	3.8×10^{-5}

Table IV.3: Comparison between SLEPc ans elsA computations for direct global stability analysis at $\alpha = 4°$.

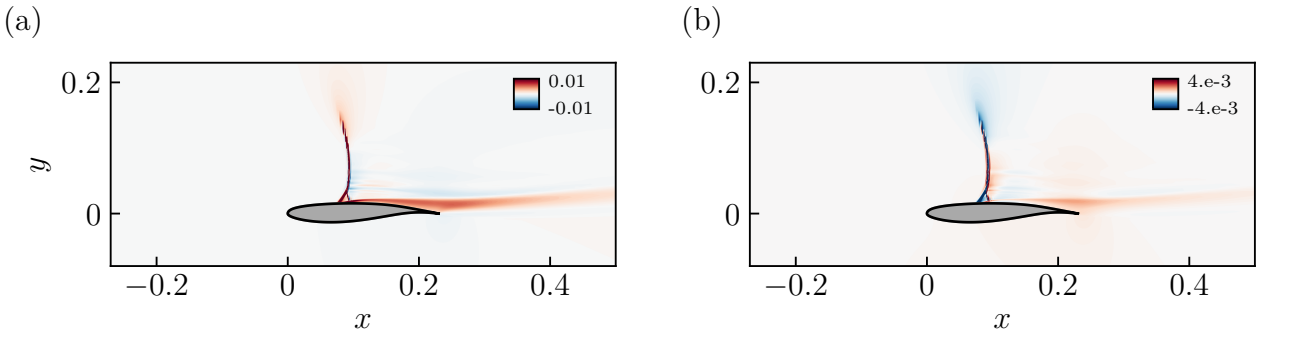


Figure IV.5: Real part of streamwise velocity perturbation (a) and density (b) for the direct unstable mode at $\alpha = 4°$.

inversions require many arrays to be allocated for the Krylov basis and the two-level overlapping preconditioning is memory-intensive. However, it should be noted that this case serves as a demonstration and is not representative of industrial cases for which it was designed. The direct modal analysis is hence validated for this RANS case, and the adjoint modal analysis is described in the following.

IV.2.4 Adjoint modal analysis

The adjoint operator is defined relative to the scalar product defined by the $6N$-by-$6N$ matrix $\mathcal{Q}_\Omega$ (see equation (III.108)). Similar to the laminar cylinder flow and in agreement with Timme [55], the adjoint problem is less stiff than the direct one, allowing the shifts to be set closer to the expected eigenvalues, resulting in fewer KS iterations for the solver to converge. As shown in table IV.4, both solvers only required one KS iteration with a tolerance of 10^{-4}. In comparison with the direct eigensolver, the memory requirements for the elsA adjoint solver are also 70% higher than those of SLEPc due to the HLU preconditioning with two RAS levels.

The evolution of both σ and St with respect to the angle of attack is shown in fig. IV.6 and compared with SLEPc. The close proximity of the eigenvalues confirms the validity of the adjoint eigensolver under RANS conditions. Furthermore, the eigenvalues computed using the

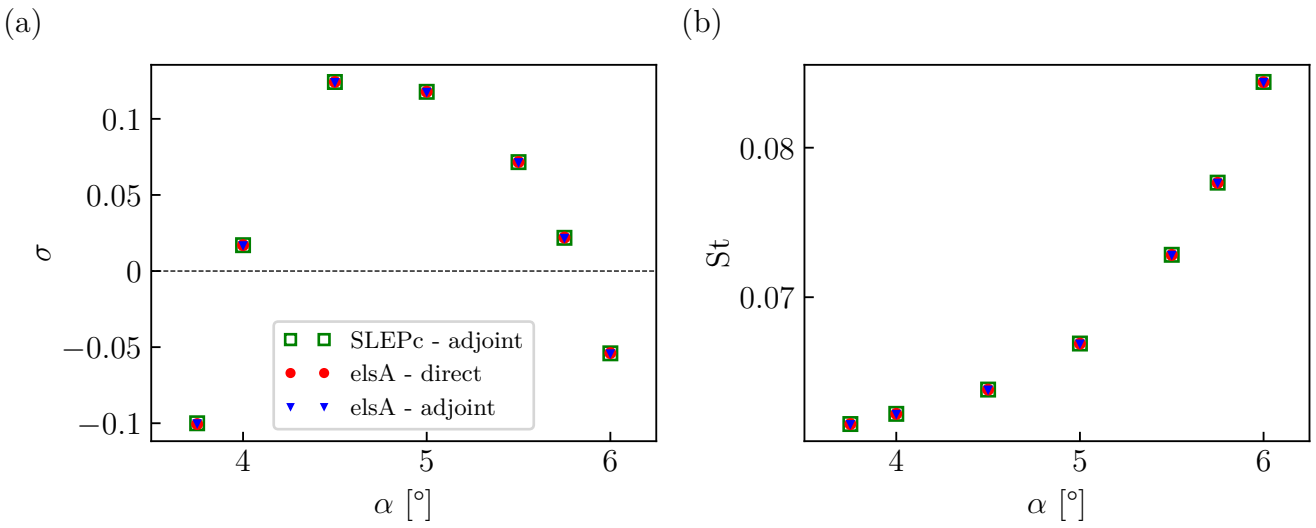


Figure IV.6: Evolutions of the growth rate (a) and Strouhal number (b) according to α for adjoint modal analysis. Critical angle of attack is found to be $\alpha_c = 4°$.

Solver	SLEPc - 1 CPU	elsA - 24 CPUs
Allocated memory after computation	10.07 GB	17.43 GB
CPU time	176 s	7391 s
KS iteration	1	1
Eigenvalue	$0.017 + 1.12i$	$0.017 + 1.12i$
$\boldsymbol{\epsilon}_{\text{rel}}^{\text{KS}}$	1.2×10^{-7}	2.51×10^{-5}

Table IV.4: Comparison between SLEPc and elsA computations for adjoint global stability analysis at $\alpha = 4°$.

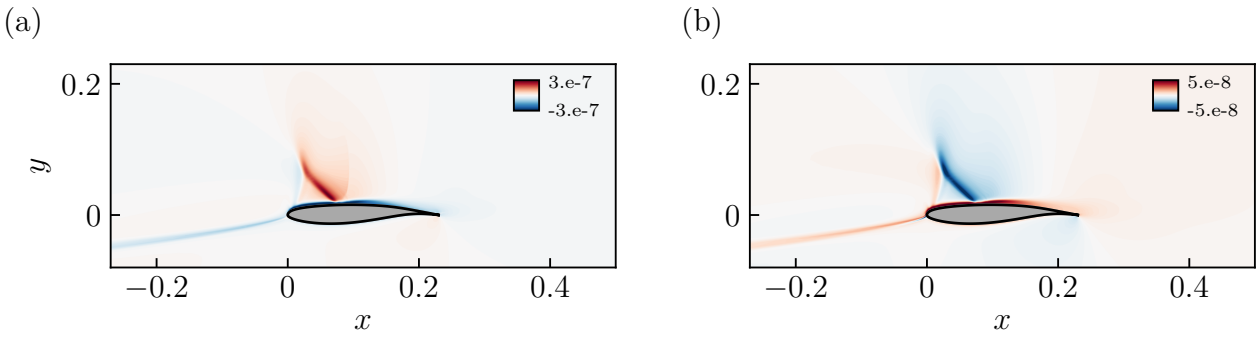


Figure IV.7: Real part of streamwise velocity perturbation (a) and density (b) for the adjoint unstable mode at $\alpha = 4°$.

adjoint solver are identical to those obtained using the direct eigensolver within a certain tolerance, demonstrating the duality between the two operators. The least stable adjoint mode calculated for α_c is displayed in fig. IV.7. In accordance with the findings of Sartor [41], the most energetic part of the adjoint is the oblique component which follows the characteristic line that impacts the suction side at the start of the separation region. This suggests a strong receptivity of the flow to external forcing in this area. As noted by Sartor, this position of the adjoint could indicate a significant influence of the start of the separation region on the flow's dynamics. Additionally, the placement of the adjoint indicates that a perturbation downstream of the airfoil could impact the instability, as a result of the propagation of upstream acoustic waves. The structural sensitivity is calculated using equation (III.109), and the contour map for α_c is shown in fig. The wavemaker is identified as having a lambda shape and is mainly located in the shock wave. As a result, the grid definition in this mesh zone should be carefully selected to capture the instability.

In agreement with the direct eigensolver, the KFGSM in its adjoint form is validated for RANS computations. This implies that the K-formulated resolvent operator can be assembled and made operational. In the following, a resolvent analysis of a sub-critical base flow is demonstrated, illustrating the flow's receptivity to the various frequencies calculated.

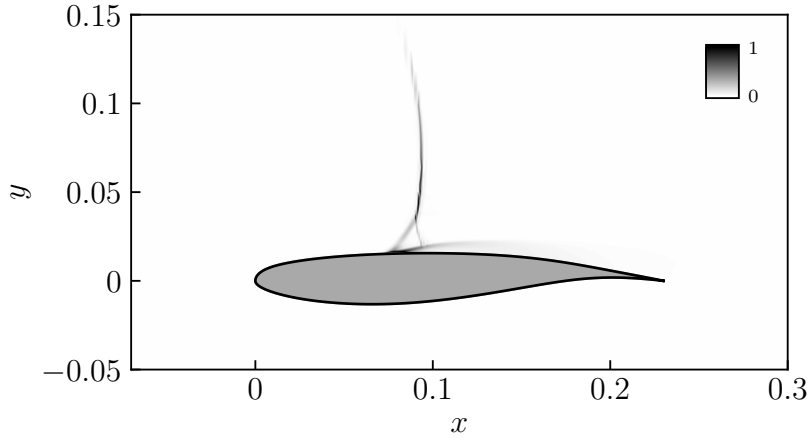


Figure IV.8: Structural sensitivity at $\alpha = 4°$.

IV.2.5 Resolvent analysis

The mass matrices are set to the $6N$-by-$6N$ matrices $\mathcal{Q}_{\text{Chu}}$ and $\mathcal{Q}_\Omega$ (see equations (III.110) and (III.108) respectively). The angle of attack is set to the sub-critical value of $\alpha = 3.75°$. The tolerance for GMRES inversions must be set lower than for the direct or adjoint eigensolver to ensure accuracy when performing matrix-vector products with the operator $\mathcal{R}^{\text{K}}\mathcal{Q}_\Omega^{\text{K}^{-1}}\mathcal{R}^H\mathcal{Q}_{\text{Chu}}^{\text{K}}$. The tolerance is set to 10^{-6} for all computations. The KS basis consists of 10 vectors, requiring 20 inversions for the first iteration and 10 for subsequent iterations, as only half of the basis is recycled through the KS iterations. This relatively small number of vectors compared to the size of the problem is due to the fact that the largest eigenvalue is typically well separated from the others, leading to rapid convergence.

The two highest gains computed for various frequencies with elsA are shown in fig. IV.9 and compared with SLEPc results. The results of are close to each other, but accuracy could be improved by carrying out the same analysis with more vectors in the KS basis. Table IV.5 provides a comparison of both solver computations. elsA require to store two HLU factorizations with two RAS level, which results in an extra 80% memory usage compared to SLEPc. The linear systems in resolvent analysis are stiff, and the relatively high tolerance set for GMRES convergence leads to a much higher CPU time than SLEPc, which operates with the faster LU resolution. This is consistent with the results from the direct and adjoint modal analysis.

From a physical perspective, the evolution of η_1 with respect to St shows two peaks. The

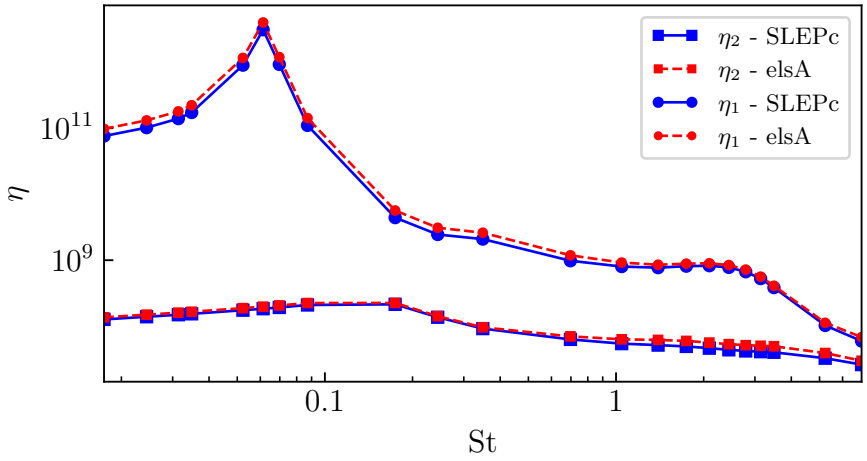


Figure IV.9: Optimal gain evolution according to St at $\alpha = 3.75°$. Two peaks are identified at St = 0.0613 (buffet) and St = 2.44 (Kelvin-Helmholtz).

Solver	SLEPc - 1 CPU	elsA - 24 CPUs
Allocated memory after computation	16.2 GB	29.7 GB
CPU time	144 s	14129 s
KS iteration	1	1
Eigenvalue	3.098×10^{12}	3.994×10^{12}
$\boldsymbol{\epsilon}_{\text{rel}}^{\text{KS}}$	2.4×10^{-13}	2.5×10^{-23}

Table IV.5: Comparison between SLEPc and elsA computations for resolvent analysis at $\alpha = 3.75°$ and St = 0.1165.

first and largest peak is observed at St = 0.0613, which corresponds to a dimensional frequency of 73.5 Hz. This frequency is the same as that of the least stable mode calculated with modal analysis (see fig. IV.4b) at this incidence. The optimal forcing and response are displayed in fig. IV.10. The contours of both streamwise velocity and density are similar to the direct and adjoint eigenmodes that correspond to the buffet phenomenon, in a similar manner to the laminar cylinder

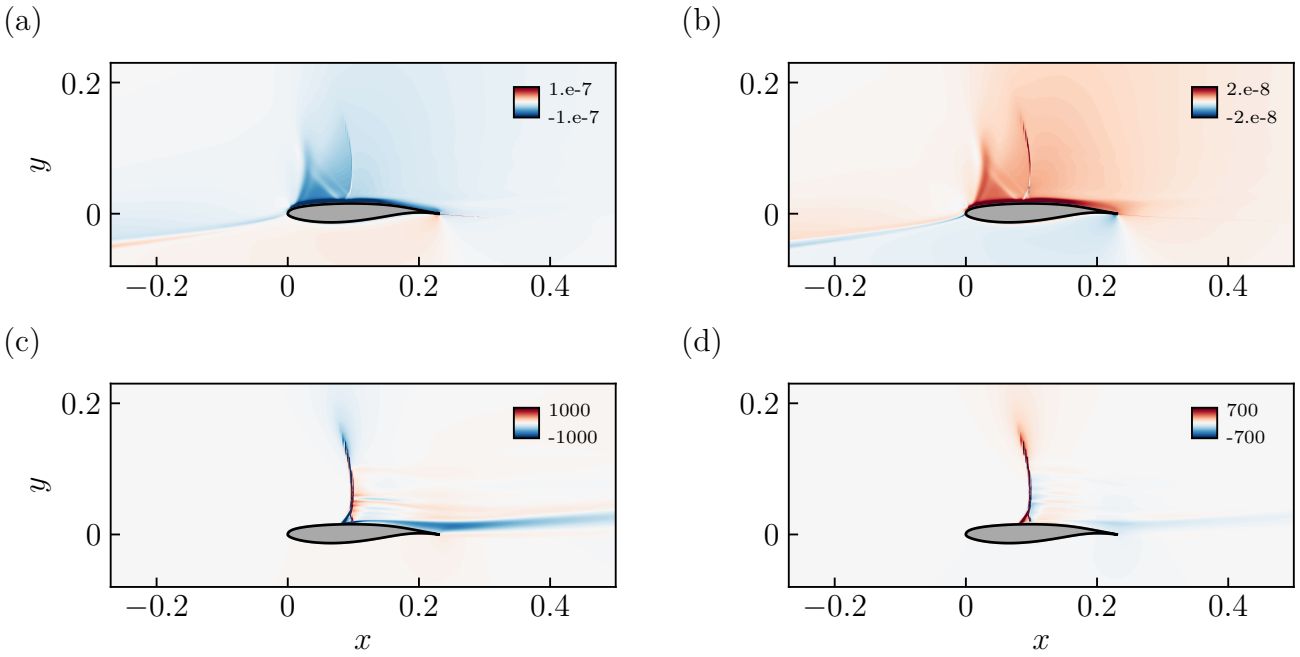


Figure IV.10: Resolvent analysis at $\alpha = 3.75°$ and St = 0.0613. (top) $\Re(\hat{u}_x)$ (a) and $\Re(\hat{\rho})$ (b) for optimal forcing. (bottom) $\Re(\hat{u}_x)$ (a) and $\Re(\hat{\rho})$ (b) for the corresponding optimal response.

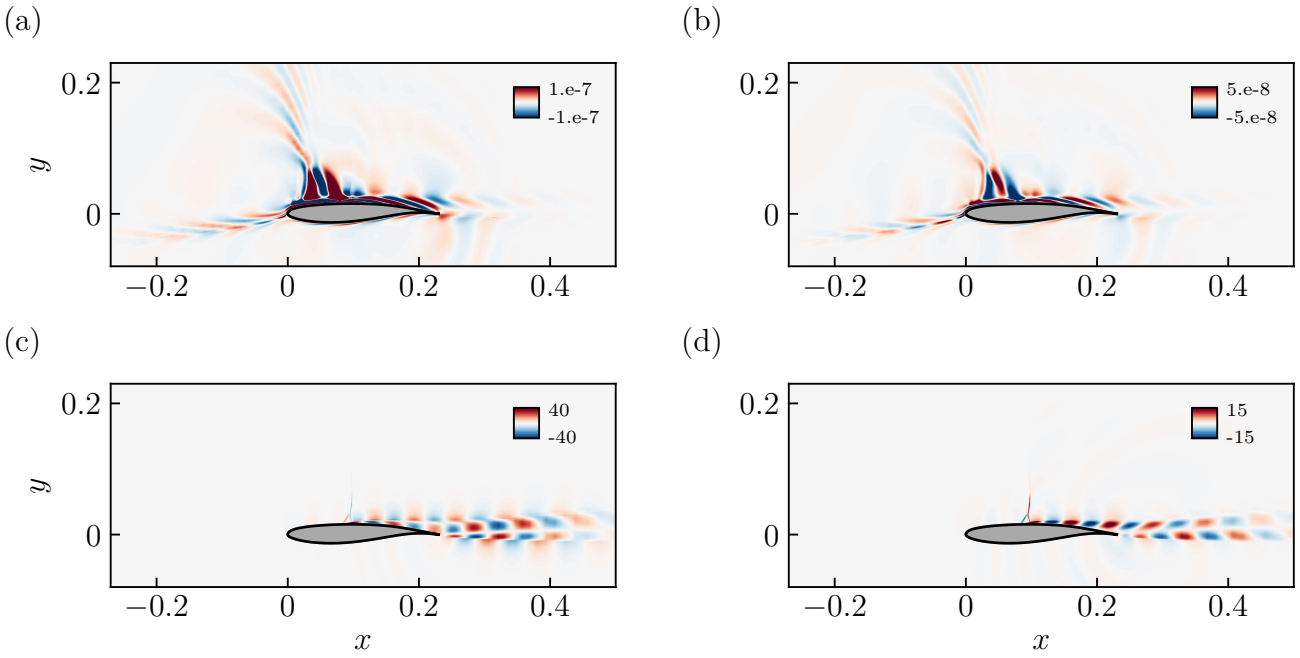


Figure IV.11: Resolvent analysis at $\alpha = 3.75°$ and St = 2.44. (top) $\Re(\hat{u}_x)$ (a) and $\Re(\hat{\rho})$ (b) of optimal forcing. (bottom) $\Re(\hat{u}_x)$ (a) and $\Re(\hat{\rho})$ (b) of the corresponding optimal response.

flow (see fig. III.6.3). This indicates that the sub-critical base flow at α_c can become unstable by amplifying such frequency. A secondary instability is identified at St = 2.44, resulting in a dimensional frequency of 3430 Hz. The corresponding optimal forcing and response are plotted in fig. IV.11. The optimal response is related to a Kelvin-Helmholtz instability. As stated by Sartor in [41], due to the higher frequency, both the optimal response and forcing exhibit small-scale structure patterns. Furthermore, because of the receptivity of the flow at these medium frequencies, the optimal forcing shape could be used to stabilize the unstable buffet phenomenon that occurs at lower frequencies.

We have just demonstrated that the developed KFGSM is fully operational for RANS cases, ranging from modal to non-modal analysis. As for the laminar cylinder case, nonlinear unsteady computations have been performed, and the results are shown below.

IV.2.6 URANS computation

The accuracy of the linear solver in predicting the unsteady buffet phenomenon was tested by conducting nonlinear URANS computations using the Gear method (see III.7). The physical non-dimensional time-step was set at $\Delta t = 2.2 \times 10^{-5}$ to maintain a CFL number at the wall of approximately 10. The calculations were initiated with the base flow perturbed by the unstable direct mode. Fig. IV.12 shows the exponential temporal evolution of the lift coefficient at $\alpha = 4.5°$. The limit cycle is reached around $t = 50$, where self-sustained oscillations are observed. The motion at two consecutive instances is depicted by plotting contours of the density gradient in fig. IV.13. A strong lambda shock is visible on the suction side of the profile, a characteristic of the buffet phenomenon [111]. This instability causes the shock to move from $x/c = 0.25$ at $t = 86$ to $x/c = 0.16$ at $t = 88$.

The temporally amplified signal is analyzed in fig. IV.14a and IV.14b to extract σ^{NL} and $\mathrm{St}^{\mathrm{NL}}$, respectively. Table IV.6 compares the values of σ and St for linear and nonlinear computations for various angles of attack. Small discrepancies are observed for the amplification rates between linear and nonlinear computations, except for $\alpha = 5.5°$, but in this case, we are far from the criticality and this may be improved by lowering the tolerance of the eigensolver. The frequencies calculated with the linear solver are in good agreement with those obtained with URANS computations, demonstrating the capability of the linear RANS solver in predicting instability both onset and frequency. As discussed by Sartor [41], this result suggests that the second-order harmonics generated by the nonlinearities remain weak, even in very unstable regimes. This outcome is not

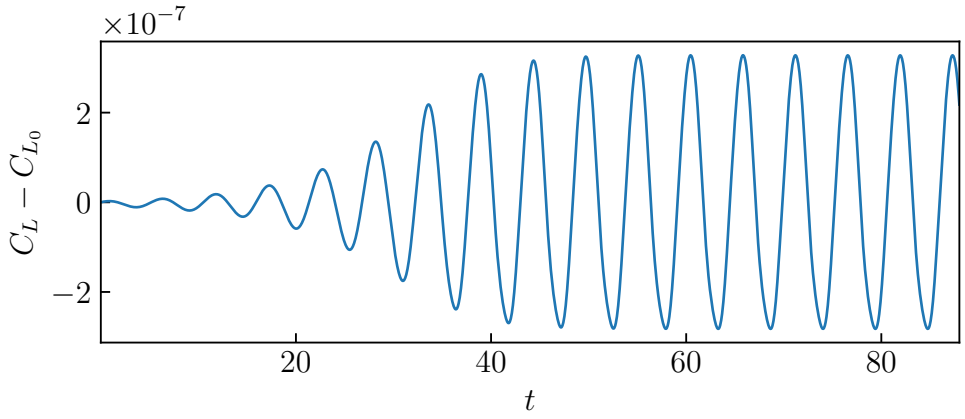


Figure IV.12: Time evolution of the exponentially growing lift coefficient at $\alpha = 4.5°$.

