Time-stepping and Krylov methods for large-scale instability problems

J.-Ch. Loiseau, M. A. Bucci, S. Cherubini, and J.-Ch. Robinet

Abstract With the ever increasing computational power available and the development of high-performances computing, investigating the properties of realistic very large-scale nonlinear dynamical systems has been become reachable. It must be noted however that the memory capabilities of computers increase at a slower rate than their computational capabilities. Consequently, the traditional matrix-forming approaches wherein the Jacobian matrix of the system considered is explicitly assembled become rapidly intractable. Over the past two decades, so-called *matrix-free* approaches have emerged as an efficient alternative. The aim of this chapter is thus to provide an overview of well-grounded matrix-free methods for fixed points computations and linear stability analyses of very large-scale nonlinear dynamical systems.

1 Introduction

Simulation of very large-scale linear or non-linear systems is a critical issue in many scientific fields. Fluid dynamics is full of examples where accurate and efficient methods having a reasonable computational cost and memory footprint are required.

J.-Ch. Loiseau
Laboratoire DynFluid, Arts et Métiers ParisTech, 151 boulevard de l'hôpital, 75013 Paris, France.
e-mail: jean-christophe.loiseau@ensam.eu

M. A. Bucci
Laboratoire DynFluid, Arts et Métiers ParisTech, 151 boulevard de l'hôpital, 75013 Paris, France.
e-mail: michele.bucci@ensam.eu

S. Cherubini
DMMM, Politecnico di Bari, via Re David 200, 70100 Bari, Italy. e-mail: s.cherubini@gmail.com

J.-Ch. Robinet
Laboratoire DynFluid, Arts et Métiers ParisTech, 151 boulevard de l'hôpital, 75013 Paris, France.
e-mail: jean-christophe.robinet@ensam.eu

The study of flow stability is no exception, especially when one is interested in flows where the degree of spatial inhomogeneity is more and more important (one, two or three inhomogeneous directions). Historically, hydrodynamic stability analysis has always evolved according to the progress of computers, but also with the development of increasingly efficient numerical methods.

Before the 1980s, only problems where the flow has a single inhomogeneous spatial direction (generally perpendicular to the advection direction) could be discussed. The first discretization methods used were naturally the spectral or spectral collocations methods [61, 60] which offer a reasonable trade-off between computational cost and resolvability. One of the earliest examples of using such a method for linear stability analysis purposes can be found in [41]. Approximately at the same time, computation of the eigenvalue spectrum for a 1D flow were carried out [42, 31, 52], most often by shooting methods or Newton method coupled with continuation methods [51, 20]. One needs to wait until the mid-70s before eigenvalue solvers based on QR or QZ decompositions [11, 12, 53] start to be used in the study of a broad class of flows [58, 30, 54, 53]. With the increased computational power, the 1980s and especially the 90s are marked by the rapid development of these methods for flows of increasing complexity. Various libraries are developed, the most famous ones being LAPACK [3], MKL [1] and ARPACK [64]. These libraries incorporate many iterative algorithms allowing for the full or partial computation of the eigenspectrum for flows with two inhomogeneous spatial directions, see [77, 78] for a review.

Most of the work carried out during this period consisted of linearizing the governing equations, discretizing them using methods such as spectral methods, finite-differences or finite elements and eventually solving the resulting eigenproblem often with an Arnoldi algorithm [5, 57, 64]. The constant increase in geometric complexity of the flows addressed eventually led to a reformulation of the stability analyses and to the integration of these methods into existing simulation codes (e.g. FreeFem++ [34], Nek5000 [26] or Nektar++ [43]). This evolution led to the increase importance of the numerical part (which was initially of theoretical nature). A glaring example of the weight of the numerics and resolution methods for very large-scale nonlinear dynamical systems can be illustrated in the computation of base flows, fixed point of the governing equations, which, unlike parallel and geometrically simple flows, can no longer be analytically obtained or simply approximated. Accurate computation of these equilibrium solutions is thus necessary. Fixed points solvers such as the selective frequency damping [2], Newton [59] and quasi-Newton [76], or more recently RPM (Recursive Projection Method) [71] and Boostconv [18] are now commonly used to compute these equilibrium solutions.