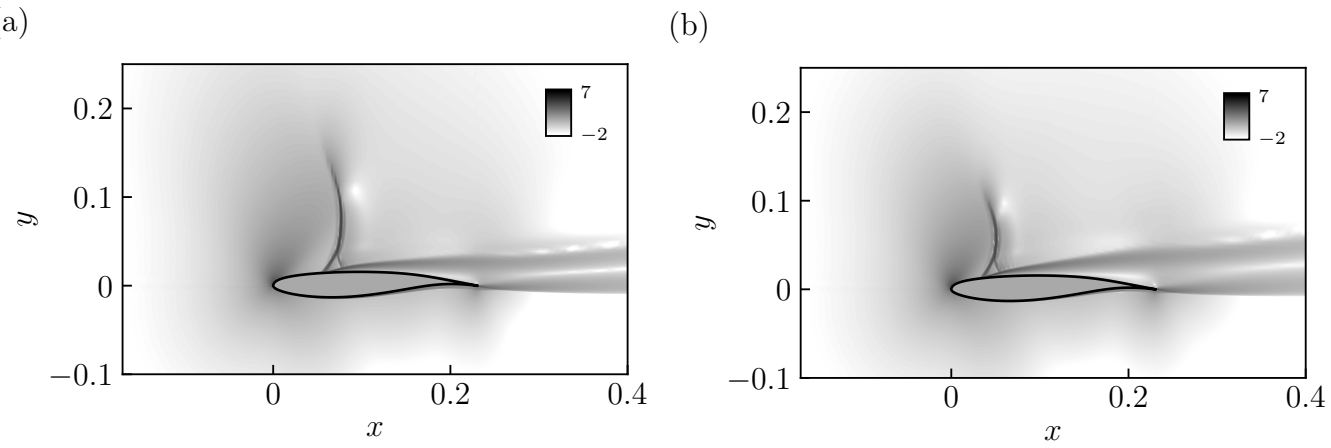


Figure IV.13: Magnitude of the density gradient $\|\nabla\rho\|_2$ map at $t = 86$ (a) and $t = 88$ (b) with logarithmic scale, at $\alpha = 4.5°$.

obvious, as in the case of the laminar cylinder, the prediction of the shedding frequency with the linear solver quickly diverges as the Reynolds number increases (see III.20). It is also worth noting that these results validate our strategy of using the negative SA model instead of the original one to avoid linearization issues.

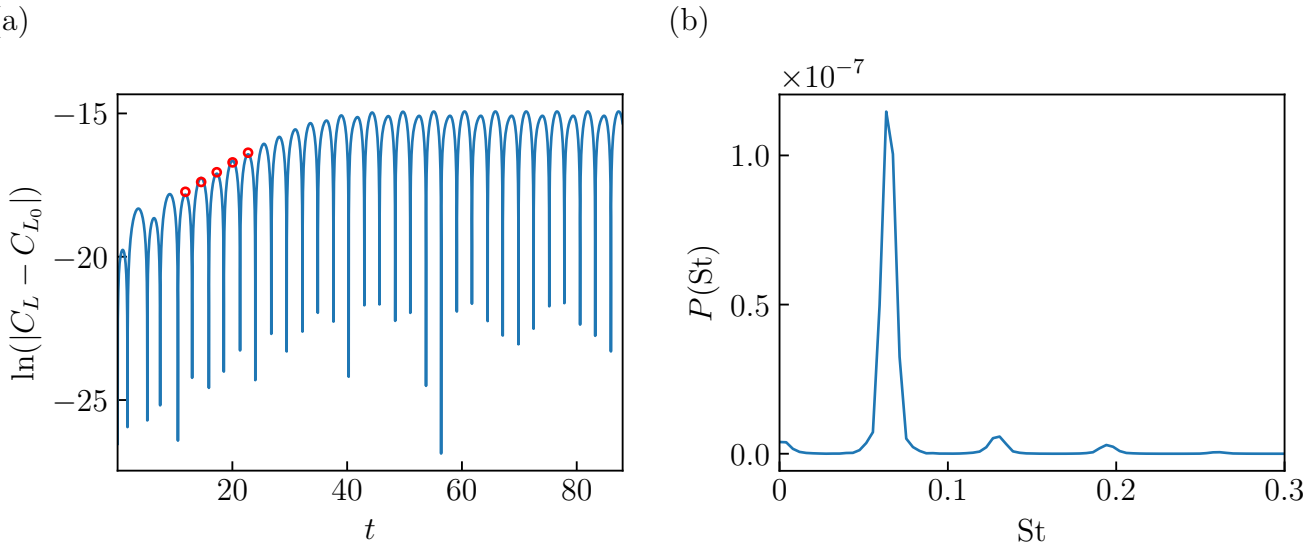


Figure IV.14: (a) Lift coefficient trend over the time where the dotted line represents $\sigma^{\mathrm{NL}} = 0.1248$. (b) Magnitude spectrum of the signal, where a pike at $\mathrm{St}^{\mathrm{NL}} = 0.063667$ is identified, as well as second and third harmonics.

$\boldsymbol{\alpha}$	$(\boldsymbol{\sigma}^{\mathbf{L}}, \mathbf{St}^{\mathbf{L}})$	$(\boldsymbol{\sigma}^{\mathbf{NL}}, \mathbf{St}^{\mathbf{NL}})$	$\boldsymbol{\Delta}\mathbf{St}$	$\boldsymbol{\Delta\sigma}$
4.5°	(0.1243, 0.0638)	(0.1248, 0.0657)	2.9%	4%
5°	(0.1185, 0.0669)	(0.113, 0.0695)	3.7%	4.8%
5.5°	(0.0717, 0.0728)	(0.062, 0.0733)	0.7%	15%

Table IV.6: Comparison between SLEPc and elsA computations for resolvent analysis at $\mathrm{Re} = 40$ and $\mathrm{St} = 0.1165$.

The second part of this chapter focuses on the 3-D subsonic laminar sphere flow. This case was chosen to evaluate the code's ability to solve large 3-D problems and to identify multiple bifurcations in the converged base flows. The bifurcations are investigated using both modal and non-modal analysis, and the results are presented subsequently.

IV.3 Subsonic laminar three-dimensional sphere flow

The unsteadiness in the subsonic laminar sphere flow is a significant area of research because of its similarity to some industrial problems, as noted by Tezuka et al. [26]. Sansica et al. have studied both subsonic and supersonic regimes in [107, 112] and describe the frequencies and eigenfunctions associated with the unstable modes that cause various instabilities. A schematic bifurcation diagram for this flow is shown in fig. IV.15. When the Reynolds number is greater than $\mathrm{Re}_c^1 \simeq 210$, the base flow is axisymmetric (AS) and has a pair of steady streamwise vortices immediately after the sphere body. After Re_c^1, a regular steady bifurcation occurs, breaking the axisymmetry and transforming the base flow into a planar-symmetric (PS) steady flow. A second bifurcation, a supercritical Hopf bifurcation, occurs around $\mathrm{Re}_c^2 = 280$, leading to unsteadiness that takes the form of a large-scale hairpin-like vortex shedding behind the sphere, similar to the laminar cylinder flow (see III.7.2). In this work, we focus on the subsonic case, and both bifurcations are analyzed using direct and adjoint global stability analyses. The receptivity of both types of base flows at various frequencies is also studied through resolvent analysis. Additionally, we demonstrate that the pseudo-transient continuation technique thoroughly used in this thesis (see II.3.2) is robust enough to compute AS base flows that do not have physical existence in regimes above the Hopf bifurcation of PS base flows, revealing another specific Hopf bifurcation.

IV.3.1 Numerical set-up

The computational domain, presented in fig. IV.16, is based on that used by Sansica in [107]. It is an unstructured grid composed of 9.3×10^6 hexahedral cells and constitutes a good test case to assess the capability of the KFGSM to perform global stability analysis on a 3-D configuration of size representative of industrial applications. The physical parameters are presented in table IV.7. Like in the cylinder case (see III.1.1), the conservative equations are non-dimensionalized by T_∞, u_∞, and ρ_∞. A second-order Roe scheme with MUSCL reconstruction and no limiter is used for convective fluxes, while diffusive fluxes are computed using the $5p_{\mathrm{cor}}$ presented in III.3.2. A no-slip velocity condition is imposed at the wall, non-reflective boundary conditions based on

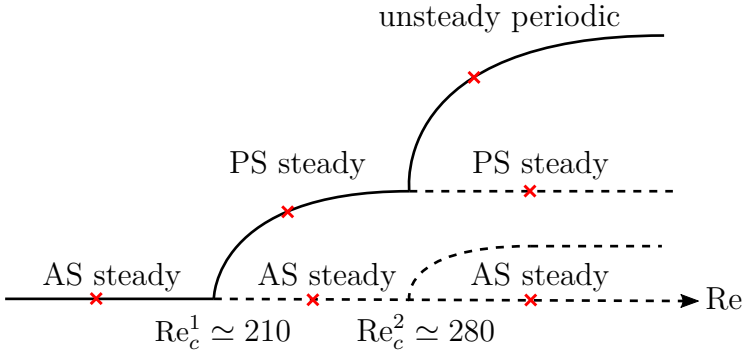


Figure IV.15: Schematic bifurcation diagram of the laminar 3-D sphere flow.

Free-stream Mach number	$\mathrm{M}_\infty = 0.1$
Free-stream temperature	$T_\infty = 300$ K
Free-stream velocity	$u_\infty = 34.7\ \mathrm{m\,s^{-1}}$
Free-stream density	$\rho_\infty = 1.177\ \mathrm{kg\,m^{-3}}$
Reynolds number	$\mathrm{Re} \in [200, 320]$

Table IV.7: Flow parameters of the subsonic laminar flow around sphere.

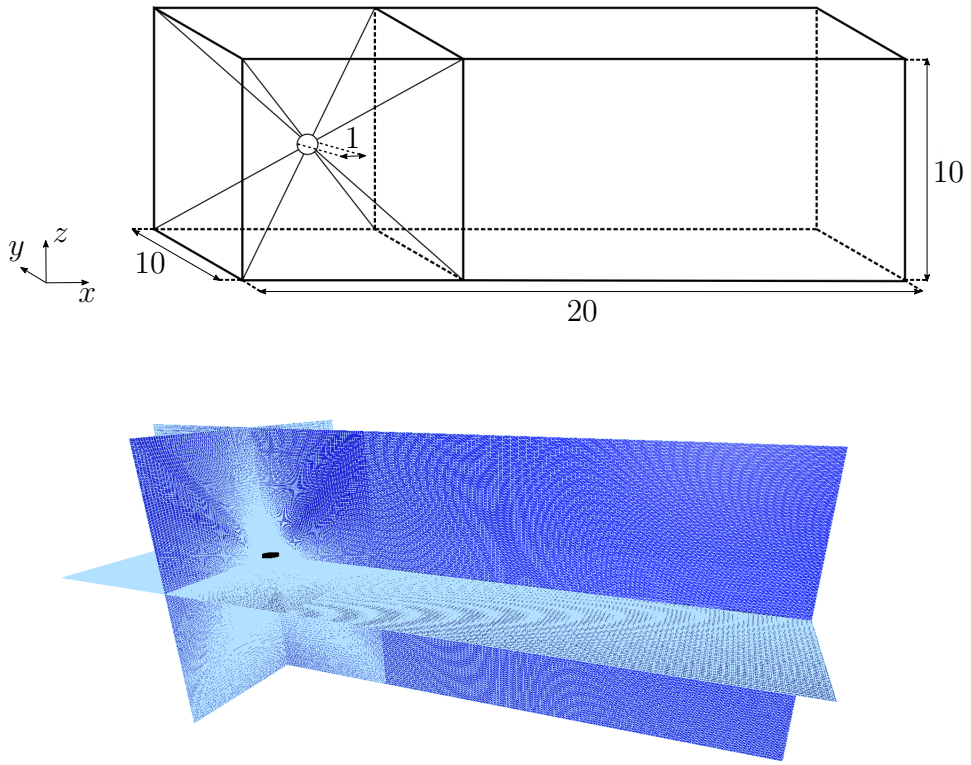


Figure IV.16: Schematic view of computational domain taken from [107] and respresentative vizualisation of the mesh.

characteristics are set at the lateral domain boundaries, a uniform inflow velocity u_∞ is imposed at the $x = 0$ plane, and an outflow condition is imposed at the $x = 20$ plane.

As with the two previous test cases, the first step of this study is to compute a base flow, a steady solution of the conservative equations. The following section provides details on the strategy used to compute both AS and PS base flows.

IV.3.2 Fixed point computation

The computations were carried out on 18 Intel Xeon Broadwell E5-2680v4 nodes, utilizing 500 CPUs. For AS base flows, the calculations were converged using the backward Euler scheme combined with the GMRES method, initialized with a uniform field. The preconditioning used was FGMRES-ILU(3) with one RAS level [51]. The internal and external Krylov bases were composed of 60 vectors, and the internal and external tolerances were set to 10^{-3} and 10^{-4} respectively.

The convergence, demonstrated at Re = 210 in fig. IV.17a, was achieved in 10 iterations with a precision of 4×10^{-16} on the density residual $\|\rho\|_2$.

A different strategy was adopted to compute PS base flows beyond the first regular bifurcation. In fact, initializing the computations with a uniform field always result in an AS base flow, even at Reynolds numbers above the supercritical Hopf bifurcation. To overcome this issue, the uniform field was advanced toward the asymptotic PS base flow using the LU-SGS computation (see III.3.3) over 10^4 iterations, performed on 200 CPUs. The resulting field was used to initialize the implicit backward Euler solution with FGMRES-ILU(3) preconditioning. The tolerance on $\|\rho\|_2$ was set to 10^{-11} and the convergence, shown in fig. IV.17b, was achieved in 41 iterations. Table IV.8 provides comparative figures between both resolution methods. As with the OAT15A case (see IV.2.2), the memory requirement of the FGMRES-ILU(3) method is much higher than that of the LU-SGS method, but the base flow is converged with higher precision, reducing numerical noise in the computed fields.

The contour maps of velocity magnitude for the AS and PS base flows at Reynolds numbers Re = 210 and Re = 280 are shown in fig. IV.18. In agreement with the flow visualizations reported by Magarvey et al. in [113], a single thread is observed within the AS base flow and symmetry across both the x-y and x-z planes is demonstrated in fig. IV.18a and IV.18c. On the other hand, the PS base flow has two threads, and fig. IV.18d and IV.18b highlight the preservation of symmetry across the x-z plane and the loss of symmetry across the x-y plane. It is worth

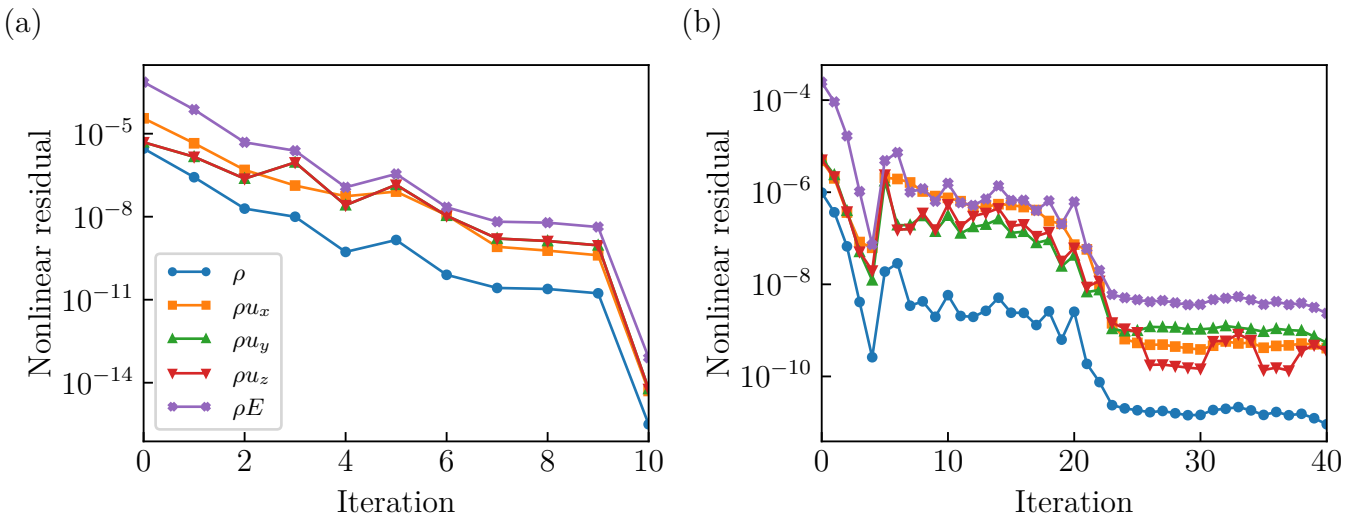


Figure IV.17: Convergence of steady base flow at Re = 210 (a) and Re = 280 (b).

Solver	LU-SGS	GMRES
nonlinear iterations	10^4	41
Final $\|\rho\|_2$	2.4×10^{-8}	9×10^{-12}
Number of CPUs	300	500
Memory/CPU after computation	0.72 GB	3.24 GB
CPU time	300 s	10.65 h

Table IV.8: Indicative quantities for both fixed point computation methods at Re = 280 for the planar-symmetric base flow.

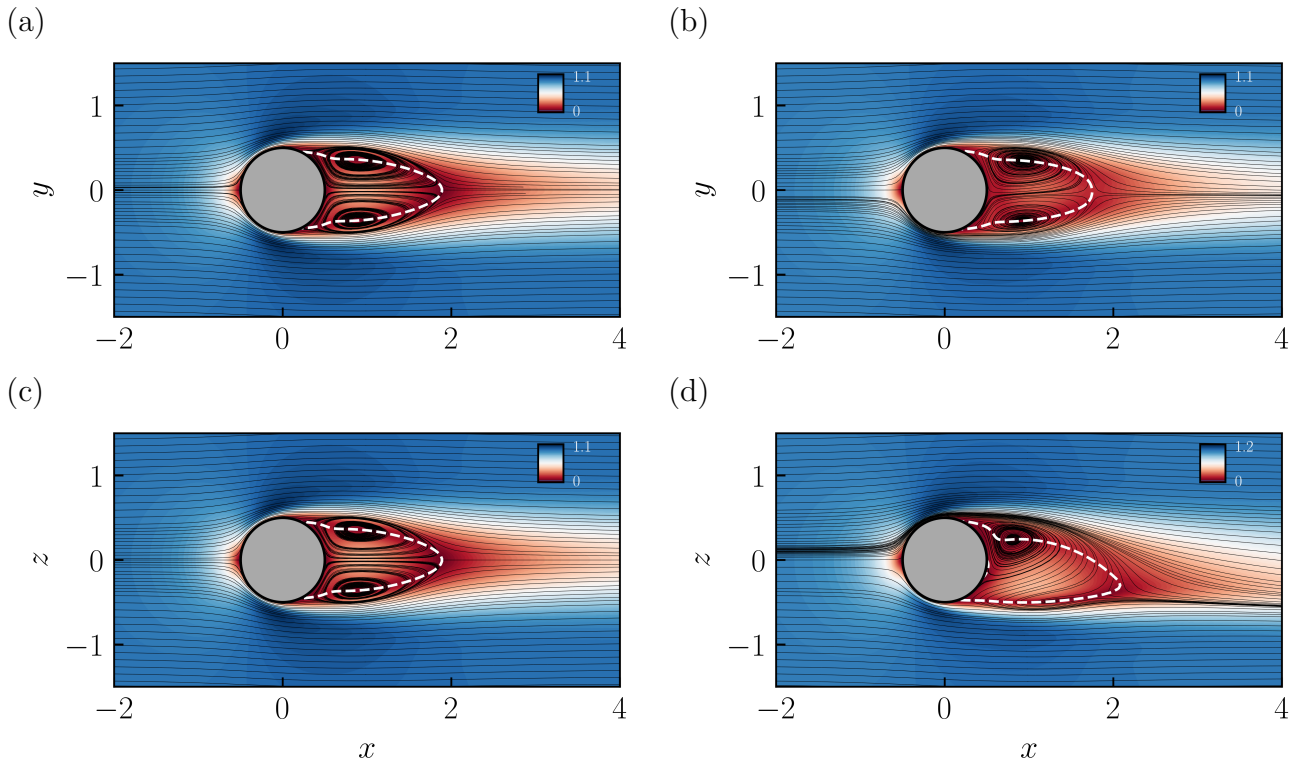


Figure IV.18: Contour map of velocity magnitude for the AS base flow at Re = 210, in the x-y (a) and x-z (c) planes, and the PS base flow at Re = 280 in the x-y (b) and x-z (d) planes. Streamlines are plotted and the dashed-lines represent the recirculation bubble.

mentioning that since the symmetry break may occur with equal probability at each azimuthal position, the PS base flow has been rotated so that its symmetric plane aligns with the x-z plane.

The evolution of L_b with respect to Re is plotted in fig. IV.19 and compared to the results reported by Sansica in [107] under similar conditions. The length of the recirculation bubble L_b for the PS base flow was calculated by taking the average of the lengths of its projections onto the x-y and x-z planes. The lengths of the AS bubbles are in good agreement with Sansica's results,

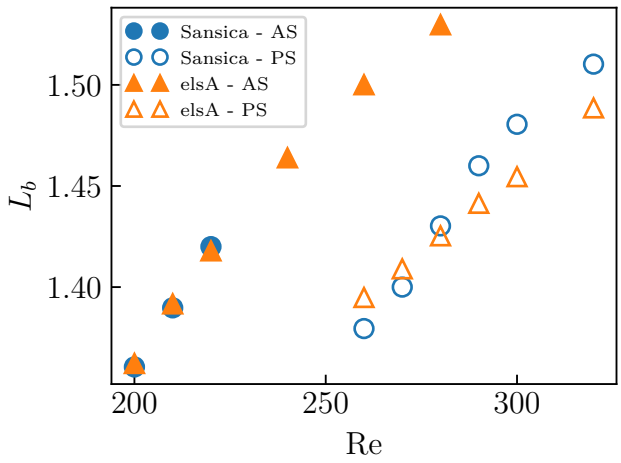


Figure IV.19: Evolution of the recirculation bubble length according to Re for axi and planar-symmetric base flows.

in contrast to the PS ones, which grow slightly slower in our case. This difference can be explained by both the different convergence rates achieved in both computations and the different numerical modelling. Despite these observations, both tendencies seem similar enough to validate our base flow computations.

Due to the distinct characteristics of AS and PS base flows, the rest of this study is organized to analyze the modal and non-modal stability of each type of base flow separately. The following sections present a global stability analysis around each of these distinct base flows.

IV.3.3 Regular bifurcation of axi-symmetric base flow

Direct modal analysis. The selected preconditioner for solving the K-formulated linear systems involved in the global stability analyses performed on the laminar 3-D sphere case was the FGMRES-ILU(3) with one RAS level. The computations were carried out on the same cluster as the fixed point computations, but 43 nodes, consisting of 1200 CPUs, were required for the modal analyses to meet the memory requirements of the preconditioner. Within the GMRES, the internal and external Krylov bases consisted of 100 and 30 vectors, respectively. The internal tolerance was set to 10^{-3} and the external one was set to 10^{-6}. The shift was set to $s = 1$, and the KS basis was composed of 22 vectors, with a relative tolerance for convergence of eigenpairs of 10^{-4}. The least stable eigenpair converged below the tolerance in about 8 KS iterations.

The eigenvalues are shown in fig. IV.20. The onset of symmetry breaking is determined to occur between $\mathrm{Re} = 200$ and $\mathrm{Re} = 210$, which is consistent with previous studies by Sansica [107], Tezuka & Suzuki [26], and Fabre et al. [114]. The least stable direct eigenfunction at $\mathrm{Re} = 210$ is depicted in fig. IV.21, normalized by the L^∞-norm of the perturbation velocity vector. To better highlight the azimuthal nature of the eigenfunctions, a transformation from cartesian coordinates (x, y, z) to cylindrical coordinates (x, r, θ) has been applied, yielding the transformed perturbation velocity components:

$$\begin{pmatrix} u_x \\ u_r \\ u_\theta \end{pmatrix} = \begin{pmatrix} u_x \\ \cos(\theta)u_y + \sin(\theta)u_z \\ -\sin(\theta)u_y + \cos(\theta)u_z \end{pmatrix}. \tag{IV.8}$$

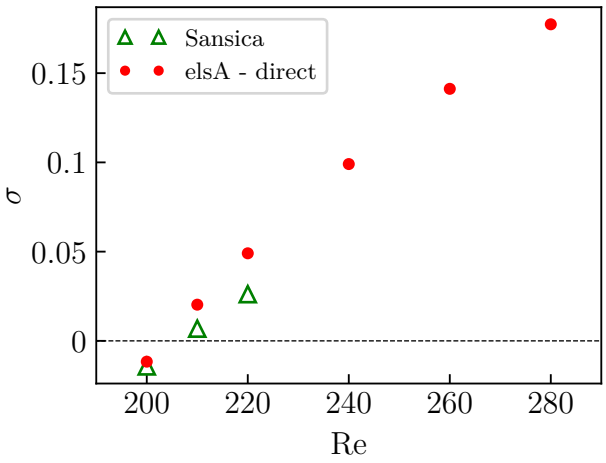


Figure IV.20: Evolution of amplification rate of the least stable mode according to Re for direct global stability analysis around AS base flow in the regular bifurcation. Symmetry breaking occurs at $\mathrm{Re}_c^1 = 210$.

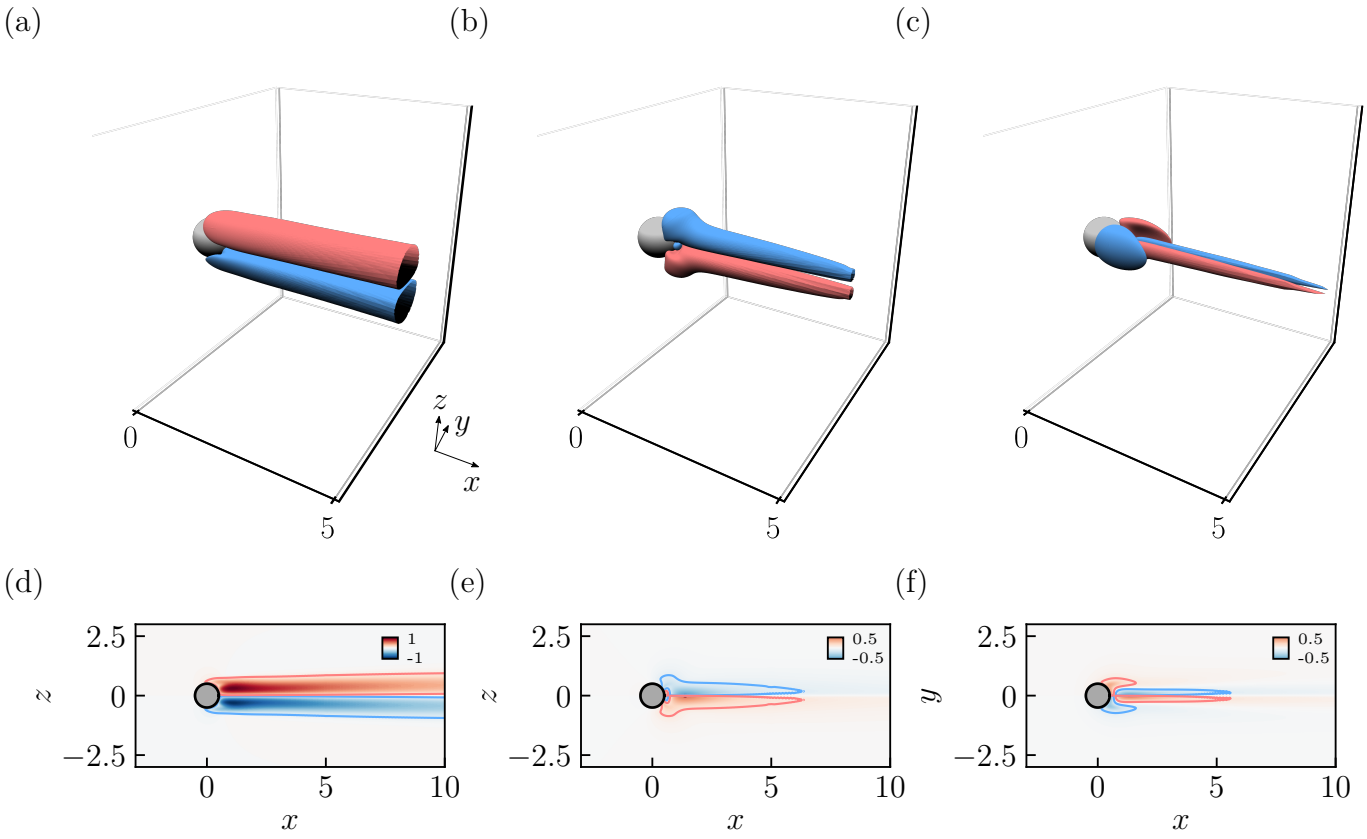


Figure IV.21: 3-D views of the direct unstable mode at Re = 210, with $(\sigma, \mathrm{St}) = (0.02027, 0)$ for AS base flow. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$. (bottom) Corresponding sectional contour plot at (d) $z = 0$, (e) $z = 0$ and (f) $y = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$.

The steady mode's spatial structure is consistent with previous findings by Sansica [107], Fabre [114] and Tezuka & Suzuki [26]. As noted by Fabre, the mode displays reflectional symmetries about longitudinal planes. The dynamic is dominated by the streamwise velocity component, which expands and grows over the domain at $x = 20$ with infinitely large scale structures due to its non-oscillatory character.

Adjoint modal analysis. The computation of adjoint eigenpairs of AS base flows was performed using the same parameters as for direct modal analysis, except for the tolerance on the eigenpairs, which was set to 10^{-3}. The convergence was reached in approximately 5 KS iterations. The resulting least stable adjoint eigenvalues are shown in Fig. IV.20. A good agreement is observed between the direct and adjoint eigenvalues. The duality between the direct and adjoint operators has already been validated in the laminar case using a cylinder, but this result shows that the duality is also valid in three dimensions. The perturbation velocity components of the corresponding unstable eigenfunction in cylindrical coordinates at Re = 210, normalized by the L^∞-norm, are shown in fig. IV.23. Unlike the direct mode, the dynamic of the adjoint mode is dominated by radial and streamwise perturbation velocity components. The main part of this mode is localized immediately after the sphere body and extends up to $x \simeq 2$. It also shares the same symmetry properties as the direct mode. To delay the breaking of symmetry for this AS base flow, it is hence

important to consider the strong receptivity of the direct mode to external forcing in these regions.

The shape of the structural sensitivity calculated using the direct and adjoint unstable steady modes is displayed in fig. IV.24. It is seen to be plane-symmetric across the x-y and x-z planes,

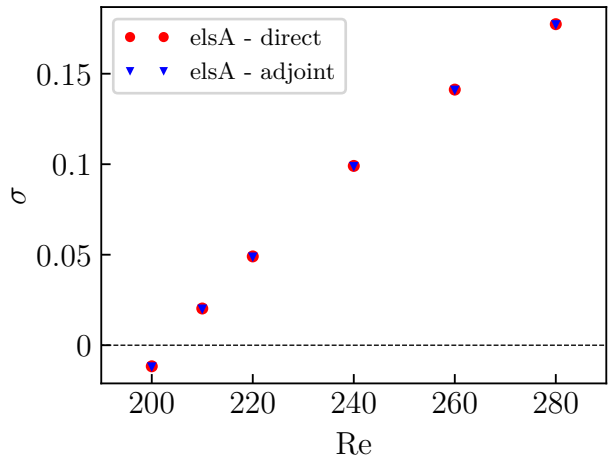


Figure IV.22: Evolution of amplification rate of the least stable mode according to Re for adjoint global stability analysis around AS base flow in the regular bifurcation.

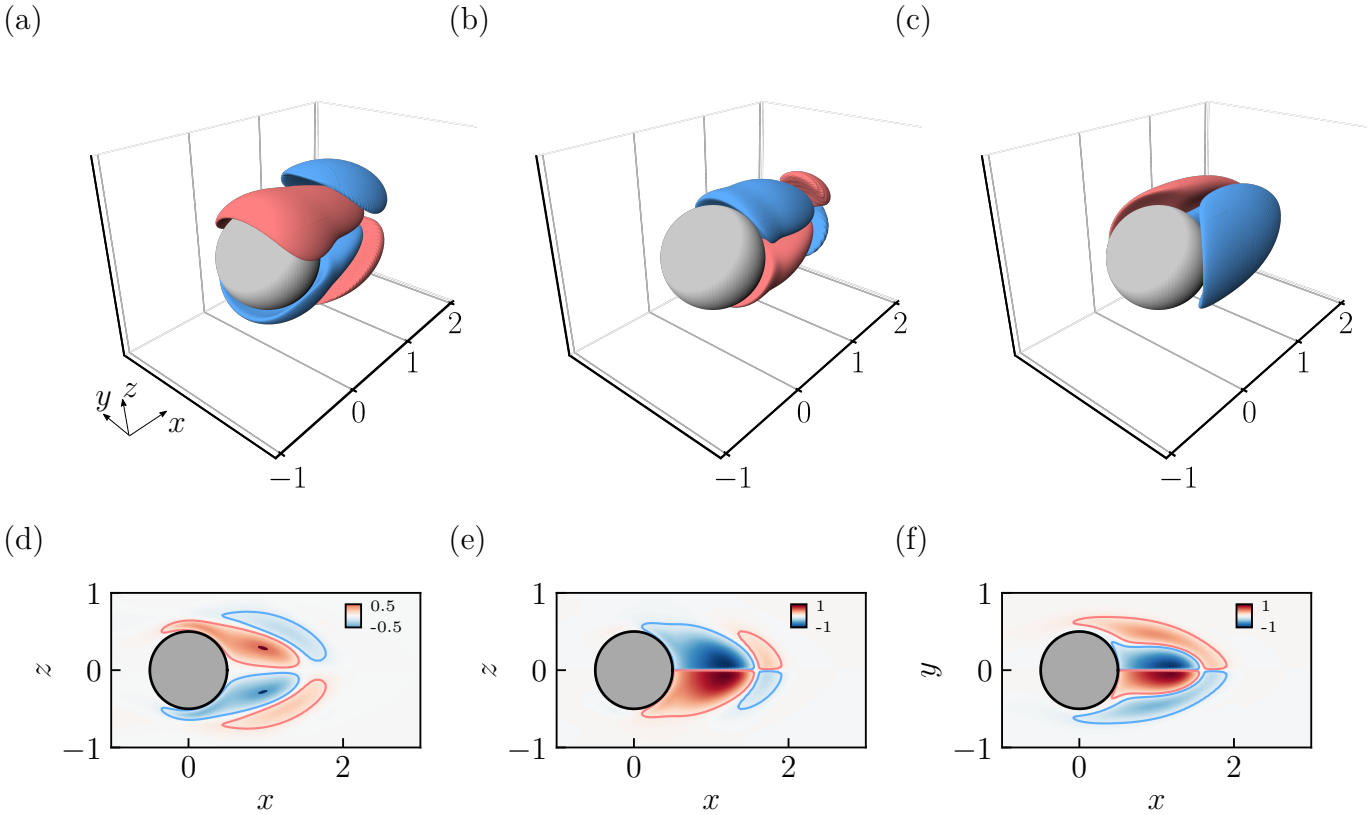


Figure IV.23: 3-D views of the adjoint unstable mode at Re = 210, with $(\sigma, \mathrm{St}) = (0.02028, 0)$ for AS base flow. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$. (bottom) Corresponding sectional contour plot at (d) $z = 0$, (e) $z = 0$ and (f) $y = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$.

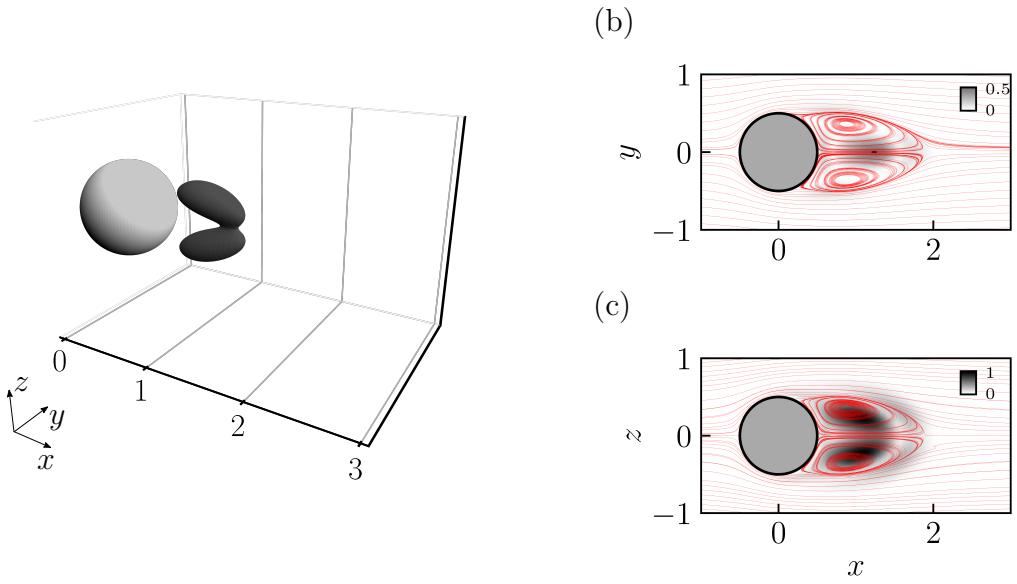


Figure IV.24: (a) Isosurface of structural sensitivity for the AS base flow at Re = 210 for the value $\boldsymbol{S} = 0.5$, and corresponding sectional contour plots at (b) $z = 0$ and (c) $y = 0$.

and it indicates that a symmetry break may occur when one of the two dark lobes visible between $x = 0.5$ and $x = 2$ is perturbed. The most significant values are observed in the x-z plane. The streamlines of the corresponding AS base flow have been added to help better understand the localization of the wavemaker in the beginning of the recirculation bubble. The two dark lobes in the recirculation bubble are therefore the zones directly responsible for the instability. As previously mentioned, the computational mesh must be small enough in this region. This quantity is of great interest in an industrial context as it enables the reduction of the mesh size while still capturing the instability. To the author's knowledge, both the adjoint eigenfunction and structural sensitivity for this first bifurcation have not been shown in the literature.

Having characterized the first bifurcation of the AS base flow, we will focus in the following on a second bifurcation that occurs in this flow. This second bifurcation is a Hopf bifurcation and results in oscillatory modes.

IV.3.4 Hopf bifurcation of axi-symmetric base flow

Above Re = 210, the AS base flow has already undergone a bifurcation towards a plane-symmetric configuration, which, in turn, becomes unsteady through a supercritical Hopf bifurcation around Re = 280 (see fig. IV.3.6 and [107, 31]). As stated in the introduction, an AS base flow computed above Re = 280 is not physically meaningful. However, as the Reynolds number is increased, a pair of complex conjugate oscillatory unstable modes appear, related to a Hopf bifurcation. Thus, two types of instability coexist consisting in one steady mode and two unsteady ones. This situation mimics more complex configurations in which multiple bifurcations can occur simultaneously and demonstrates the robustness of our approach in computing them. Being able to compute each of these modes can help to understand the scenario of transition from steady to unsteady. To this end, direct and adjoint modal oscillating modes are calculated for the cases Re = 240, Re = 260, and Re = 280.

Direct modal analysis. The KS bases used in these computations consisted of 30 vectors and half of the basis was reused from one KS iteration to the next. The shift was set to $s = 0.5 + 0.82i$ in order to target the oscillatory unstable mode, which, based on resolvent analysis (to be presented in IV.3.5), should have a frequency close to St = 0.13. The errors associated to the unstable oscillatory eigenpairs drop below 10^{-4} in 4 KS iterations. The computations yield both steady and unsteady modes. Fig. IV.25 shows the spectra of the least stable eigenvalues calculated for the various Reynolds numbers considered. It turns out that the oscillatory mode becomes unstable

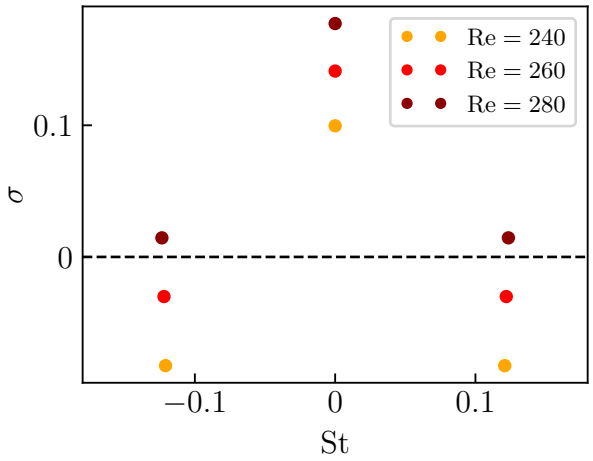


Figure IV.25: Spectra consisting of the least stable direct modes at various Reynolds number for AS base flows. The oscillatory mode destabilizes between Re = 260 and Re = 280.

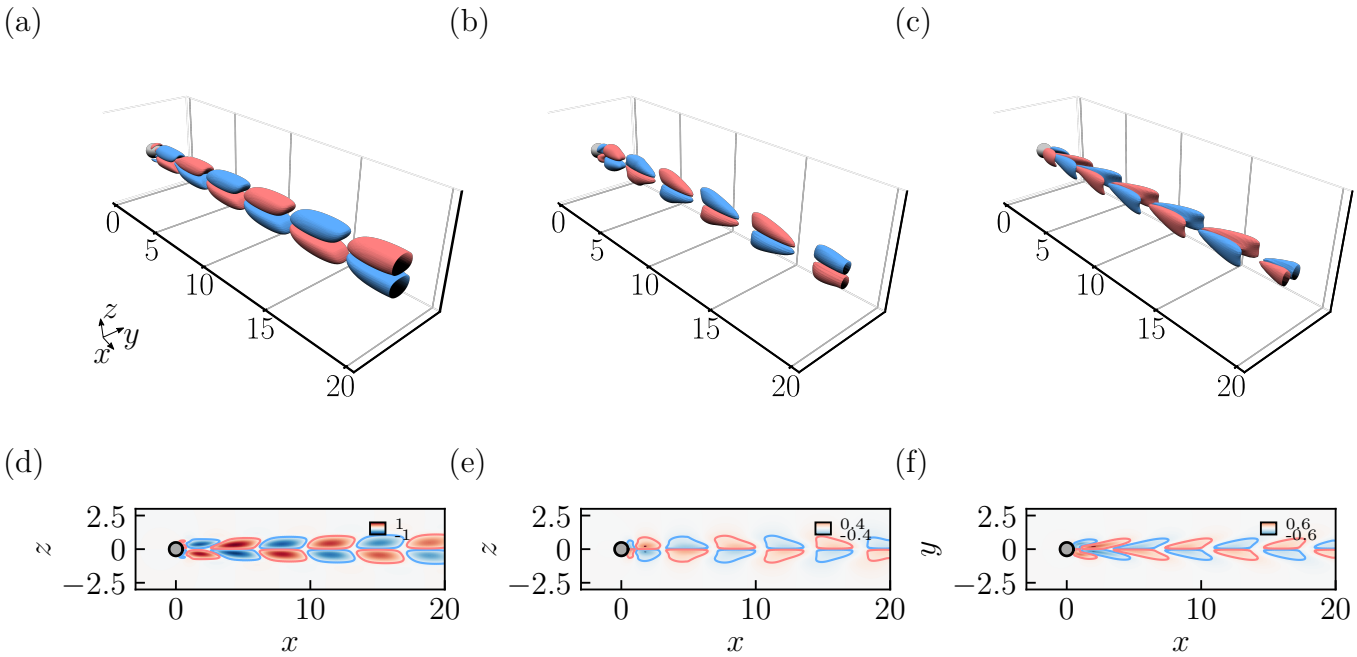


Figure IV.26: 3-D views of oscillatory unstable direct mode at Re = 280, with $(\sigma, \mathrm{St}) = (0.0145, 0.1235)$ for AS base flow. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$. (bottom) Corresponding sectional contour plot at (d) $y = 0$, (e) $y = 0$ and (f) $z = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.1$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$.

between Re = 260 and Re = 280. This is an interesting result since the Hopf bifurcation of the PS base flow also occurs in this range of Reynolds numbers (see fig. IV.3.6). To the author's knowledge, this result has never been reported in the literature.

Isosurfaces and sectional contours of the velocity components, normalized by the L^∞ norm, are shown in fig. IV.26 for Re = 280. Since each mode computed using linear stability analysis is defined by one phase and one amplitude, and due to the axisymmetry of the base flow, we chose the phase so that the symmetry plane aligns with the x-z plane. The mode consists of small-scale structures that are advected downstream from the sphere body until $x = 20$. A symmetry pattern across the longitudinal direction is also observed. It is worth noting that the mode has a similar spatial structure to the directly unstable mode computed by Sansica in [107] and Citro in [31] for the planar-symmetric base flow. The steady mode is not displayed here, but it has a shape that is very similar to that calculated at Re = 210 and shown in fig. IV.21.

Adjoint modal analysis. For these adjoint computations, the same parameters were used as for the direct ones. As previously demonstrated for the direct case, the spectra corresponding to the specified Reynolds number are shown in Fig. IV.27. The adjoint eigenvalues match the direct eigenvalues exactly, further confirming the duality between the direct and adjoint eigensolvers.

Isosurfaces and sectional contours of the perturbation velocity components, normalized by the L^∞-norm, are shown in fig. IV.28. The most energetic part of the mode is located in the recirculation bubble downstream of the spherical body. This eigenfunction shares similarities with the adjoint mode calculated by Citro in [31] for the Hopf bifurcation of PS base flows. Similar to the direct mode, a symmetry pattern along the longitudinal direction is observed. However, unlike the direct mode, this adjoint mode is convected upstream in the flow domain, which is analogous to the results from the laminar cylinder case (see III.6.2).