Regarding the computation of the eigenpairs of the linearized Navier-Stokes operator, different strategies have been proposed over the years. When one tries to compute the stability of a fully three-dimensional flow, the computation and the manipulation of the Jacobian operator is a key problem mainly related to its dimension, of the order 10^6-10^8. In the literature, two major approaches have emerged. The first one, known as "matrix-forming", explicitly assembles the Jacobian matrix The advantage of such an approach is that it is simple to compute the adjoint oper-

ator, which in this case is the hermitian of the discrete Jacobian matrix. However, this approach currently runs into computational difficulties for three-dimensional flows. Indeed, eigenvalue solvers typically require the computation of the inverse of the Jacobian, whose computational cost becomes almost out of reach. In the second approach, called "matrix-free", the Jacobian matrix is not explicitly assembled. Instead, one only needs to be able to evaluate the matrix-vector product so as to generate a Krylov sequence from which the spectral properties of the Jacobian are approximated. This method has the advantage of making stability analyses of very large-scale problems doable. One of its major drawbacks however is that one needs to write the continuous adjoint equations if interested into receptivity, sensitivity or non-modal stability problems.

The aim of this chapter is to take the point of view of the latter approach and to describe the main principles for both modal and non-modal analyses within a matrix-free and time-stepper computational framework. In that aspect, it follows the works of [79] and [21]. The different algorithms enabling the computation of the fixed points and the analysis of their modal and non-modal stability properties will be presented in detail. Advantages and limitations of each method will also be presented and illustrated by simple examples. The second objective is to give the reader a guide on how to use the different methods in order to implement them into an existing CFD code. For that purpose, the chapter is organized as follows: first, the theoretical frameworks of fixed points computation and modal and non-modal stability analyses are presented. The other sections present the different algorithms one needs to use for such analyses, taking care to compare their performances and to illustrate them on representative cases. Finally, the chapter ends with a conclusion and perspectives highlighting the most recent evolution of these methods and their possible extensions to more complex dynamics, especially to very large-scale time-periodic nonlinear dynamical systems.

2 Theoretical framework

Our attention is focused on the characterization of very high-dimensional nonlinear dynamical systems typically arising from the spatial discretization of partial differential equations such as the incompressible Navier-Stokes equations. In general, the resulting dynamical equations are written down as a system of firt order differential equations

$$\dot{X}_j = \mathcal{F}_j\left(\{X_i(t);\ i = 1,\cdots,n\},t\right)$$

where the integer n is the *dimension* of the system, and $\dot{X}_j$ denotes the time-derivative of X_j. Using the notation $\mathbf{X}$ and $\boldsymbol{\mathcal{F}}$ for the sets $\left\{X_j,\ i = 1,\cdots,n\right\}$ and $\left\{\mathcal{F}_j,\ i = 1,\cdots,n\right\}$, this system can be compactly written as

$$\dot{\mathbf{X}} = \boldsymbol{\mathcal{F}}(\mathbf{X},t), \tag{1}$$

where $\mathbf{X}$ is the $n \times 1$ *state vector* of the system and t is a continuous variable denoting time. Alternatively, accounting also for temporal discretization gives rise to a discrete-time dynamical system

$$X_{j,k+1} = \mathcal{G}_j\left(\left\{X_{i,k};\ i = 1, \cdots, n\right\}, k\right)$$

or formally

$$\mathbf{X}_{k+1} = \boldsymbol{\mathcal{G}}\left(\mathbf{X}_k, k\right) \tag{2}$$

where the index k now denotes the discrete time variable. If one uses first-order Euler extrapolation for the time discretization, the relation between $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ is given by

$$\boldsymbol{\mathcal{G}}(\mathbf{X}) = \mathbf{X} + \Delta t \boldsymbol{\mathcal{F}}(\mathbf{X}),$$

where Δt is the time-step and the explicit dependences on t and k have been dropped for the sake of simplicity.

In the rest of this section, the reader will be introduced to the concepts of fixed points and linear stability, two concepts required to characterize a number of properties of the system investigated. Particular attention will be paid to *modal* and *nonmodal stability*, two approaches that have become increasingly popular in fluid dynamics over the past decades. Note that the concept of *nonlinear optimal perturbation*, which has raised a lot attention lately, is beyond the scope of the present contribution. For interested readers, please refer to the recent work by [44] and references therein.

Finally, while we will mostly use the continuous-time representation (1) when introducing the reader to the theoretical concepts exposed in this section, using the discrete-time representation (2) will prove more useful when discussing and implementing the different algorithms presented in §3.

2.1 Fixed points

Nonlinear dynamical systems described by Eq. (1) or Eq. (2) tend to admit a number of different equilibria forming the backbone of their phase space. These different equilibria can take the form of fixed points, periodic orbits, torus or strange attractors for instance. In the rest of this work, our attention will be solely focused on fixed points.