The structural sensitivity calculated from the direct and adjoint oscillatory modes have been computed and the resulting spatial structure is shown in Fig. IV.29. Similar to the structural sensitivity calculated at Re = 210 for the steady mode (see fig. IV.24), this one is symmetric across both the x-y and x-z planes. The wavemakers are also located in the recirculation region, but at the periphery of the counterclockwise rotating vortices, which differ slightly from these of the regular bifurcation. Additionally, the wavemaker is found to lie in the x-z plane, which agrees

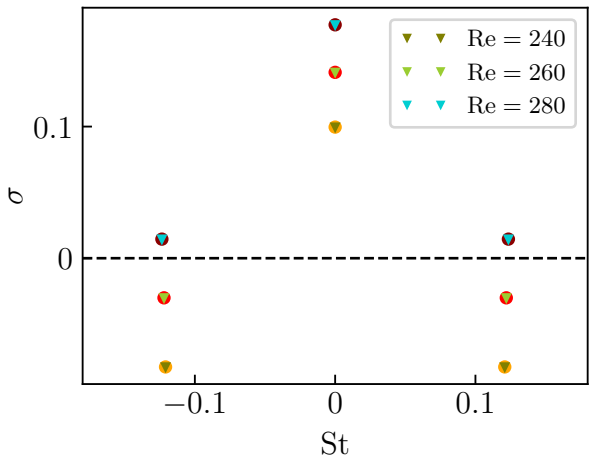


Figure IV.27: Spectra consisting of the least stable adjoint eigenvalues at various Reynolds number for AS base flows. The corresponding direct eigenvalues are also displayed with circles.

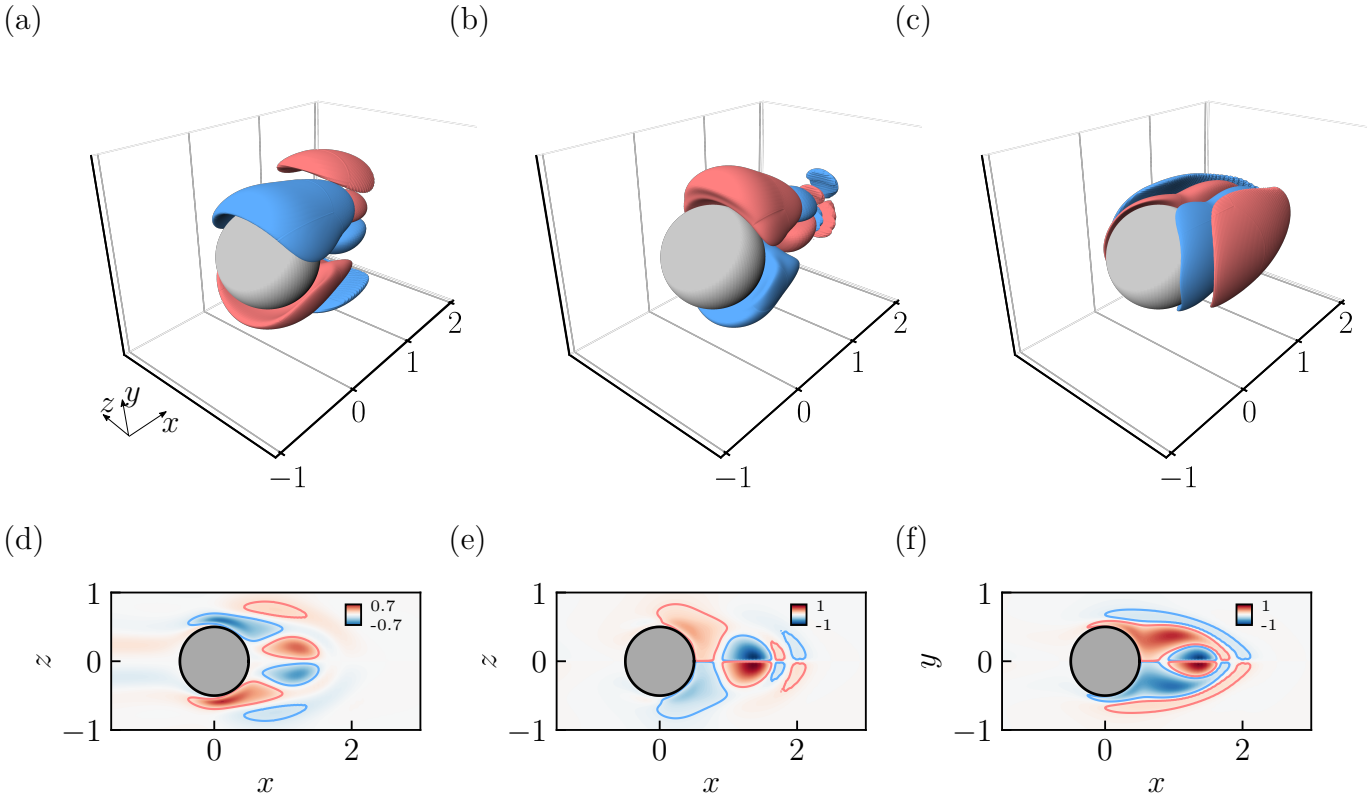


Figure IV.28: 3-D views of oscillatory unstable direct mode at Re = 280, with $(\sigma, \mathrm{St}) = (0.0145, 0.1235)$ for AS base flow. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 0.15$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$. (bottom) Corresponding sectional contour plot at (d) $y = 0$, (e) $y = 0$ and (f) $z = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.15$, $u_r = \pm 0.1$, $u_\theta \pm 0.1$.

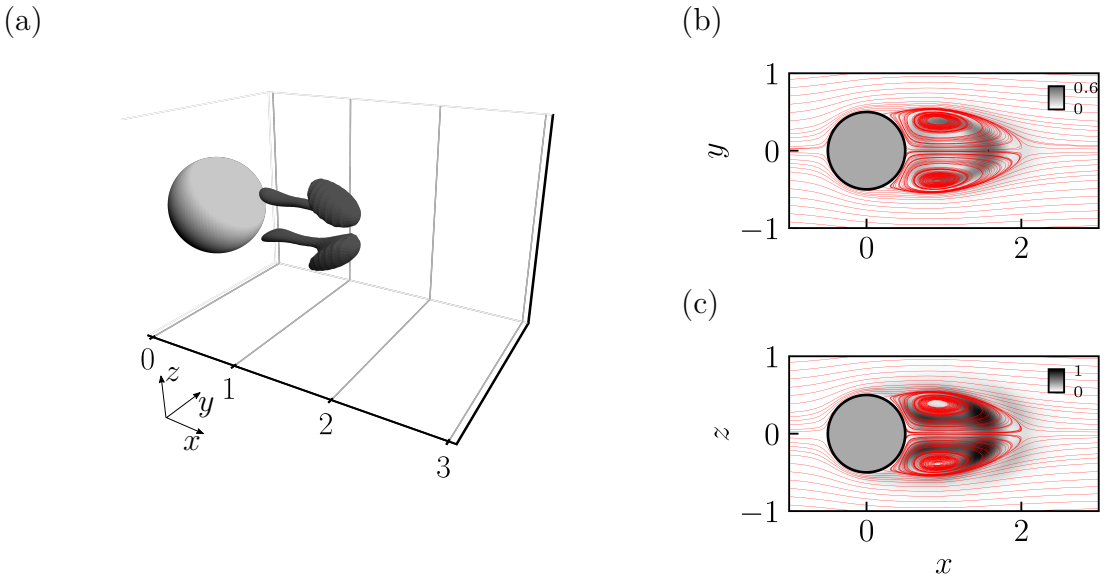


Figure IV.29: (a) Isosurface of structural sensitivity for the AS base flow at Re = 280 for the value $\boldsymbol{S} = 0.65$, and corresponding sectional contour plots at (b) $z = 0$ and (c) $y = 0$.

with the spatial distributions of both the direct and adjoint modes that are dominated by the streamwise velocity, which is anti-symmetric across the x-y plane, but symmetric across the x-z plane. Once again, to the best of the author's knowledge, these results have not been previously shown in the literature.

To conclude the global stability analysis of the AS base flows, a resolvent analysis is performed and described in the following on a subcritical AS base flow. As demonstrated below, even far from the Hopf bifurcation, a peak in the energetic gain curve is observed, indicating a strong receptivity of the flow at the frequency corresponding to the oscillatory axisymmetric mode.

IV.3.5 Resolvent analysis of axi-symmetric base flow

This study focuses on the subcritical AS base flow at Reynolds number 200. Like the previous two test cases in this work, the mass matrices used to solve the eigenvalue problem are selected as $\mathcal{Q}_1 = \mathcal{Q}_{\text{Chu}}$ and $\mathcal{Q}_2 = \mathcal{Q}_\Omega$. As previously stated, the resolvent analysis requires the computation of two preconditioners for the application of $\mathcal{R}^{\text{K}}\mathcal{Q}_\Omega^{\text{K}\,-1}\mathcal{R}^H\mathcal{Q}_{\text{Chu}}^{\text{K}}$. The number of CPUs used for direct and adjoint computation is not sufficient to meet the memory requirements. Thus, there are two possible strategies: either we store only one preconditioner at a time, meaning that for each inversion, one preconditioner is formed and the old one is erased, resulting in lower memory requirements but more CPU time; or we can double the number of processors to increase the total available memory to meet the memory requirements for storing two preconditioners. Given the imposed usage of ONERA's clusters, we chose the second strategy. This results in computations run on 2800 CPUs, using FGMRES-ILU(3) preconditioning with one RAS level. Inversion of the adjoint operator is stiff in this case, so both Krylov bases have been composed of 100 vectors to ensure inversions fall below the tolerance of 10^{-6}. The internal tolerance was set to 5×10^{-2}. Based on the experience of the OAT15A case (see IV.2.5), the KS basis is composed of only 9 vectors, resulting in 18 inversions at each KS iteration. These parameters are sufficient to converge the most energetic gain with a relative precision of 5×10^{-4}, which is usually well separated from the others. The total memory used during the computations is about 12 TB, and the CPU time is about 9 hours.

The optimal gain for various St is shown in fig. IV.30. There is no peak near zero-frequency, unlike what was expected based on previous modal analysis results. This might be improved by

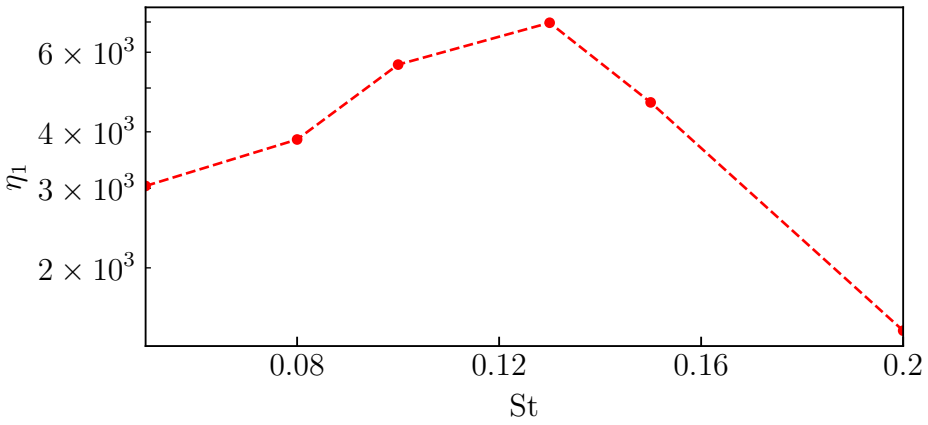


Figure IV.30: Evolution of optimal gain for the resolvent analysis of the sub-critical AS base flow at Re = 200. A peak is identified at St = 0.13.

conducting resolvent analysis closer to St = 0, but this would require solving ill-conditioned linear systems that may not be possible with current tools. Tests were conducted at St = 10^{-4}, but the GMRES method did not converge. However, a peak is seen near St = 0.13, indicating strong flow receptivity at this frequency, which corresponds to that of the previously calculated unstable oscillatory modes. This means that this mode can be expressed as a convective instability by exciting this frequency, even in a sub-critical regime where the base flow is still AS. Fig. IV.31 and IV.32 show views of the normalized velocity components of the optimal forcing and response at St = 0.13, respectively. Both forcing and response have a longitudinal symmetry pattern and are dominated by streamwise velocity. As expected, they are similar to the adjoint and direct modes shown in fig. IV.28 and fig. IV.26. These results demonstrate the effectiveness of resolvent analysis in predicting the most amplified frequencies in a flow, even far from the Hopf bifurcation thresholds.

This analysis terminates the study of the bifurcation scenario for the AS base flow. We can now focus on the Hopf one that occurs on PS base flows, to draw an exhaustive picture of the bifurcation diagram of the laminar subsonic flow around the sphere.

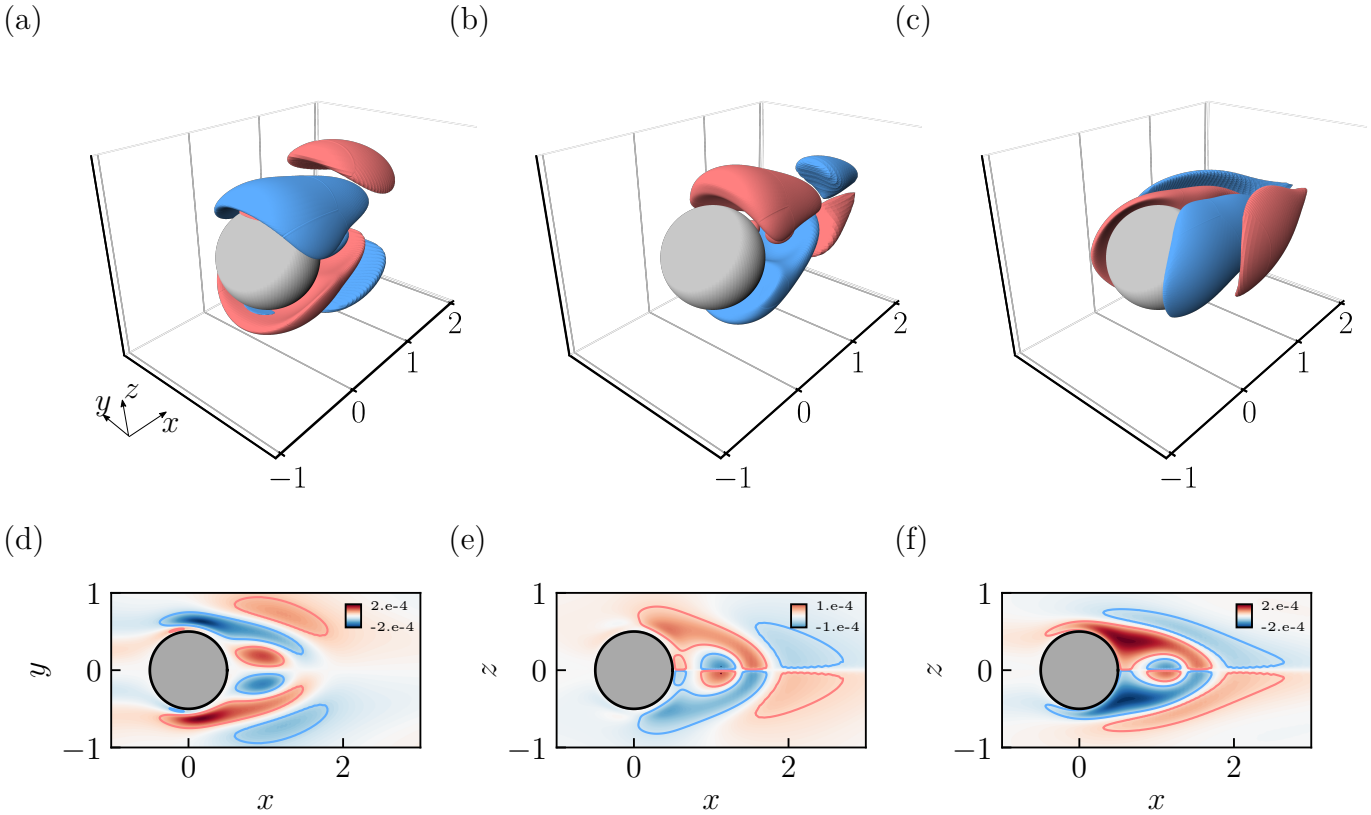


Figure IV.31: 3-D views of optimal forcing at Re = 200 calculated at St = 0.13. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 5 \times 10^{-5}$, $u_r = \pm 3.5 \times 10^{-5}$, $u_\theta \pm 3.5 \times 10^{-5}$. (bottom) Corresponding sectional contour plot at (d) $y = 0$, (e) $y = 0$ and (f) $z = 0$, with solid-lines indicating iso-contours at $u_x = \pm 5 \times 10^{-5}$, $u_r = \pm 3.5 \times 10^{-5}$, $u_\theta \pm 3.5 \times 10^{-5}$.

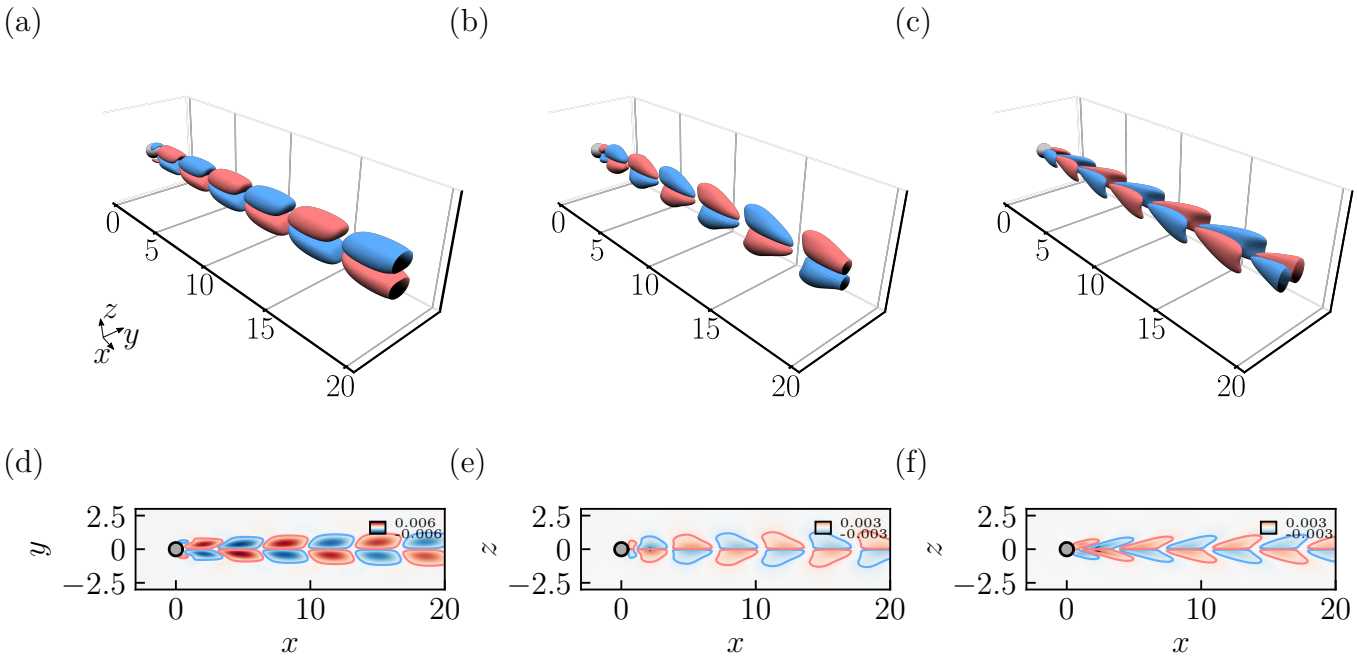


Figure IV.32: 3-D views of optimal response at Re = 200 calculated at St = 0.13. (top) Isosurfaces of streamwise (a), radial (b) and azimuthal (c) perturbation velocities for the values $u_x = \pm 6 \times 10^{-4}$, $u_r = \pm 6 \times 10^{-4}$, $u_\theta \pm 6 \times 10^{-4}$. (bottom) Corresponding sectional contour plot at (d) $y = 0$, (e) $y = 0$ and (f) $z = 0$, with solid-lines indicating iso-contours at $u_x = \pm 6 \times 10^{-4}$, $u_r = \pm 6 \times 10^{-4}$, $u_\theta \pm 6 \times 10^{-4}$.

IV.3.6 Hopf bifurcation of planar-symmetric base flow

In the following, the KFGSM is used to predict the onset of the Hopf bifurcation on the PS base flow presented in IV.3.2. This bifurcation has been well described in the literature, and the results of the direct and adjoint global stability analyses are compared to those obtained by Sansica in [107] and Citro in [31].

Direct modal analysis. The inversion parameters used in the KFGSM are the same as those used in direct and adjoint computations around the AS base flows. The KS basis consisted of 22 vectors, and the shift was set at $s = 0.5 + 0.82i$ to target oscillatory unstable modes. For each Reynolds number, the relative error on the least stable eigenpairs decreased below 10^{-4} after approximately 8 KS iterations.

Fig. IV.33a and IV.33b respectively show the evolution of the growth rate, and the Strouhal number with respect to the Reynolds number. The values of σ obtained are in good agreement with those calculated by Sansica in [107]. However, some discrepancies are observed in the evolution of St between the elsA results and those of Sansica and Citro. This may be attributed to different numerical schemes and levels of convergence. Nevertheless, the overall trend is similar and validates the curve obtained in elsA, which compares well with the results obtained by both Sansica and Citro [31]. The instability threshold is found to be Re = 280, which is consistent with the literature [26, 107]. These results obtained from the linear solver are further validated by nonlinear unsteady computations presented in IV.3.8.

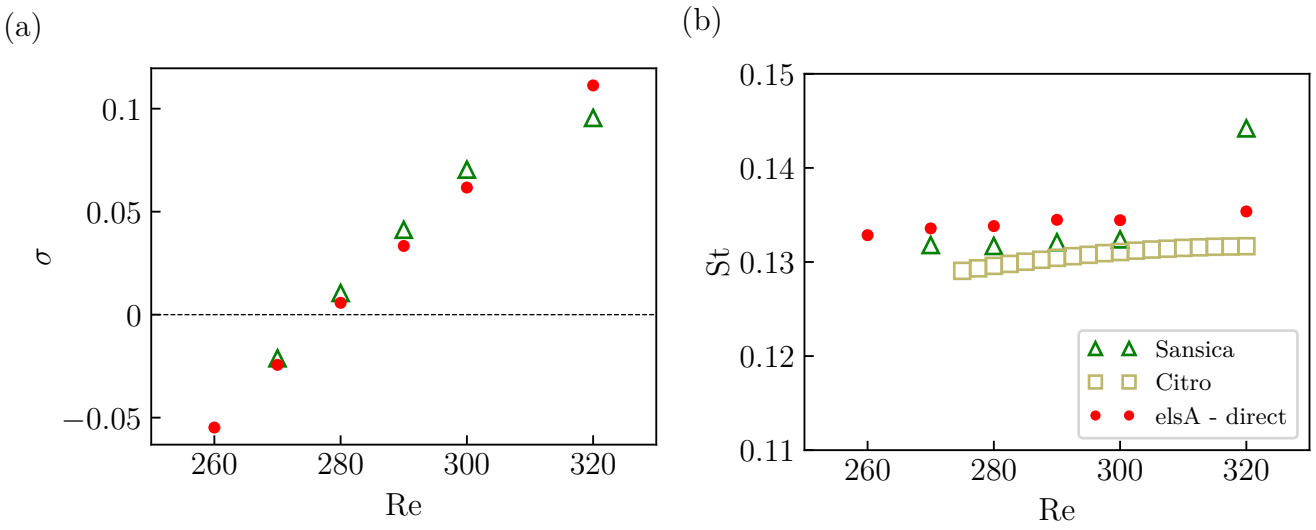


Figure IV.33: (a) Growth rate and (b) Strouhal number evolutions for direct modal analysis of PS base flow. Critical Reynolds number is $\mathrm{Re}_c^2 = 280$.

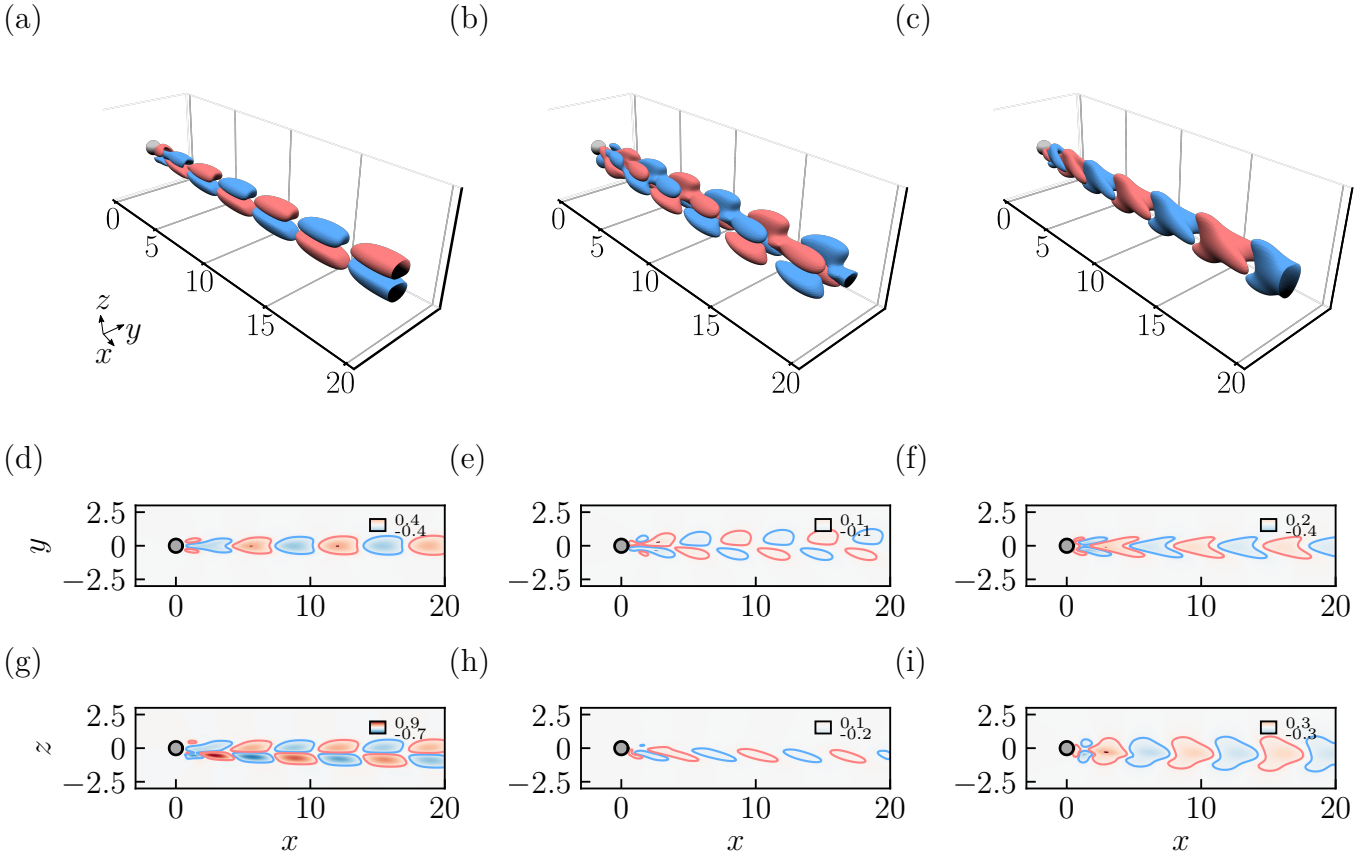


Figure IV.34: 3-D views of oscillatory unstable direct mode at Re = 280 for PS base flow. (top) Isosurfaces of streamwise (a), cross-stream (b) and transverse (c) perturbation velocities for the values $u_x = \pm 0.08$, $u_y = \pm 0.03$, $u_z \pm 0.05$. (bottom) Corresponding sectional contour plot at (d) $z = 0$, (e) $z = 0$, (f) $z = 0$, (g) $y = 0$, (h) $y = 0$ and (i) $y = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.08$, $u_y = \pm 0.03$, $u_z \pm 0.05$.

Adjoint modal analysis. Adjoint computations are carried out using the same KS parameters as for direct computations. The errors for the least stable mode drop below the tolerance set at

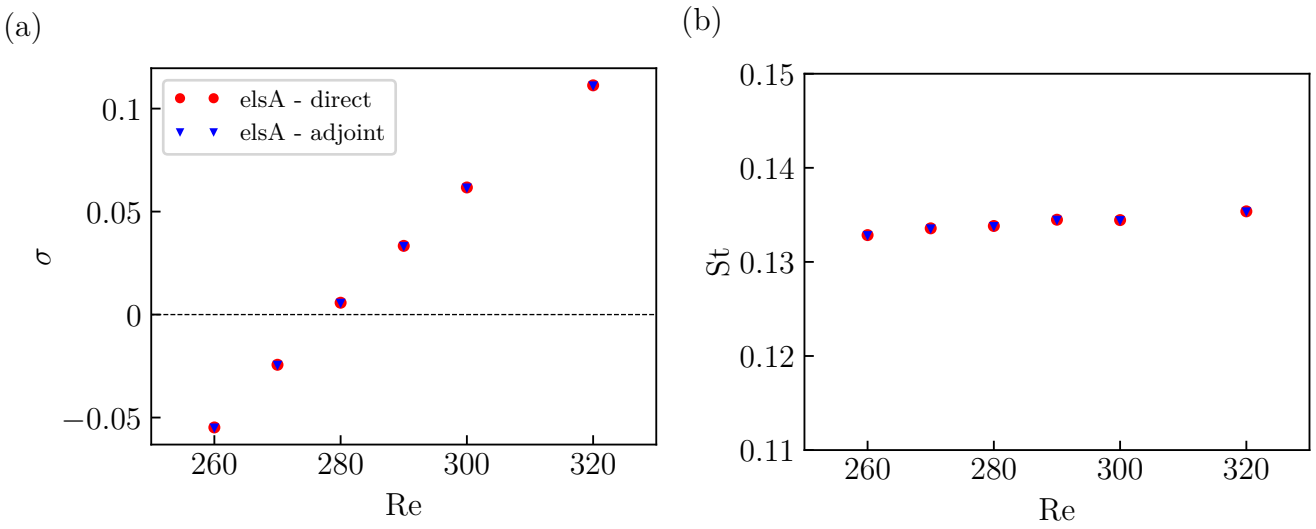


Figure IV.35: (a) Growth rate and (b) Strouhal number evolutions for adjoint modal analysis of PS base flow. Critical Reynolds number is $\mathrm{Re}_c^2 = 280$.

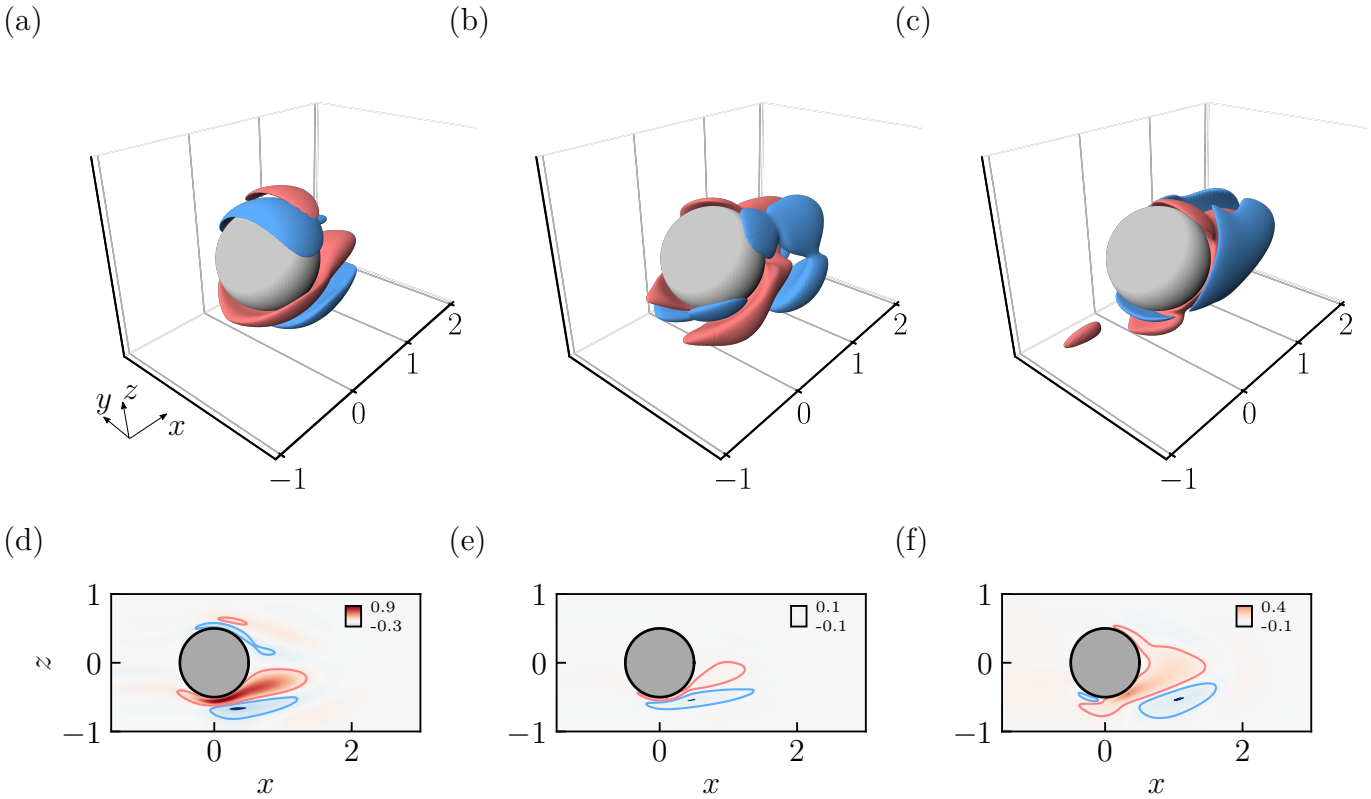


Figure IV.36: 3-D views of oscillatory unstable adjoint mode at Re = 280 for PS base flow. (top) Isosurfaces of streamwise (a), cross-stream (b) and transverse (c) perturbation velocities for the values $u_x = \pm 0.1$, $u_y = \pm 0.025$, $u_z \pm 0.05$. (bottom) Corresponding sectional contour plot at (d) $y = 0$, (e) $y = 0$ and (f) $y = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.1$, $u_y = \pm 0.025$, $u_z \pm 0.05$.

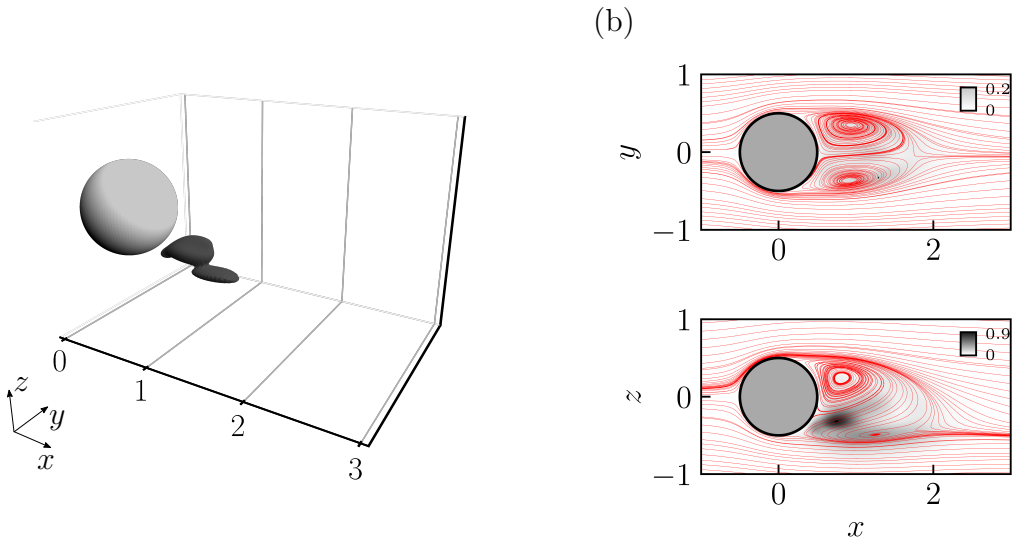


Figure IV.37: (a) Isosurface of structural sensitivity for the PS base flow at Re = 280 for the value $\boldsymbol{S} = 0.4$, and corresponding sectional contour plots at (b) $z = 0$ and (c) $y = 0$.

10^{-3} after approximately 6 KS iterations. The resulting eigenvalues for each Reynolds number considered are shown in fig. IV.35. Once again, the duality between the direct and adjoint operators is confirmed.

The adjoint mode at Re = 280 is shown in fig. IV.36, where the perturbation velocities have been normalized with the L^∞-norm. The first observation is that the spatial structure is in good agreement with that reported by Citro in [31]. The adjoint mode is advected upstream of the domain, in contrast to the direct mode, which is consistent with what was mentioned earlier. The eigenfunction has a similar reflectional symmetry pattern across the x-z plane as the direct mode. Secondly, as with the direct mode, it shares similarities with the adjoint mode of the Hopf bifurcation of the AS base flow.

The structural sensitivity is computed from both the direct and adjoint unstable modes at Re = 280, and the resulting isosurfaces and sectional contours are presented in fig. IV.37. The shape is consistent with the result obtained by Citro [31]. Unlike the structural sensitivities calculated for the AS base flow, only a symmetry across the x-z plane is observed. The wavemaker is located immediately downstream of the spherical body, at the bottom of the recirculation vortex. It differs from the wavemakers identified in the AS case.

As with the AS base flow, this study can be concluded by analyzing the receptivity of the flow to external forcings. To do so, a resolvent analysis is performed on a sub-critical PS base flow, and the results are discussed below.

IV.3.7 Resolvent analysis of planar-symmetric base flow

As previously identified with modal analysis and further confirmed with unsteady computations, the PS base flow evolves toward an unsteady solution above Re = 280. The mode responsible for this unsteadiness has been calculated, as well as the precise region where the instability is triggered, thanks to the structural sensitivity. The natural extension of this work is to conduct a resolvent analysis.

The analysis is conducted on a stable PS base flow computed for Re = 260. The mass matrices defining the resolvent operator are, again, $\mathcal{Q}_1 = \mathcal{Q}_{\text{Chu}}$ and $\mathcal{Q}_2 = \mathcal{Q}_\Omega$. Both the KS parameters and

the number of CPUs used are the same as those used for the resolvent analysis of the AS base flow (see IV.3.5). Similar to the latter, the frequencies where the linear systems were most difficult to solve were those close to St = 0. For instance, for the case with St = 0.025, an average of 95 external GMRES iterations were required during inversions of the adjoint part of the resolvent operator to reduce the error below 10^{-6}.

The resulting gain curve is presented in Fig. IV.38. It exhibits a peak at St $\simeq$ 0.13. The

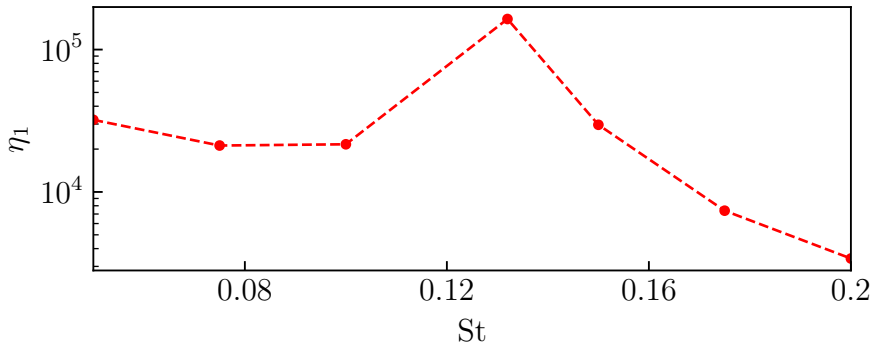


Figure IV.38: Evolution of optimal gain for the resolvent analysis of the sub-critical PS base flow at Re = 260. A peak is identified at St = 0.13.

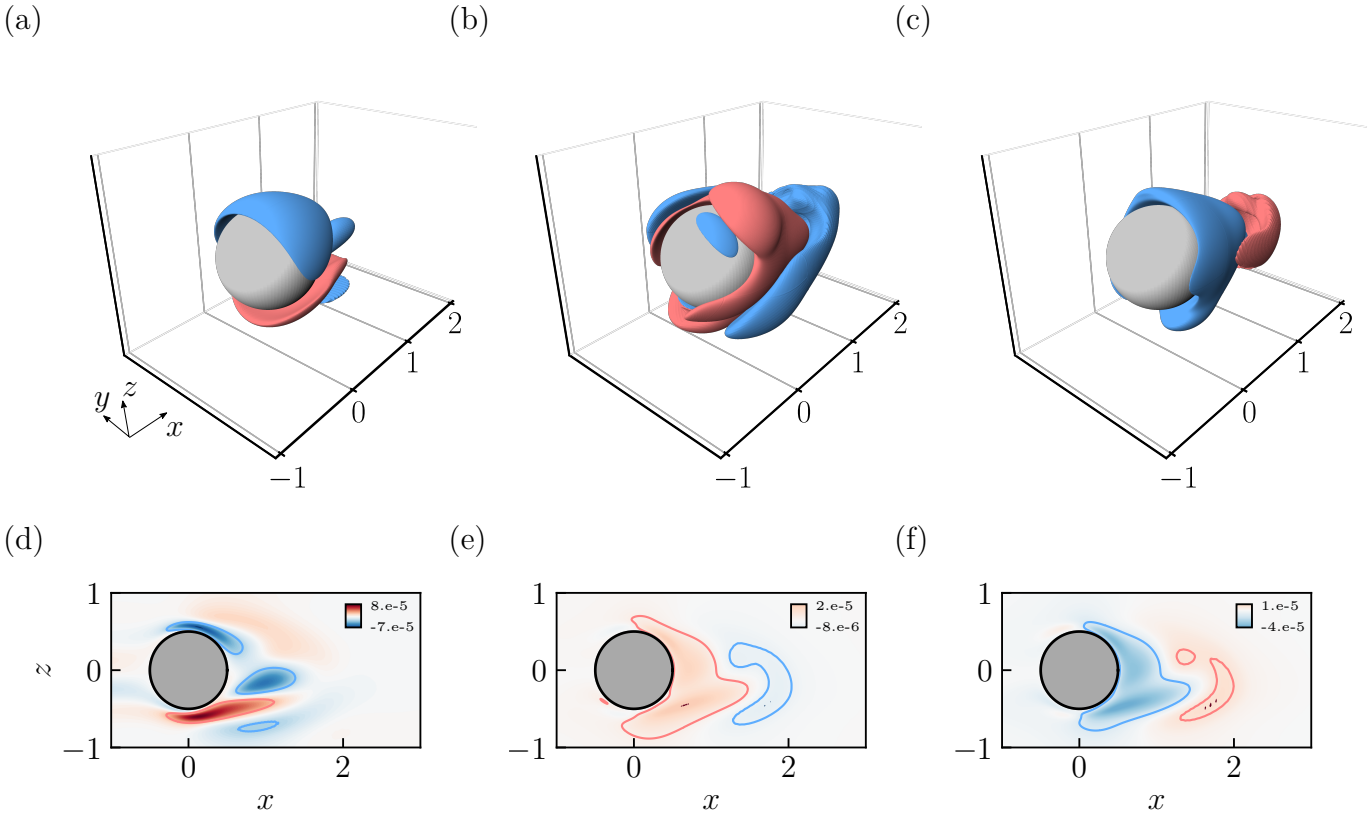


Figure IV.39: 3-D views of optimal forcing at Re = 260, calculated for St = 0.13. (top) Isosurfaces of streamwise (a), cross-stream (b) and transverse (c) perturbation velocities for the values $u_x = \pm 2.5 \times 10^{-5}$, $u_y = \pm 5 \times 10^{-6}$, $u_z \pm 1.1 \times 10^{-5}$. (bottom) Corresponding sectional contour plot at (d) $z = 0$, (e) $z = 0$ and (f) $z = 0$, with solid-lines indicating iso-contours at $u_x = \pm 2.5 \times 10^{-5}$, $u_y = \pm 5 \times 10^{-6}$, $u_z \pm 1.1 \times 10^{-5}$.

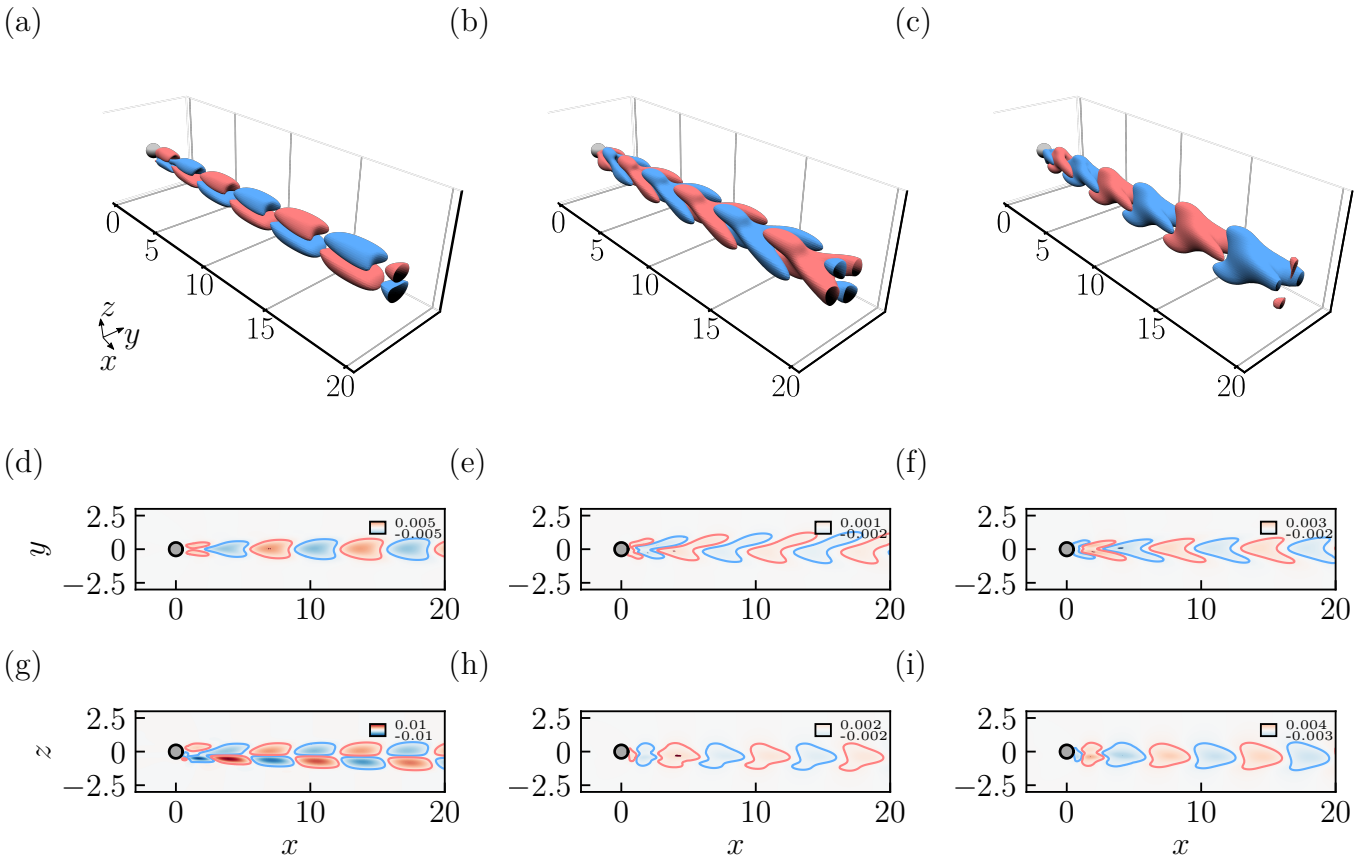


Figure IV.40: 3-D views of optimal response at Re = 260 calculated at St = 0.13. (top) Isosurfaces of streamwise (a), cross-stream (b) and transverse (c) perturbation velocities for the values $u_x = \pm 0.01$, $u_y = \pm 4 \times 10^{-4}$, $u_z \pm 6 \times 10^{-4}$. (bottom) Corresponding sectional contour plot at (d) $z = 0$, (e) $z = 0$, (f) $z = 0$, (g) $y = 0$, (h) $y = 0$ and (i) $y = 0$, with solid-lines indicating iso-contours at $u_x = \pm 0.01$, $u_y = \pm 4 \times 10^{-4}$, $u_z \pm 6 \times 10^{-4}$.