For a continuous-time dynamical system described by Eq. (1), fixed points $\mathbf{X}^*$ are solution to

$$\boldsymbol{\mathcal{F}}(\mathbf{X}) = 0. \tag{3}$$

Conversely, fixed points $\mathbf{X}^*$ of a discrete-time dynamical system described by Eq. (2) are solution to

$$\boldsymbol{\mathcal{G}}(\mathbf{X}) = \mathbf{X}. \tag{4}$$

It must be emphasized that both Eq. (3) and Eq. (4) may admit multiple solutions. Such a multiplicity of fixed points can easily be illustrated by a dynamical system as simple as the following Duffing oscillator

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= -\frac{1}{2}y + x - x^3. \end{aligned} \tag{5}$$

Despite its apparent simplicity, this Duffing oscillator admits three fixed points, namely

- a saddle at the origin $\mathbf{X}^* = (0,0)$,
- two linearly stable spirals located at $\mathbf{X}^* = (\pm 1, 0)$.

All of these fixed points, along with some trajectories, are depicted on figure 1 for the sake of illustration. Such a multiplicity of fixed points also occurs in dynamical systems as complex as the Navier-Stokes equations. Determining which of these fixed points is the most relevant one from a physical point of view is problem-dependent and left for the user to decide. Note however that computing these equilibrium points is a prerequisite to all of the analyses to be described in this chapter. Numerical methods to solve Eq. (3) or Eq. (4) will be discussed in §3.1.

2.2 Linear stability analysis

Having computed a given fixed point $\mathbf{X}^*$ of a continuous-time nonlinear dynamical system given by Eq. (1), one may ask whether it corresponds to a stable or unstable equilibrium of the system. Before pursuing, the very notion of *stability* needs to be explained. It is traditionally defined following the concept of Lyapunov stability. Having computed the equilibrium state $\mathbf{X}^*$, the system is perturbed around this state. If it returns back to the equilibrium point, the latter is deemed stable, otherwise, it is regarded as unstable. It has to be noted that, in the concept of Lyapunov stability, an infinite time horizon is allowed for the return to equilibrium.

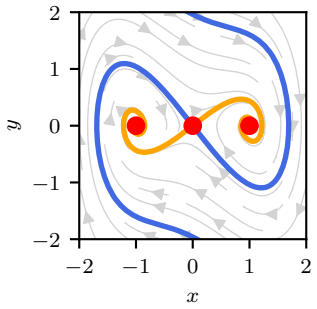

Fig. 1 Phase portrait of the unforced Duffing oscillator (5). The red dots denote the three fixed points admitted by the system. The blue (resp. orange) thick line depicts the stable (resp. unstable) manifold of the saddle point located at the origin. Grey lines highlight a few trajectories exhibited for different initial conditions.

The dynamics of a perturbation $\mathbf{x} = \mathbf{X} - \mathbf{X}^*$ are governed by

$$\dot{\mathbf{x}} = \boldsymbol{\mathcal{F}}(\mathbf{X}^* + \mathbf{x}). \tag{6}$$

Assuming the perturbation $\mathbf{x}$ is infinitesimal, $\boldsymbol{\mathcal{F}}(\mathbf{X})$ can be approximated by its first-order Taylor expansion around $\mathbf{X} = \mathbf{X}^*$. Doing so, the governing equations for the perturbation $\mathbf{x}$ simplify to

$$\dot{\mathbf{x}} = \boldsymbol{\mathcal{A}}\mathbf{x}, \tag{7}$$

where $\boldsymbol{\mathcal{A}} = \partial\boldsymbol{\mathcal{F}}/\partial\mathbf{X}$ is the $n \times n$ Jacobian matrix of $\boldsymbol{\mathcal{F}}$. Starting from an initial condition $\mathbf{x}_0$, the perturbation at time t is given by

$$\mathbf{x}(t) = \exp(\boldsymbol{\mathcal{A}}t)\,\mathbf{x}_0. \tag{8}$$

The operator $\boldsymbol{\mathcal{M}}(t) = \exp(\boldsymbol{\mathcal{A}}t)$ is known as the *exponential propagator*. Introducing the spectral decomposition of $\boldsymbol{\mathcal{A}}$

$$\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{V}}\boldsymbol{\Lambda}\boldsymbol{\mathcal{V}}^{-1},$$