Strouhal number of the least stable oscillating mode calculated at this Reynolds number is 0.133. It turns out that a convective instability might be triggered by exciting the stable base flow at this frequency. The shapes of both the optimal forcing and response calculated at St = 0.13 are respectively shown in fig. IV.39 and IV.40. As expected, the optimal forcing has the same spatial characteristics as the adjoint eigenmode calculated at Re = 280 (see IV.36). The optimal forcing is mainly localized immediately after the spherical body, is convected upstream, and is dominated by streamwise velocity, which is symmetric across the x-z plane. It highlights the zones most sensitive to external forcing to trigger or control this convective instability. The optimal response, in turn, is similar to the unstable direct eigenmode calculated at Re = 280 (see IV.34). It is convected downstream and shares the same symmetry pattern as the optimal forcing. The spatial wavelength for this eigenfunction is $\Lambda_x \simeq 6.1L$, which is of the same order as that measured for the direct eigenmode at Re = 280.

This description of the resolvent analysis carried out on a sub-critical PS base flow completes the global stability analysis of the subsonic laminar 3-D sphere flow. The last step was to perform unsteady computations to compare the results with the modal analysis, which is the aim of the following.

IV.3.8 Unsteady computation of planar-symmetric base flow.

Unsteady nonlinear calculations are conducted at Reynolds numbers of Re = 300 and Re = 320. The physical non-dimensional time-step is set to $\Delta t = 10^{-3}$, resulting in a CFL number around 10 at the wall. As for the two precedent test cases, the Gear method was used for these computations. The base flow, perturbed by the oscillating direct mode, is used as the initial condition. A probe in the wake of the sphere measures the perturbation velocity signal, as shown in fig. IV.41. The temporal evolution of the signal indicates an exponential growth until reaching the limit cycle around $t = 85$.

Fig. IV.42 displays a typical isosurface of the Q-criterion extracted at $t = 124$ from nonlinear unsteady calculations at Re = 300. The shedding of hairpin vortices is identified at a Strouhal number of St = 0.1344, which is consistent with the value reported by Sansica in [107]. This

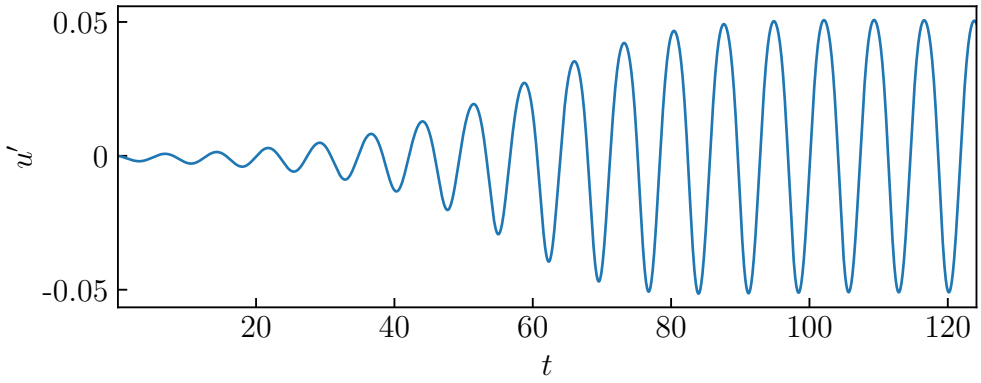


Figure IV.41: Temporal evolution of lift coefficient at Re = 300. Calculation was initialized with PS base flow perturbed by direct unstable mode.

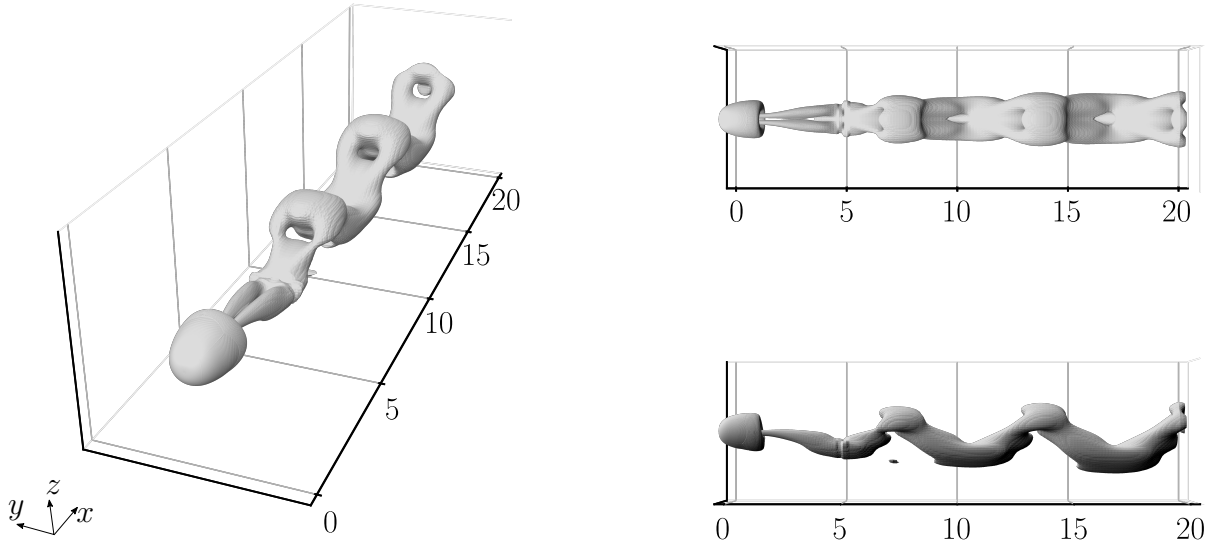


Figure IV.42: 3-D views of isosurface of Q-criterion for the value $Q = 10^{-4}$, for Re = 300 in unsteady computation.

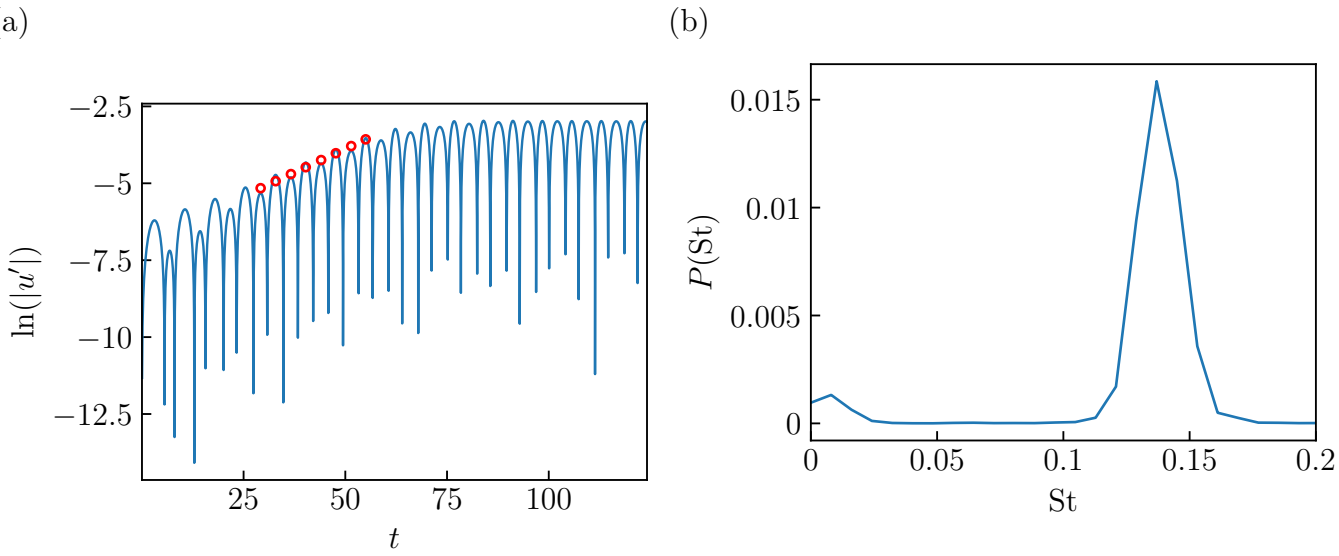


Figure IV.43: (a) Lift coefficient trend over the time where the dotted line represents $\sigma^{\mathrm{NL}} = 0.1248$. (b) Magnitude spectrum of the signal, where a pike at $\mathrm{St}^{\mathrm{NL}} = 0.063667$ is identified.

St	$(\boldsymbol{\sigma}^{\mathbf{L}}, \mathbf{St}^{\mathbf{L}})$	$(\boldsymbol{\sigma}^{\mathbf{NL}}, \mathbf{St}^{\mathbf{NL}})$	$\boldsymbol{\Delta}$**St**	$\boldsymbol{\Delta\sigma}$
300	$(0.0617, 0.1344)$	$(0.0615, 0.1371)$	1.9%	3.2%
320	$(0.1113, 0.1354)$	$(0.1052, 0.1373)$	1.4%	5.8%

Table IV.9: Amplification rate and Strouhal number of linear and nonlinear computations.

sequence of interconnected coherent structures extends and grows far from the sphere boundary at $x = 20$. Additionally, the spatial wavelength $\Lambda_x \simeq 6.5L$ is very close to that identified for the direct unstable mode at $\mathrm{Re} = 280$.

Both σ^{NL} and $\mathrm{St}^{\mathrm{NL}}$ are extracted from the perturbation velocity signal and shown in fig. IV.43a and IV.43b, respectively. The amplification rates obtained from linear calculations are very close to those of nonlinear computations. The same applies to the Strouhal number, for which a relative difference of only 1.4% is observed, even far from the instability threshold at $\mathrm{Re} = 320$. As stated by Sansica [107], this finding contrasts with the case of the laminar cylinder flow, where the prediction of shedding frequency diverges quickly as the Reynolds number increases (see table III.8). While the linear stability theory is less predictive for the amplification rate, it may be enhanced by reducing the tolerance of the eigensolver.

Before concluding this section on the laminar sphere flow, some data concerning the scaling up process for this case will be provided below.

IV.3.9 Data about the scaling up process

As mentioned in the introduction of this chapter, a finer mesh consisting of about 25×10^6 cells (representing 125×10^6 DOF) was also considered. In particular, direct and adjoint computations were carried out on the PS base flow at $\mathrm{Re} = 280$, with the same parameters as for the regular mesh. For both the direct and adjoint unstable modes, the relative error in the associated eigenvalue with respect to that computed with the regular mesh was 4.4%. This small error validates the global stability computations on this refined mesh.

Table IV.10 compares the memory footprint of global stability computations carried out on the

Mesh	$N_{\mathbf{DOF}}$	$N_{\mathbf{CPU}}$	Memory elsA	Memory LU
regular	47×10^6	1200	4.9 TB	$\backsim$ 300 TB
fine	125×10^6	4788	20.5 TB	$\backsim$ 1000 TB

Table IV.10: Quantitative figures from modal stability analysis on regular and fine mesh.

two types of mesh, with the theoretical memory footprint of the corresponding LU factorizations, based on the estimations built in III.1.1. A factor of 50 is identified between the memory footprint of the KFGSM and the LU factorization. This result is satisfying as it allows us to conduct global stability computations on such 3-D configurations. As a reminder, and to the author's knowledge, the largest global stability analysis ever performed with the type of method used in this thesis was carried out by Timme [55] on a 3-D plane configuration composed of about 50×10^6 DOF.

IV.4 Conclusion

This chapter aims to demonstrate the capabilities of the KFGSM on challenging test cases. First, the code is validated on the RANS case of the flow around an OAT15A airfoil. Fixed points of this stiff problem are computed with high accuracy, and the method performs well for direct and adjoint modal analyses. The compatibility with the linear solver and nonlinear URANS computations with a negative-SA turbulence model is also verified. Resolvent analysis is carried out and characterizes both the buffet phenomenon for sub-critical base flow and the secondary Kelvin-Helmholtz instability, as tackled by Sartor [41].

The 3-D case of the laminar flow around a sphere body is then addressed. The computational domain for this case is composed of approximately 10×10^6 DOF. AS and PS base flows are studied. The code allows the prediction of regular and Hopf bifurcations for the AS base flow. A resolvent analysis at sub-critical Reynolds numbers illustrates the similarity between optimal forcing and response and adjoint and direct modes for the Hopf bifurcation. New results concerning adjoint steady mode of AS base flow, and the corresponding structural sensitivity, were demonstrated. A complete description of the Hopf bifurcation occurring in AS base flow at relatively high Reynolds number is also provided, which, to the author's knowledge, have not been shown in the literature.

Finally, the data provided to illustrate the scaling up of the KFGSM demonstrate its efficiency in terms of memory storage relative to classical global stability methods involving LU factorization. The global stability analysis was successfully scaled up, which was the main objective of this thesis.

Chapter V

Conclusion

As discussed in chapter I, global stability analysis is a powerful tool for improving the understanding of unsteady flows. After a base flow has been computed, this method may be seen as a thorough post-processing that returns modes of the flow that may be linearly amplified and potentially responsible for the unsteadiness. In particular, it could be used to predict the appearance of the rotating stall phenomenon occurring in axial compressors like the CME2, and thus help aircraft manufacturers to adapt their design to delay instability, reduce the surge margin, and improve performance. However, this methodology requires the inversion of many linear systems. When addressing large-scale problems exceeding 10×10^6 DOF, inversions cannot be managed using direct methods like LU factorization, as is often the case [22, 25].

The purpose of this thesis was to develop a robust method capable of scaling up global stability analysis to industrial cases like the CME2 compressor. To tackle this problem, the main constraint was to find a way to solve extremely large complex-valued linear systems while preserving high-performance computing optimizations, with the knowledge that complex arithmetic cannot be vectorized. To do so, a new method was designed in which complex problems were reformulated in the real-equivalent K formulation [60], a formulation that preserves the sparsity pattern of the original complex operators, and where inversions are managed by the iterative preconditioned GMRES algorithm [58, 52]. A mathematical background was established to build the method, named the K-Formulated Global Stability Method (KFGSM). Even not presented in this work, the latter was first validated on a toy code that solved the global stability problems for the nonlinear advection-diffusion Burgers equation [59].

To implement the KFGSM in the elsA software, several elementary bricks needed to be assembled. The first step was to adapt the existing preconditioned GMRES based on obtaining Jacobian-vector products using Algorithmic Differentiation (AD) through the Tapenade software developed by Hascoët and Pascual [76], and the efficient preconditioning tools developed by Guilbert [51] to solve the real-valued problems arising in the computation of fixed points in compressible CFD. A collaboration with Hascoët led to a new feature of Tapenade that computes the forward and backward K-formulated Jacobian-vector products. The preconditioning tools of Guilbert were also adapted so that the resolution of linear systems with preconditioned GMRES is operative on K-formulated problems. The final step was to implement the Krylov-Schur eigensolver in real arithmetic, where the Schur matrix is in real form, incorporating the K-formulated GMRES. Slight modifications in the convergence test needed to be carried out to ensure the algorithm returned the eigenpairs of the original system and not the K-formulated one. The algorithm have been developed in three versions corresponding to the resolution of eigenproblems arising in direct and adjoint modal analysis and resolvent analysis.

The method was challenged on three test cases of gradual difficulty. The simplest is the laminar flow around a cylinder. Fixed points at various Reynolds numbers were computed with Flexible GMRES preconditioned with ILU(3) factorization. A good convergence near machine precision is obtained, demonstrating the robustness of the fixed point computation method developed by [51]. The direct, adjoint, and resolvent analysis conducted with the KFGSM are in good agreement with the literature [29, 30], validating the method on the laminar case and demonstrating its capability to predict the onset of von K'arm'an vortex street. Moreover, a comparison with unsteady computations showed the compatibility between linear and nonlinear solvers.

The second problem addressed is the RANS case of the flow around the OAT15A airfoil. This case has been selected due to its high stiffness [51]. Fixed points were computed in sub and supercritical regimes with high accuracy. The KFGSM showed good results at predicting the buffet onset on this RANS case with direct and adjoint modal analysis. Such results were also successfully compared with URANS computations. From the resolvent analysis, two energy peaks were observed. The largest one corresponds to the buffet frequency, while the second is linked with a secondary instability that is a Kelvin-Helmholtz one [41], illustrating the capability of the resolvent analysis to study in detail the receptivity of the flow within a large range of frequencies.

The code was ultimately tested on the three-dimensional subsonic laminar flow around a sphere body. The computational grid was composed of about 10×10^6 degrees of freedom (DOF), representative of the industrial cases the KFGSM has been designed for. After fixed point computation near machine precision were carried out for axi and planar-symmetric base flows, the two bifurcations were studied in detail with direct, adjoint, and resolvent analysis. In particular, our fixed point computation method allowed the calculation of axi-symmetric base flow above a second bifurcation that, to the author's knowledge, had never been documented. The resolution of both direct and adjoint eigenproblems allowed the calculation of the structural sensitivity [30]. The second bifurcation occurring on the planar-symmetric base flow was also successfully characterized. On this particular type of base flow, a global stability analysis was even performed on a refined mesh composed of about 125×10^6 DOF. Such results validated the scale up of global stability analysis, as the largest reported global stability case is the characterization of the buffet phenomenon on a full airplane configuration composed of about 50×10^6, carried out by [55].

Chapter VI

Perspectives

Various perspectives can be proposed for this work. At ONERA, important work is currently being carried out to improve the GMRES solver. For instance, Jang [102] has improved the formation of the Krylov basis by implementing the randomized Gram-Schmidt process in elsA. This process can be less computationally expensive than the classical Gram-Schmidt process, while being at least as numerically stable as the modified Gram-Schmidt process used in this work. Another perspective for improving the GMRES algorithm is to improve the preconditioning tools. As seen with the laminar flow around the sphere body, the memory cost for the storage of the preconditioner remains high when dealing with large problems. For problems stiffer than the sphere case, ILU(k) factorization may not be sufficient to allow the GMRES to converge, and more advanced techniques like the HLU may need to be used. However, the HLU method is currently unusable on the ONERA cluster for such large cases. To circumvent the potential memory issue related to preconditioning, a thesis is currently being carried out by Dubois at ONERA. The aim is to reduce the memory cost of preconditioning.

Improvements to the Krylov-Schur solver can also be cited. In fact, given the time limit imposed on the computing cluster at ONERA, a job may not be sufficient to carry out the entire global stability analysis. To this end, it could be useful to include restarting so that the Krylov-Schur basis can be extracted at the end of the computation if not all the desired eigenpairs have yet converged. This basis could then be read in a subsequent job to restart the computation where it stopped in the previous job.

Another natural perspective of this work is to perform a global stability analysis on the isolated compressor CME2 to characterize the genesis of the rotating stall phenomenon. The full-annulus case represents a domain of about 30×10^6 cells. The usual boundary conditions for this case are an injection condition at the inlet, no-slip at walls, and a radial equilibrium with a valve law at the outlet. However, such a radial condition is not yet linearized in the elsA software. To circumvent this difficulty, computations have been carried out with an extended domain representing about 40 chords, and an outflow condition has been applied at the outlet. To check the validity of this configuration, compressor characteristics have been calculated by conducting RANS computations with the LU-SGS method (which doesn't require the linearized matrix-vector products), where the convective fluxes are discretized with a second-order Roe scheme, the diffusive fluxes with the $5p_{\text{cor}}$ method, and the turbulence model is the negative Spalart-Allmaras. The results are given in fig. VI.1. The pressure distribution is slightly lower in the extended vein with outflow condition compared with the normal vein with radial equilibrium condition. However, until the stability limit occurring around $Q = 4\ \text{kg}\,\text{s}^{-1}$, the extended vein with outflow condition is in good agreement with the extended vein with radial equilibrium. Based on these positive results, a URANS computation

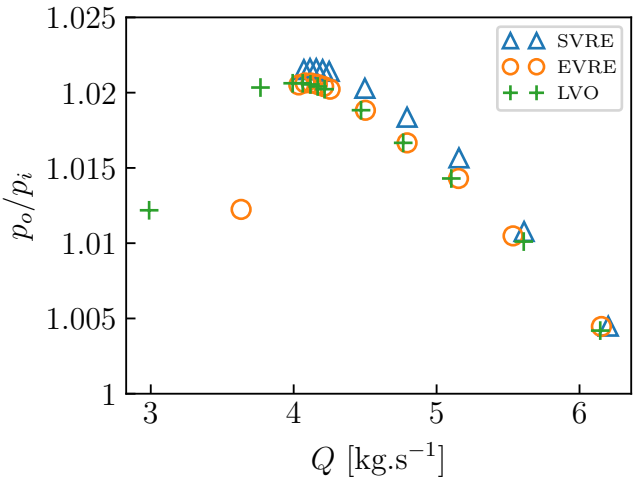


Figure VI.1: Characteristic curve of the single channel flow for various tested cases, with SVRE for Short Vein Radial Equilibrium; EVRE for Extended Vein Radial Equilibrium; EVO for Extended Vein Outflow.

has been carried out on the full-annulus CME2 with the extended vein and outflow condition over 20 laps. The signal recorded by three probes placed on three successive channels is shown in fig. VI.2a, which clearly highlights the appearance of self-sustained oscillations characteristic of the rotating stall. The instantaneous entropy field after 12 laps is displayed in fig. VI.2b, and shows that each stall cell has merged into one large cell extending over 15 channels.