Eq. (8) can be rewritten as

$$\mathbf{x}(t) = \boldsymbol{\mathcal{V}}\exp(\boldsymbol{\Lambda}t)\,\boldsymbol{\mathcal{V}}^{-1}\mathbf{x}_0, \tag{9}$$

where the i$^{\text{th}}$ column of $\boldsymbol{\mathcal{V}}$ is the eigenvector $\mathbf{v}_i$ associated to the i$^{\text{th}}$ eigenvalue $\lambda_i = \boldsymbol{\Lambda}_{ii}$, with $\boldsymbol{\Lambda}$ a diagonal matrix. Assuming that the eigenvalues of $\boldsymbol{\mathcal{A}}$ have been sorted by decreasing real part, it can easily be shown that

$$\lim_{t\to+\infty} \exp(\boldsymbol{\mathcal{A}}t)\,\mathbf{x}_0 = \lim_{t\to+\infty} \exp(\lambda_1 t)\,\mathbf{v}_1.$$

The asymptotic fate of an initial perturbation $\mathbf{x}_0$ is thus entirely dictated by the real part of the leading eigenvalue λ_1:

- if $\Re(\lambda_1) > 0$, a random initial perturbation $\mathbf{x}_0$ will eventually grow exponentially rapidly. Hence, the fixed point $\mathbf{X}^*$ is deemed *linearly unstable*.
- If $\Re(\lambda_1) < 0$, the initial perturbation $\mathbf{x}_0$ will eventually decay exponentially rapidly. The fixed point $\mathbf{X}^*$ is thus *linearly stable*.

The case $\Re(\lambda_1) = 0$ is peculiar. The fixed point $\mathbf{X}^*$ is called *elliptic* and one cannot conclude about its stability solely by looking at the eigenvalues of $\boldsymbol{\mathcal{A}}$. In this case, one needs to resort to *weakly non-linear analysis* which essentially looks at the properties of higher-order Taylor expansion of $\boldsymbol{\mathcal{F}}(\mathbf{X})$. Once again, this is beyond the scope of the present chapter. Interested readers are referred to [74] for more details about such analyses.

Illustration

Let us illustrate the notion of linear stability on a simple example. For that purpose, we will consider the same linear dynamical system as in [69]. This system reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{100} - \frac{1}{Re} & 0 \\ 1 & -\frac{2}{Re} \end{bmatrix}}_{\boldsymbol{\mathcal{A}}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{10}$$

where Re is a control parameter. For such a simple case, it is obvious that the eigenvalues of $\boldsymbol{\mathcal{A}}$ are given by

$$\lambda_1 = \frac{1}{100} - \frac{1}{Re}$$

and

$$\lambda_2 = -\frac{2}{Re}.$$

While λ_2 is constantly negative, λ_1 is negative for $Re < 100$ and positive otherwise. Figure 2 depicts the time-evolution of $\|\mathbf{x}\|_2^2 = x_1^2 + x_2^2$ for two different values of Re. Please note that the short-time ($t < 100$) behavior of the perturbation will be discussed in §2.3. It is clear nonetheless that, for $t > 100$, the time-evolution of the perturbation can be described by an exponential function. Whether this exponential increases or decreases as a function of time is solely dictated by the sign of λ_1, negative for $Re = 50$ and positive for $Re = 100$. For $Re = 50$, the equilibrium point $\mathbf{X}^* = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ is thus stable, while it is unstable for $Re = 125$.

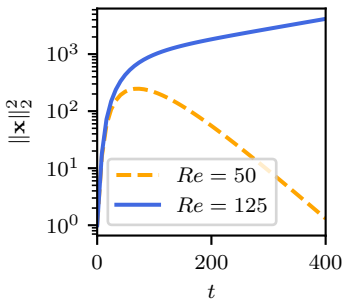

Fig. 2 Evolution as a function of time of $\|\mathbf{x}\|_2^2 = x_1^2 + x_2^2$ for the toy-model (10). For $Re = 50$ (resp. $Re = 125$), the asymptotic fate of $\|\mathbf{x}\|_2^2$ is described by a decreasing (resp. increasing) exponential. For $Re = 50$, the equilibrium point is thus linear stable, while it is linearly unstable for $Re = 125$.

2.3 Non-modal stability analysis

Looking once more at figure 2, it can be seen that, although the system is linearly stable for $Re = 50$, the perturbation $\mathbf{x}$ can experience a transient growth of its energy for a short period of time, roughly given by $0 < t < 100$ in the present case, before its eventual exponential decay. This behavior is related to the *non-normality* of $\boldsymbol{\mathcal{A}}$, i.e.