This may be challenging because, in the author's opinion, the instability may involve the rotation of the stall cells, making it a stiff problem. One approach is to converge a detached base flow on a single-channel case, duplicate it, and use the result as the starting field for the base flow computation. Another possibility is to use the Schmid approach, already utilized by Paladini in a previous study, which allows for the computation of eigenpairs of the full-annulus compressor by

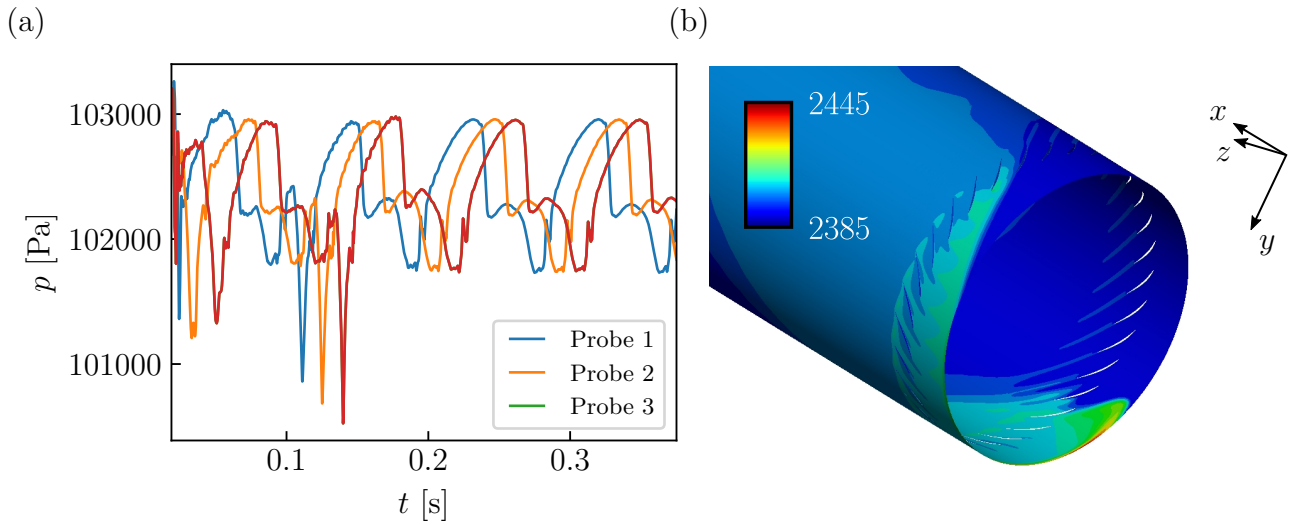


Figure VI.2: (a) Time evolution of a pressure signal recorded on three probes during URANS computation of the full-annulus compressor CME2. (b) Instantaneous entropy field after 12 revolutions.

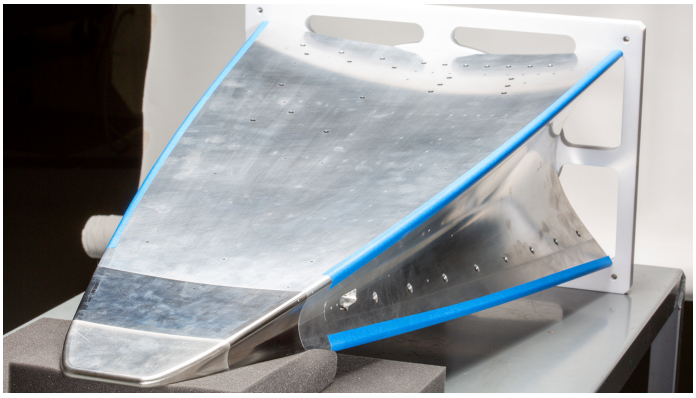


Figure VI.3: Hypersonic BOLT configuration.

performing a stability analysis on only three channels.

The continuation of this work is to perform a fixed point computation to obtain a converged base flow with high accuracy. This may be challenging because, in the author's opinion, the instability may involve the rotation of the stall cells. Thus, the unstable base flow has to be already detached and could result in a stiff problem. A possibility is to converge a detached base flow on a single channel case, to duplicate it and use the result as the starting field for the base flow computation. Another possibility would be to benefit from the periodicity of the compressor CME2 and use the Schmid [115] approach. The latter, already exploited by Paladini in [27], allows the computation of eigenpairs of the full-annulus compressor by carrying a stability analysis on only three channels.

More globally, the tools developed in elsA in this work allow for the analysis of modal and non-modal phenomena in an industrial context. This will allows for the characterization of oscillating phenomena on a linear basis, which is very interesting for predicting undesirable unsteady phenomena. This work also provides robust tools to characterize noise-amplified phenomena through resolvent analysis. In a flow, multiple unsteadiness may coexist with highly variable time-scales. A computation that captures each phenomenon requires performing with all time-scales, which could result in very long and costly calculations. The resolvent is hence of interest for many applications involving super or hypersonic flows and could improve the knowledge about the laminar-to-turbulent transition scenario on such configurations. One can cite, for instance, the studies currently carried out at ONERA on the BOLT configuration, presented in fig. VI.3, which was expressly designed to study the laminar-to-turbulent transition on hypersonic configurations.

References

[1] I. J. Day. Stall, Surge, and 75 Years of Research. *Journal of Turbomachinery*, 138(1), 10 2015.

[2] E. M. Greitzer. Surge and Rotating Stall in Axial Flow Compressors—Part I: Theoretical Compression System Model. *Journal of Engineering for Power*, 98:190–198, 1976.

[3] E. M. Greitzer. Surge and Rotating Stall in Axial Flow Compressors—Part II: Experimental Results and Comparison With Theory. *Journal of Engineering for Power*, 98:199–211, 1976.

[4] N. Courtiade, X. Ottavy, and N. Gourdain. Experimental investigation of rotating stall in a high speed multi-stage axial compressor. *9th European Conference on Turbomachinery: Fluid Dynamics and Thermodynamics, ETC 2011 - Conference Proceedings*, 1, 01 2011.

[5] G. Margalida, P. Joseph, O. Roussette, and A. Dazin. Comparison and sensibility analysis of warning parameters for rotating stall detection in an axial compressor. *International Journal of Turbomachinery, Propulsion and Power*, 5(3), 2020.

[6] A. Dazin, P. Joseph, F. Romano, Q. Gallas, J. Marty, G. Aigouy, M. Stöβel, and R. Niehuis. The ACONIT project: an innovative design approach of active flow control for surge prevention in gas turbines. In *10th EASN International Conference on "Innovation in Aviation & Space to the Satisfaction of the European Citizens"*, volume 1024 of *IOP Conference Series: Materials Science and Engineering,*, VIRTUEL, Italy, September 2020.

[7] J. Hamilton, D. and O. Ismael, J. Parallel Simulation of Stall Hysteresis in a Two Dimensional Rotor Blade Row. In *International Society for Air Breathing Engines*, 2003.

[8] L. He. Computational Study of Rotating-Stall Inception in Axial Compressors. *Journal of Propulsion and Power*, 13(1):31–38, 1997.

[9] N. Gourdain. *Simulation numérique des phénomènes de décollement tournant dans les compresseurs axiaux*. PhD thesis, École centrale de Lyon , 2005. Thèse de doctorat.

[10] J. Marty, L. Castillon, and P. Joseph. Numerical Investigations on the Rotating Stall in an Axial Compressor and its Control by Flow Injection at Casing. In *Turbomachinery Technical Conference and Exposition GT2022*, page 12p., Rotterdam, Netherlands, June 2022. ASME.

[11] N. Gourdain. Prediction of the unsteady turbulent flow in an axial compressor stage. Part 1: Comparison of unsteady RANS and LES with experiments. *Computers and Fluids*, vol. 106:pp. 119–129, 2015.

[12] W. Thomson. XLVI. Hydrokinetic solutions and observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 42(281):362–377, 1871.

[13] H. von Helmholtz. Über discontinuierliche flüssigkeits-bewegungen [on the discontinuous movements of fluids]. *Monatsberichte der Königlichen Preussische Akademie der Wissenschaften zu Berlin [Monthly Reports of the Royal Prussian Academy of Philosophy in Berlin]*, 23:215 – 228, 1868.

[14] H. Bénard. Comptes rendus de l'Académie des Sciences; T. CXLV; 2e semestre 1907. *J. Phys. Theor. Appl.*, 147:839 — 842, 970 — 972, 1908.

[15] T. von Karman and H. Rubach. The mechanism of the uid and air-resistance. *Phys. Z*, 13:49, 1912.

[16] O. Reynolds. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal society of London*, (174):935–982, 1883.

[17] C. C. Lin. On the stability of two-dimensional parallel flows. *Proceedings of the National Academy of Sciences*, 30(10):316–324, 1944.

[18] G. K. Batchelor and A. Gill. Analysis of the stability of axisymmetric jets. *Journal of fluid mechanics*, 14(4):529–551, 1962.

[19] P. G. Drazin and W. H. Reid. Hydrodynamic stability. *Press, Cambridge*, pages 8–14, 1981.

[20] P. Schmid and D. Henningson. *Stability and Transition in Shear Flows*, volume 142. Springer, 01 2001.

[21] P. Huerre and P. A. Monkewitz. Local and global instabilities in spatially developing flows. *Annual review of fluid mechanics*, 22(1):473–537, 1990.

[22] D. Sipp, O. Marquet, Philippe Meliga, and Alexandre Barbagallo. Dynamics and control of global instabilities in open-flows: A linearized approach. *Applied Mechanics Reviews*, 63(3):030801, 2010.

[23] V. Theofilis. Advances in global linear instability analysis of nonparallel and three-dimensional flows. *Progress in aerospace sciences*, 39(4):249–315, 2003.

[24] R. T. Pierrehumbert and S. Widnall. The two-and three-dimensional instabilities of a spatially periodic shear layer. *Journal of Fluid Mechanics*, 114:59–82, 1982.

[25] V. Theofilis. Global Linear Instability. *Annual Review of Fluid Mechanics*, 43(1):319–352, 2011.

[26] A. Tezuka and K. Suzuki. Three-dimensional global linear stability analysis of flow around a spheroid. *Aiaa Journal - AIAA J*, 44:1697–1708, 08 2006.

[27] E. Paladini, S. Beneddine, J. Dandois, D. Sipp, and J.-C. Robinet. Transonic buffet instability: From two-dimensional airfoils to three-dimensional swept wings. *Phys. Rev. Fluids*, 4:103906, Oct 2019.

[28] X. Xie, Z. Liu, and X. Li. Stall inception analysis of a transonic compressor based on a global stability analysis method. In *Proceedings of Global Power and Propulsion Society*, 2022.

[29] O. Marquet, D. Sipp, and L. Jacquin. Sensitivity analysis and passive control of cylinder flow. *Journal of Fluid Mechanics*, 615:221–252, 2008.

[30] F. Giannetti and P. Luchini. Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics*, 581:167–197, 2007.

[31] V. Citro, L. Siconolfi, D. Fabre, F. Giannetti, and P. Luchini. Stability and sensitivity analysis of the secondary instability in the sphere wake. *AIAA Journal*, 55:1–8, 07 2017.

[32] 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. *Applied Mechanics Reviews*, 66(2), 03 2014. 024803.

[33] V. Romanov. Stability of plane-parallel Couette flow. *Functional Analysis and Its Applications*, 7:137–146, 1973.

[34] A. Lundbladh and A. V. Johansson. Direct simulation of turbulent spots in plane couette flow. *Journal of Fluid Mechanics*, 229:499–516, 1991.

[35] S. Reddy, P. Schmid, and D. Henningson. Pseudospectra of the Orr–Sommerfeld Operator. *Siam Journal on Applied Mathematics - SIAMAM*, 53, 02 1993.

[36] L. N. Threfethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261(5121):578–584, 1993.

[37] S. Beneddine. *Characterization of unsteady flow behavior by linear stability analysis*. Thèse de doctorat, Université Paris-Saclay, 2017.

[38] S. Beneddine, D. Sipp, A. Arnault, J. Dandois, and L. Lesshafft. Conditions for validity of mean flow stability analysis. *Journal of Fluid Mechanics*, 798:485–504, 2016.

[39] Brandt, L. and Sipp, D. and Pralits, J. O. and Marquet, O. Effect of base-flow variation in noise amplifiers: the flat-plate boundary layer. *Journal of Fluid Mechanics*, 687:503–528, 2011.

[40] X. Garnaud, L. Lesshafft, P. J. Schmid, and P. Huerre. The preferred mode of incompressible jets: linear frequency response analysis. *Journal of Fluid Mechanics*, 716:189–202, 2013.

[41] F. Sartor, C. Mettot, and D. Sipp. Stability, receptivity, and sensitivity analyses of buffeting transonic flow over a profile. *AIAA Journal*, 53:1980–1993, 07 2015.

[42] C.-A. Yeh and K. Taira. Resolvent-analysis-based design of airfoil separation control. *Journal of Fluid Mechanics*, 867:572–610, 2019.

[43] M. Lugrin. *Study of the viscous interaction with separation at hypersonic regime*. PhD thesis, Institut Polytechnique de Paris, December 2021.

[44] J. G. F. Francis. The QR Transformation A Unitary Analogue to the LR Transformation—Part 1. *The Computer Journal*, 4(3):265–271, 01 1961.

[45] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics*, 9(1):17–29, 1951.

[46] G. Stewart. A Krylov-Schur Algorithm for Large Eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23, 09 2000.

[47] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Society for Industrial and Applied Mathematics, 2011.

[48] K. N. Christodoulou and L. E. Scriven. Finding leading modes of a viscous free surface flow: An asymmetric generalized eigenproblem. *Journal of scientific computing*, 3(4):355–406, 1988.

[49] J. R. Bunch and J. E. Hopcroft. Triangular factorization and inversion by fast matrix multiplication. *Mathematics of Computation*, 28(125):231–236, 1974.

[50] Jean-Yves L'Excellent. Multifrontal Methods: Parallelism, Memory Usage and Numerical Aspects, September 2012.

[51] N. Guilbert. *Amélioration de l'inversion de grand systèmes creux pour la simulation numérique en mécanique des fluides*. Thèse de doctorat, Université Pierre Marie Curie, 2021.

[52] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, second edition, 2003.

[53] C. J. Mack and P. J. Schmid. A preconditioned Krylov technique for global hydrodynamic stability analysis of large-scale compressible flows. *Journal of Computational Physics*, 229(3):541–560, 2010.

[54] S. Timme and R. Thormann. Towards Three-Dimensional Global Stability Analysis of Transonic Shock Buffet. 06 2016.

[55] S. Timme. Global instability of wing shock-buffet onset. *Journal of Fluid Mechanics*, 885:A37, 2020.

[56] I. Mary. Flexible aerodynamic solver technology in an HPC environment.

[57] B. Maugars, S. Bourasseau, C. Content, B. Michel, B. Berthoul, J. Nunez Ramirez, I. Salah el Din, P. Raud, and L. Hascoët. Algorithmic Differentiation for an effcient CFD solver. In *ECCOMAS*, 2022.

[58] Y. Saad and M. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *Siam Journal on Scientific and Statistical Computing*, 7:856–869, 1986.

[59] J. M. Burgers. A mathematical model illustrating the theory of turbulence. *Advances in applied mechanics*, 1:171–199, 1948.

[60] D. Day and M. Heroux. Solving Complex-Valued Linear Systems via Equivalent Real Formulations. *Siam Journal on Scientific Computing*, 23, 08 2001.

[61] L. Cambier, S. Heib, and S. Plot. The Onera elsA CFD software: input from research and feedback from industry. *Mechanics & Industry*, 14(3):159–174, 2013.

[62] C. Hirsch. *Numerical Computation of Internal and External Flows*. Elsevier, second edition, 2007.

[63] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer Berlin Heidelberg, 2009.

[64] S. K. Godunov and I. Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik*, 47(89)(3):271–306, 1959.

[65] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357 – 372, 1981.

[66] B. Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23(3):276 – 299, 1977.

[67] C. Bogey and C. Bailly. A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *Journal of Computational Physics*, 194(1):194 – 214, 2004.

[68] O. Guba and J. Lorenz. Continuous spectra and numerical eigenvalues. *Mathematical and Computer Modelling*, 54(11):2616 – 2622, 2011.

[69] A. Crivellini and F. Bassi. An implicit matrix-free Discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. *Computers Fluids*, 50:81–93, 11 2011.

[70] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.

[71] J. E.V. Peter and R. P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, 2010.

[72] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 1988.

[73] J. Peter. *Contributions to discrete adjoint method in aerodynamics for shape optimization and goal-oriented mesh-adaptation*. Habilitation à diriger des recherches, UNIVERSITE DE NANTES, September 2020.

[74] C. Mettot. *Linear stability, sensitivity, and passive control of turbulent flows using finite differences*. Thèse de doctorat, 2013.

[75] D. Knoll and D. Keyes. Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications. *Journal of Computational Physics*, 193:357–397, 01 2004.

[76] L. Hascoet and V. Pascual. The Tapenade Automatic Differentiation tool: principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3), 2013.

[77] P. Luchini and A. Bottaro. Adjoint Equations in Stability Analysis. *Annual Review of Fluid Mechanics*, 46(1):493–517, 2014.

[78] J. J. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29:245 – 262, 2003.

[79] C. Paulin. Automatic verification of algorithmically differentiated code. In *ECCOMAS congress*, 2022.

[80] M. P. Juniper, A. Hanifi, and V. Theofilis. Modal Stability Theory: Lecture notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Applied Mechanics Reviews*, 66(2), 03 2014.

[81] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: A Portable Linear Algebra Library for High-Performance Computers. In *Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, Supercomputing '90, page 2–11, Washington, DC, USA, 1990. IEEE Computer Society Press.

[82] H. Vicente, E. R. Jose, and V. Vicente. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software*, 31(3):351–362, 2005.

[83] B. T. Chu. On the energy transfer to small disturbances in fluid flow (part i). *Acta Mechanica*, 1:215–234, 1965.

[84] J. R. Rice. Experiments on Gram-Schmidt Orthogonalization. *Mathematics of Computation*, 20(94):325–328, 1966.

[85] D. Kressner. *Numerical Methods for General and Structured Eigenvalue Problems*, volume 46. Springer, 01 2005.

[86] V. Hernández, J. E. Roman, A. Tomás, and V. Vidal. Krylov-Schur Methods in SLEPc. Technical report, Universitat Politècnica de València, 2007.

[87] O. Marquet. *Stabilité globale et contrôle d'écoulements de recirculation*. PhD thesis, 2007.

[88] W. Sutherland. LII. The viscosity of gases and molecular force. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 36(223):507–531, 1893.

[89] L. Prandtl. *Applications of modern hydrodynamics to aeronautics*. US Government Printing Office, 1925.

[90] P. Walters. *An Introduction to Ergodic Theory*. Graduate Texts in Mathematics. Springer, 1982.

[91] A.J. Favre. *The equations of compressible turbulent gases*. Defense Technical Information Center, 1965.

[92] P. Spalart and S. Allmaras. *A one-equation turbulence model for aerodynamic flows*.

[93] S. R. Allmaras and F. T. Johnson. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *Seventh international conference on computational fluid dynamics (ICCFD7)*, volume 1902. ICCFD7-1902 Big Island, Hawaii, 2012.

[94] C. Corre. Contribution à la simulation et à l'analyse des écoulements compressibles, 01 2004.

[95] S. Yoon and A. Jameson. *An LU-SSOR scheme for the Euler and Navier-Stokes equations.*

[96] Sébastien Bourasseau. *Contribution à une méthode de raffinement de maillage basée sur le vecteur adjoint pour le calcul de fonctions aérodynamiques.* Theses, Université Nice Sophia Antipolis, Dec 2015.

[97] P.-Y. Outtier. *A new dynamic code architecture for CFD computations : application to the development of an overset-grid compact high-order solver for compressible aerodynamics.* Theses, Ecole nationale supérieure d'arts et métiers - ENSAM, September 2014.

[98] C. Weber. *Développement de méthodes implicites pour les équations de Navier-Stokes moyennées et la simulation des grandes échelles : application à l'aérodynamique externe.* Thèse de doctorat, IMFT, 1998.

[99] W. Givens. Computation of Plain Unitary Rotations Transforming a General Matrix to Triangular Form. *Journal of the Society for Industrial and Applied Mathematics*, 6(1):26–50, 1958.

[100] G. Aditi, L. Cao, and J. Xiangmin. A comparison of preconditioned krylov subspace methods for large-scale nonsymmetric linear systems, 2018.

[101] L. Giraud and S. Gratton. A Set of Flexible GMRES Routines for Real and Complex Arithmetics on High Performance Computers. *ACM Transactions on Mathematical Software - TOMS*, 09 2000.

[102] Y. Jang, L. Grigori, E. Martin, and C. Content. Randomized flexible GMRES with Singular Vectors Based Deflated Restarting. *Journal of Computational Physics*, 2022.

[103] D. Lecas. Vectorisation SIMD, 2019.

[104] R. W. Aldhahcri and H. K. Khalil. A real schur form method for modeling singularly perturbed systems. In *1988 American Control Conference*, pages 1716–1721. IEEE, 1988.

[105] P. Cinnella and C. Content. High-order implicit residual smoothing time scheme for direct and large eddy simulations of compressible flows. *Journal of Computational Physics*, 326:1–29, 12 2016.

[106] D. Barkley. Linear analysis of the cylinder wake mean flow. *EPL (Europhysics Letters)*, 75(5):750, 2006.

[107] A. Sansica, J.-Ch. Robinet, F. Alizard, and E. Goncalves. Three-dimensional instability of a flow past a sphere: Mach evolution of the regular and hopf bifurcations. *Journal of Fluid Mechanics*, 855:1088–1115, 2018.

[108] Brunet, V. and Deck, S. and Molton, P. and Thiery, M. A complete Experimental and Numerical Study of the Buffet Phenomenon over the OAT15A Airfoil. 01 2005.

[109] J.D. Crouch, A. Garbaruk, and D. Magidov. Predicting the onset of flow unsteadiness based on global instability. *Journal of Computational Physics*, 224(2):924–940, 2007.

[110] B. Godard. Compresseurs et turbines, 2017.

[111] C. Tarsia Morisco. *Dynamique nonlinéaire et stabilité linéaire d'une tuyère sur-détendue*. Thèse de doctorat, ENSAM, 2020.

[112] A. Sansica, Y. Ohmichi, J.-C. Robinet, and A. Hashimoto. Laminar supersonic sphere wake unstable bifurcations. *Physics of Fluids*, 32:126107, 2020.

[113] R. Magarvey and Roy Bishop. Transition ranges for three-dimensional wakes. *Canadian Journal of Physics*, 39:1418–1422, 02 2011.

[114] D. Fabre, A. Franck, and J. Magnaudet. Bifurcations and symmetry breaking in the wake of axisymmetric bodies. *Physics of Fluids*, 20, 05 2008.

[115] P. J. Schmid, M. F. de Pando, and N. Peake. Stability analysis for n-periodic arrays of fluid systems. *Physical Review Fluids*, 2(11):113902, 2017.

Remerciements

Ça y est, voilà le bout de l'aventure. Après 1201 jours l'esprit constamment occupé par ce travail de thèse, voici qu'il se termine. Ce fut un parcours semé d'embûches qui m'a beaucoup appris sur moi-même et ma capacité de résilience face à l'obstacle. Ces 39 mois, entrecoupés de confinements et de couvre-feux, m'ont fait progresser sur le plan intellectuel, et devenir le scientifique que je suis désormais. Le chemin fut long et difficile, entre l'année que j'ai mise pour pleinement comprendre mon sujet, les (grosses) galères de développements dans elsA, les soucis pour reproduire certains résultats de la littérature, ou encore cette folle course contre-la-montre pour rendre le manuscrit à temps. Courir un marathon en chaussettes, c'est dur. Courir un semi-marathon pieds nus, c'est très dur. Mais aller au bout de cette thèse aura constitué le plus grand défi que j'ai réalisé dans ma vie. Et pour cela, je crois pouvoir dire que je suis fier de moi.

Évidemment, tout cela n'aurait pas été possible sans mon équipe d'encadrement. Aussi, je tiens en premier lieu à vous remercier Séb et Cédric. Tout le temps que vous m'avez consacré, ce soucis du détail permanent, et vos immenses connaissances de la CFD m'ont aidé à développer une certaine rigueur qui me servira toujours. J'ai aussi beaucoup apprécié ce séjour à Oslo en votre compagnie. Jean-Christophe, je te remercie d'avoir été mon directeur de thèse. Je me souviendrai longtemps de tous ces échanges très intéressants à l'ENSAM, de tes explications toujours claires et précises, et de ta présence rassurante même dans les moments tendus. Samir, merci d'avoir été là lorsque j'en avais besoin pour me partager, calmement, ton sens de la physique et tes grandes connaissances relatives à la stabilité globale. Denis S., c'est grâce à toi que j'ai réalisé cette thèse. J'ai toujours apprécié l'extrême finesse de tes interventions lors de nos points de thèse. Beaucoup de fois, tu m'as permis de dissiper des doutes ou d'approfondir une idée, parfois au moyen d'une seule phrase, et pour cela je te remercie.

J'ai également une pensée pour tous les collègues avec qui j'ai passé ces trois ans. Niels, sans ta sorcellerie de préconditionnement, je n'aurais jamais pu mener à bien cette thèse. Christina, j'ai beaucoup apprécié toutes nos conversations de couloir qui continuent désormais sur WhatsApp. Cyril, nous avons connu sensiblement les mêmes galères dans elsA, c'était dur mais précieux de ne pas être seul dans ce bourbier. Je te souhaite tout mon courage pour finir ta thèse !! Je pense aussi à Julien C., Clément, ainsi que tous les autres membres de l'unité CLEF avec qui les discussions étaient toujours passionantes et éclairées. Denis G., je te remercie particulièrement pour m'avoir donné autant de liberté dans mon travail. Au début de cette thèse, j'ai également effectué un séjour dans l'unité H2T, dont je remercie tous les membres pour leur accueil. Julien M., merci de m'avoir accordé autant de temps pour la mise en place du CME2, malgré ton emploi du temps très chargé. Enfin, côté DynFluid, j'aimerais te remercier Nicolas pour tous nos échanges. Tes questions toujours perspicaces m'ont fait réfléchir et avancer tout au long de cette thèse.

Le 23 janvier 2023, est venu le moment de soutenir mon travail. Cette journée, je ne l'oublierai jamais. En premier lieu, je remercie tous les membres du jury pour votre bienveillance lors de nos échanges post-présentation d'une part, et vos analyses très fines de mon travail d'autre part. Cela a

été vraiment formateur pour moi. Durant cette soutenance, la salle Contensou du site de Châtillon était presque pleine. Lorsque j'ai envoyé des invitations pour cet évènement début janvier, j'étais très surpris et touché de voir que 26 personnes allaient répondre présentes. D'abord, les potes de Lyon, ma deuxième famille. Céd, Mamoule, Jé, Anto, Rapace, Rapacette, Camouille, Laure, vous avez fait le déplacement jusqu'à Châtillon city pour venir me soutenir et c'est quelque chose qui restera à jamais gravé dans mon cœur. Ensuite, tous les potes de la Kill et proches affiliés. Dufal, tu crois qu'on savait dans quoi on se lançait il y a 5 ans, au détour d'une discussion au Bar'Bu, quand on a décidé quasiment ensemble de faire une thèse ? Je n'oublierai pas ces trois premiers mois de thèse, incroyablement productifs, passés dans le même bureau, de même que toutes nos discussions méca flu. Tiens bon, tu n'es plus qu'à quelques mois de finir ta thèse !! Je remercie bien chaleureusement tous les autres qui ont fait le déplacement pour être avec moi ce jour-là : Bib, Louis, S'il vint, Gorce, Ch'Arvor, Larrin, Marblé, Thoyus, Rochette, Nounouck, Lucho. Je remercie aussi ceux qui n'ont pas pu être là mais qui m'ont probablement soutenu à distance comme Étienne, Saman, Dr. Labat ou encore futur Dr. Beuz'. Vous êtes tous de vrais amis sur qui je pense toujours pouvoir compter.

J'en arrive à ma famille que j'aimerais remercier pour son soutien indéfectible. Caro, Ana, vous avez toujours été des pilliers dans ma vie, des sources de soutien, de sagesse, de partage de connaissances mais surtout d'amour inconditionnel. Je vous aime si fort. Je pense également à vos familles respectives : Charlotte, dont j'ai eu la chance d'être le coloc pendant 2 mois, votre petite Sashounette qui, j'en suis sûr, ira loin, mais aussi François, mon grand Malo et ma petite Lou. Ensuite, Manu et Laulau, mes p'tits rarons (sans oublier Roro, Yummi et Car bien sûr), vous avez été un bol d'air incroyable lorsque nous nous voyions régulièrement à la fin de la rédaction de mon manuscrit, et le fait que vous veniez me voir le jour J m'a tellement touché. Je vous remercie de tout mon cœur. J'ai une grande pensée pour tous mes oncles, tantes, cousins et cousines qui, pour la plupart, m'ont suivi à distance. Vous n'étiez pas là physiquement, mais bien présents dans mon esprit. Pap, Mam, j'ai beaucoup de chance de vous avoir. Vous êtes des épaules sur qui toujours me reposer en cas de doutes, des soutiens infaillibles, et avec vous, il y a toujours une solution à tous les problèmes ! Je vous aime très fort. Je pense aussi à Jean-Phi et Barbara, merci d'être vous et d'avoir été là pour moi depuis toutes ces années. J'ai également une pensée ému pour mes quatre grands-parents qui, j'en suis sûr, seraient heureux que leur petit-fils soit devenu docteur.

Pour finir, Anne. La réussite de cette thèse, c'est un peu la tienne tant nous avons été proches depuis tout ce temps. Elle a pris beaucoup de place, et je t'en ai fait baver, mais tu as toujours été là pour faire redescendre la tension et dissiper mes angoisses. La bonne tenue de cette soutenance, je la dois en grande partie à toi et je t'en remercie sincèrement. Tu as été un phare constant dans ma vie et je ne l'oublierai jamais. Sache qu'à ton tour, quoiqu'il arrive, tu pourras toujours compter sur moi.

J'étais sûr en les commençant que ces remerciements ne tiendraient pas sur une page. Ce n'est pas grave, je suis content d'avoir pris le temps d'avoir une pensée pour vous tous. Peu importe où la vie me mènera, je sais que ne suis pas seul, et ça c'est quelque chose de très précieux. Désormais, il est l'heure pour moi de prendre un repos que je pense dûment mérité 😊.

Valentin Fer, le 14 février 2023

Appendix A

AD code of L^2 scalar product

```
1  subroutine scalProdL2(n, vec1, vec2, scalProd)
2    implicit none
3  ! > INPUT variables
4    integer                 :: n
5    real*8, dimension(n) :: vec1, vec2
6    real*8, dimension(n) :: scalProd
7  ! > LOCAL variables
8    integer                 :: i
9
10   scalProd = 0.d0
11   do i=1,n
12     scalProd = scalProd + vec1(i)*vec2(i)
13   end do
14 end subroutine scalProdL2
```

Fortran code A.1: Implementation of L^2 scalar product.

```
1  subroutine scalprodL2_d(n, vec1, vec1d, vec2, vec2d, scalprod, scalprodd)
2    implicit none
3  ! > INPUT variables
4    integer, :: n
5    real*8, dimension(n) :: vec1, vec2
6    real*8, dimension(n) :: vec1d, vec2d
7    real*8                  :: scalprod
8    real*8                      :: scalprodd
9  ! > LOCAL variables
10   integer                 :: i
11
12   scalprod = 0.d0
13   scalprodd = 0.d0
14   do i=1,n
15     scalprodd = scalprodd + vec2(i)*vec1d(i) + vec1(i)*vec2d(i)
16     scalprod = scalprod + vec1(i)*vec2(i)
17   end do
18 end subroutine scalprodL2_d
```

Fortran code A.2: Forward differentiation of L^2 scalar product.

```
subroutine scalprodL2_b(n, vec1, vec1b, vec2, vec2b, scalprod, scalprodb)
  implicit none
! > INPUT variables
  integer,     :: n
  real*8, dimension(n) :: vec1, vec2
  real*8, dimension(n) :: vec1b, vec2b
  real*8,              :: scalprod
  real*8,              :: scalprodb
! > LOCAL variables
  integer              :: i

  vec1b = 0.d0
  vec2b = 0.d0
  do i=1,n
    vec1b(i) = vec1b(i) + vec2(i)*scalprodb
    vec2b(i) = vec2b(i) + vec1(i)*scalprodb
  end do
  scalprodb = 0.d0
end subroutine scalprodL2_b
```

Fortran code A.3: Adjoint differentiation of L^2 scalar product.

Appendix B

AD code of primitive computation

```
1  subroutine computeprimitive(ncell, w, p, gamma)
2    implicit none
3  ! > INPUT variables
4    integer                       :: ncell
5    real*8, dimension(ncell, 5)   :: w
6    real*8, dimension(ncell, 5)   :: p
7    real*8                        :: gamma
8  ! > LOCAL variables
9    integer                       :: icell
10
11   do icell=1,ncell
12     p(icell, 1) = w(icell, 1)
13     p(icell, 2) = w(icell, 2)/w(icell, 1)
14     p(icell, 3) = w(icell, 3)/w(icell, 1)
15     p(icell, 4) = w(icell, 4)/w(icell, 1)
16     p(icell, 5) = (gamma-1.)*(w(icell, 5)-0.5/w(icell, 1)*(w(icell, 2)*w&
17 &      (icell, 2)+w(icell, 3)*w(icell, 3)+w(icell, 4)*w(icell, 4)))
18   end do
19 end subroutine computeprimitive
```

Fortran code B.1: Primitive routine.

```
1  subroutine computeprimitive_d(ncell, w, wd, p, pd, gamma)
2    implicit none
3  ! > INPUT variables
4    integer                       :: ncell
5    real*8, dimension(ncell, 5)   :: w
6    real*8, dimension(ncell, 5)   :: wd
7    real*8, dimension(ncell, 5)   :: p
8    real*8, dimension(ncell, 5)   :: pd
9    real*8                        :: gamma
10 ! > LOCAL variables
11   integer                       :: icell
12   real*8                        :: temp
13
14   pd = 0.d0
15   do icell=1,ncell
16     pd(icell, 1) = wd(icell, 1)
17     p(icell, 1) = w(icell, 1)
18     temp = w(icell, 2)/w(icell, 1)
```

```
19      pd(icell, 2) = (wd(icell, 2)-temp*wd(icell, 1))/w(icell, 1)
20      p(icell, 2) = temp
21      temp = w(icell, 3)/w(icell, 1)
22      pd(icell, 3) = (wd(icell, 3)-temp*wd(icell, 1))/w(icell, 1)
23      p(icell, 3) = temp
24      temp = w(icell, 4)/w(icell, 1)
25      pd(icell, 4) = (wd(icell, 4)-temp*wd(icell, 1))/w(icell, 1)
26      p(icell, 4) = temp
27      temp = (w(icell, 2)*w(icell, 2)+w(icell, 3)*w(icell, 3)+w(icell, 4)*&
28     &    w(icell, 4))/w(icell, 1)
29      pd(icell, 5) = (gamma-1.)*(wd(icell, 5)-0.5*(2*w(icell, 2)*wd(icell&
30     &    , 2)+2*w(icell, 3)*wd(icell, 3)+2*w(icell, 4)*wd(icell, 4)-temp*wd&
31     &    (icell, 1))/w(icell, 1))
32      p(icell, 5) = (gamma-1.)*(w(icell, 5)-0.5*temp)
33    end do
34  end subroutine computeprimitive_d
```

Fortran code B.2: Forward differentiation of primitive routine.

```
1   subroutine computeprimitive_d_kform(ncell, w, wdr, wdi, p, pdr, pdi, gamma)
2     implicit none
3   ! > INPUT variables
4     integer                         :: ncell
5     real*8, dimension(ncell, 5) :: w
6     real*8, dimension(ncell, 5) :: wdr
7     real*8, dimension(ncell, 5) :: wdi
8     real*8, dimension(ncell, 5) :: p
9     real*8, dimension(ncell, 5) :: pdr
10    real*8, dimension(ncell, 5) :: pdi
11    real*8                          :: gamma
12  ! > LOCAL variables
13    integer                         :: icell
14    real*8                          :: temp
15
16    pdr = 0.d0
17    pdi = 0.d0
18    do icell=1,ncell
19      pdr(icell, 1) = wdr(icell, 1)
20      pdi(icell, 1) = wdi(icell, 1)
21      p(icell, 1) = w(icell, 1)
22      temp = w(icell, 2)/w(icell, 1)
23      pdr(icell, 2) = (wdr(icell, 2)-temp*wdr(icell, 1))/w(icell, 1)
24      pdi(icell, 2) = (wdi(icell, 2)-temp*wdi(icell, 1))/w(icell, 1)
25      p(icell, 2) = temp
26      temp = w(icell, 3)/w(icell, 1)
27      pdr(icell, 3) = (wdr(icell, 3)-temp*wdr(icell, 1))/w(icell, 1)
28      pdi(icell, 3) = (wdi(icell, 3)-temp*wdi(icell, 1))/w(icell, 1)
29      p(icell, 3) = temp
30      temp = w(icell, 4)/w(icell, 1)
31      pdr(icell, 4) = (wdr(icell, 4)-temp*wdr(icell, 1))/w(icell, 1)
32      pdi(icell, 4) = (wdi(icell, 4)-temp*wdi(icell, 1))/w(icell, 1)
33      p(icell, 4) = temp
34      temp = (w(icell, 2)*w(icell, 2)+w(icell, 3)*w(icell, 3)+w(icell, 4)*&
35     &    w(icell, 4))/w(icell, 1)
36      pdr(icell, 5) = (gamma-1.)*(wdr(icell, 5)-0.5*(2*w(icell, 2)*wdr(icell&
37     &    , 2)+2*w(icell, 3)*wdr(icell, 3)+2*w(icell, 4)*wdr(icell, 4)-temp*wdr&
```

```
38 &      (icell, 1))/w(icell, 1))
39       pdi(icell, 5) = (gamma-1.)*(wdi(icell, 5)-0.5*(2*w(icell, 2)*wdi(icell&
40 &      , 2)+2*w(icell, 3)*wdi(icell, 3)+2*w(icell, 4)*wdi(icell, 4)-temp*wdi&
41 &      (icell, 1))/w(icell, 1))
42       p(icell, 5) = (gamma-1.)*(w(icell, 5)-0.5*temp)
43     end do
44 end subroutine computeprimitive_d_kform
```

Fortran code B.3: K-formulated forward differentiation of primitive routine.

```
1  subroutine computeprimitive_b(ncell, w, wb, p, pb, gamma)
2    implicit none
3  ! > INPUT variables
4    integer :: ncell
5    real*8, dimension(ncell, 5) :: w
6    real*8, dimension(ncell, 5) :: wb
7    real*8, dimension(ncell, 5) :: p
8    real*8, dimension(ncell, 5) :: pb
9    real*8                      :: gamma
10 ! > LOCAL variables
11   integer                     :: icell
12   real*8                      :: tempb
13
14   wb = 0.d0
15   do icell=ncell,1,-1
16     wb(icell, 5) = wb(icell, 5) + (gamma-1.)*pb(icell, 5)
17     tempb = -(0.5*(gamma-1.)*pb(icell, 5)/w(icell, 1))
18     wb(icell, 2) = wb(icell, 2) + 2*w(icell, 2)*tempb
19     wb(icell, 3) = wb(icell, 3) + 2*w(icell, 3)*tempb
20     wb(icell, 4) = wb(icell, 4) + 2*w(icell, 4)*tempb
21     wb(icell, 1) = wb(icell, 1) - (w(icell, 2)**2+w(icell, 3)**2+w(icell&
22 &      , 4)**2)*tempb/w(icell, 1)
23     tempb = pb(icell, 4)/w(icell, 1)
24     wb(icell, 4) = wb(icell, 4) + tempb
25     wb(icell, 1) = wb(icell, 1) - w(icell, 4)*tempb/w(icell, 1)
26     tempb = pb(icell, 3)/w(icell, 1)
27     wb(icell, 3) = wb(icell, 3) + tempb
28     wb(icell, 1) = wb(icell, 1) - w(icell, 3)*tempb/w(icell, 1)
29     tempb = pb(icell, 2)/w(icell, 1)
30     wb(icell, 2) = wb(icell, 2) + tempb
31     wb(icell, 1) = wb(icell, 1) + pb(icell, 1) - w(icell, 2)*tempb/w(&
32 &      icell, 1)
33   end do
34 end subroutine computeprimitive_b
```

Fortran code B.4: Adjoint differentiation of primitive routine.

```
1  subroutine computeprimitive_b_kform(ncell, w, wb, p, pb, gamma)
2    implicit none
3  ! > INPUT variables
4    integer :: ncell
5    real*8, dimension(ncell, 5) :: w
6    real*8, dimension(ncell, 5) :: wbr
7    real*8, dimension(ncell, 5) :: wbi
8    real*8, dimension(ncell, 5) :: p
```

```
9   real*8, dimension(ncell, 5) :: pbr
10  real*8, dimension(ncell, 5) :: pbi
11  real*8                      :: gamma
12 ! > LOCAL variables
13  integer                     :: icell
14  real*8                      :: tempbr
15  real*8                      :: tempbi
16
17  wbr = 0.d0
18  wbi = 0.d0
19  do icell=ncell,1,-1
20    wbr(icell, 5) = wbr(icell, 5) + (gamma-1.)*pbr(icell, 5)
21    wbi(icell, 5) = wbi(icell, 5) + (gamma-1.)*pbi(icell, 5)
22    tempbr = -(0.5*(gamma-1.)*pbr(icell, 5)/w(icell, 1))
23    tempbi = -(0.5*(gamma-1.)*pbi(icell, 5)/w(icell, 1))
24    wbr(icell, 2) = wbr(icell, 2) + 2*w(icell, 2)*tempbr
25    wbi(icell, 2) = wbi(icell, 2) + 2*w(icell, 2)*tempbi
26    wbr(icell, 3) = wbr(icell, 3) + 2*w(icell, 3)*tempbr
27    wbi(icell, 3) = wbi(icell, 3) + 2*w(icell, 3)*tempbi
28    wbr(icell, 4) = wbr(icell, 4) + 2*w(icell, 4)*tempbr
29    wbi(icell, 4) = wbi(icell, 4) + 2*w(icell, 4)*tempbi
30    wbr(icell, 1) = wbr(icell, 1) - (w(icell, 2)**2+w(icell, 3)**2+w(icell&
31 &     , 4)**2)*tempbr/w(icell, 1)
32    wbi(icell, 1) = wbi(icell, 1) - (w(icell, 2)**2+w(icell, 3)**2+w(icell&
33 &     , 4)**2)*tempbi/w(icell, 1)
34    tempbr = pbr(icell, 4)/w(icell, 1)
35    tempbi = pbi(icell, 4)/w(icell, 1)
36    wbr(icell, 4) = wbr(icell, 4) + tempbr
37    wbi(icell, 4) = wbi(icell, 4) + tempbi
38    wbr(icell, 1) = wbr(icell, 1) - w(icell, 4)*tempbr/w(icell, 1)
39    wbi(icell, 1) = wbi(icell, 1) - w(icell, 4)*tempbi/w(icell, 1)
40    tempbr = pbr(icell, 3)/w(icell, 1)
41    tempbi = pbi(icell, 3)/w(icell, 1)
42    wbr(icell, 3) = wbr(icell, 3) + tempbr
43    wbi(icell, 3) = wbi(icell, 3) + tempbi
44    wbr(icell, 1) = wbr(icell, 1) - w(icell, 3)*tempbr/w(icell, 1)
45    wbi(icell, 1) = wbi(icell, 1) - w(icell, 3)*tempbi/w(icell, 1)
46    tempbr = pbr(icell, 2)/w(icell, 1)
47    tempbi = pbi(icell, 2)/w(icell, 1)
48    wbr(icell, 2) = wbr(icell, 2) + tempbr
49    wbi(icell, 2) = wbi(icell, 2) + tempbi
50    wbr(icell, 1) = wbr(icell, 1) + pbr(icell, 1) - w(icell, 2)*tempbr/w(&
51 &     icell, 1)
52    wbi(icell, 1) = wbi(icell, 1) + pbi(icell, 1) - w(icell, 2)*tempbi/w(&
53 &     icell, 1)
54  end do
55 end subroutine computeprimitive_b_kform
```

Fortran code B.5: K-formulated adjoint differentiation of primitive routine.

Abstract

In specific physical conditions, rotating stall is an unsteady phenomenon that may be triggered in axial compressors of turbomachinery and could lead to drastic loads on the structures. Nowadays, its understanding remains partial and could be enhanced by performing a global stability analysis. After finding a fixed point of the governing equations of the flow, it consists of solving an eigenvalue problem where large sparse complex-valued linear systems need to be inverted. A usual strategy involves direct factorizations that, for systems exceeding 10×10^6 degrees-of-freedom, lead to memory overflow. To overcome such a limit, the use of preconditioned iterative algorithms becomes mandatory. Furthermore, as complex arithmetic loops can't be vectorized, the complex systems need to be rewritten to preserve HPC optimizations, for which important work has been carried out recently at ONERA. In this thesis, a method to scale up the global stability problems to industrial cases has been developed: the KFGSM, for K-formulated Global Stability Analysis Method. The linearized matrix-vector products used to solve the eigenvalue problem are obtained by means of Algorithmic Differentiation, which ensures the preservation of loop optimizations. The complex problems have also been reformulated in the real-equivalent K formulation that preserves the sparsity patterns of the considered operators, allowing the reuse of the most advanced preconditioning techniques of compressible CFD. The reformulated eigenvalue problem is solved by means of a Krylov-Schur algorithm where inversions are managed by the GMRES algorithm. The KFGSM is validated on three test cases with gradually increasing difficulty.

Keywords: Global Stability; Fixed point; Krylov Methods; Algorithmic Differentiation; K Formulation; Sensitivity

Résumé

Dans des conditions physiques spécifiques, le décollement tournant est un phénomène instationnaire pouvant se produire dans les compresseurs axiaux de turbomachines. Même de nos jours, la compréhension de ce type d'évènement potentiellement destructeur demeure incomplète mais pourrait être améliorée par une étude de stabilité globale. Après avoir calculé un point fixe des équations de conservation gouvernant l'écoulement, cette technique consiste à résoudre un problème aux valeurs propres dont l'inversion nécessite de nombreuses résolutions de grands systèmes linéaires creux à valeurs complexes. La stratégie usuelle est de recourir à des méthodes d'inversions directes de type factorisation LU. Ce genre de pratique devient néanmoins inutilisable sur des cas dépassant les 10×10^6 degrés de liberté du fait des limitations mémoires. Cette difficulté peut être outrepassée en utilisant des méthodes itératives de type Krylov préconditionnées. Par ailleurs, l'arithmétique complexe n'étant pas vectorisable, les sytèmes linéaires complexes impliqués dans les problèmes de stabilité globale doivent être reformulés dans une version réelle équivalente, ceci afin de pouvoir bénéficier des optimisations HPC récemment menées à l'ONERA. Dans cette thèse, une méthode permettant le passage à l'échelle d'outils de stabilité globale a été développée, la KFGSM pour K-Formulated Global Stability Method. Le produit matrice-vecteur utilisé pour résoudre le problème aux valeurs propres est obtenu par différentiation algorithmique qui préserve l'optimisation des pattern de boucle. Les systèmes complexes impliqués dans les problèmes de stabilité globale ont été réécrits en K formulation, qui préserve la sparsité des opérateurs et permet la réutilisation des routines de préconditionnement les plus avancées en CFD compressible. Les problème aux valeurs propres en K formulation sont résolus grâce à l'algorithme de Krylov-Schur dans lequel les inversions sont réalisées à l'aide d'un GMRES préconditionné. La KFGSM est ensuite validée sur des cas tests à difficultés croissantes.

***Mots-clés** : Stabilité Globale; Point Fixe; Méthodes de Krylov; Différentiation Algorithmique; Formulation K; Sensibilité*