$$\boldsymbol{\mathcal{A}}^\dagger \boldsymbol{\mathcal{A}} \neq \boldsymbol{\mathcal{A}} \boldsymbol{\mathcal{A}}^\dagger, \tag{11}$$

where $\boldsymbol{\mathcal{A}}^\dagger$ is the *adjoint* of $\boldsymbol{\mathcal{A}}$. As a result of this non-normality, the eigenvectors of $\boldsymbol{\mathcal{A}}$ do not form an orthonormal set of vectors[1]. The consequences of this non-orthogonality of the set of eigenvectors can be visualized on figure 3 where the trajectory stemming from a random unit-norm initial condition $\mathbf{x}_0$ is depicted in the phase plane of our toy-model (10). The perturbation $\mathbf{x}(t)$ is first attracted toward the linear manifold associated to the least stable eigenvalue λ_1, causing in the process the transient growth of its energy by a factor 300. Once it reaches the vicinity of the linearly stable manifold, the perturbation eventually decays exponentially rapidly along this eigendirection of the fixed point. The next sections are devoted to the introduction of mathematical tools particularly useful to characterize phenomena resulting from this non-normality of $\boldsymbol{\mathcal{A}}$, both in the time and frequency domains, when the fixed point considered is stable.

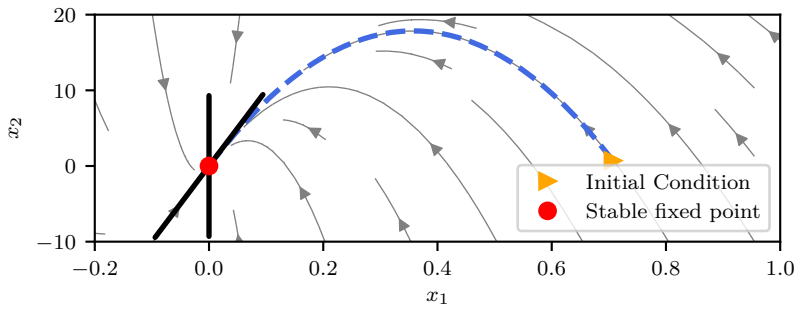

Fig. 3 The blue (dashed) line shows the trajectory stemming from a random unit-norm initial condition $\mathbf{x}_0$. The thick black lines depict the two linear manifolds of the fixed point. The diagonal one corresponds to $\lambda_1 = 1/100 - 1/Re$, while the vertical one is associated to $\lambda_2 = -2/Re$. In the present case, Re is set to 50, thus corresponding to a situation where the fixed point is linearly stable.

[1] Note that the non-normality of $\boldsymbol{\mathcal{A}}$ also implies that its right and left eigenvectors are different. This observation may have large consequences in fluid dynamics, particularly when addressing the problems of optimal linear control and/or estimation of strongly non-parallel flows.

2.3.1 Optimal perturbation analysis

Having observed that a random initial condition can experience a relatively large transient growth of its energy over a short period of time even though the fixed point is stable, one may be interested in the worst case scenario, i.e. finding which initial condition $\mathbf{x}_0$ is amplified as much as possible before it eventually decays. Searching for such a perturbation is known as *optimal perturbation analysis* and can be addressed by two different methods:

- Optimization,
- Singular Value Decomposition (SVD).

Both approaches will be presented. Although it requires the introduction of additional mathematical concepts, the approach relying on optimization will be introduced first in §2.3.1 as it is easier to grasp. The approach relying on singular value decomposition of the exponential propagrator $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}t)$ will then be presented in §2.3.1.

Formulation as an optimization problem

The aim of optimal perturbation analysis is to find the unit-norm initial condition $\mathbf{x}_0$ that maximizes $\|\mathbf{x}(T)\|_2^2$, where T is known as the *target time*. Note that we here consider only the 2-norm of $\mathbf{x}(T)$ for the sake of simplicity, although one could formally optimize different norms, see [27, 28, 29, 23] for examples from fluid dynamics. For a given target time T, such a problem can be formulated as the following constrained maximization problem

$$\begin{aligned} &\underset{\mathbf{x}_0}{\text{maximize}}\ \mathcal{J}(\mathbf{x}_0) = \|\mathbf{x}(T)\|_2^2 \\ &\text{subject to } \dot{\mathbf{x}} - \boldsymbol{\mathcal{A}}\mathbf{x} = 0 \\ &\qquad\qquad \|\mathbf{x}_0\|_2^2 - 1 = 0, \end{aligned} \tag{12}$$

where $\mathcal{J}(\mathbf{x}_0)$ is known as the *objective function*. It must be emphasized that problem (12) is not formulated as a convex optimization problem[2]. As such, it may exhibit local maximum. Nonetheless, this constrained maximization problem can be recast into the following unconstrained maximization problem

[2] Formally, a convex optimization problem reads

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}}\ \mathcal{J}(\mathbf{x}) \\ &\text{subject to } g_i(\mathbf{x}) \leq 0,\ i = 1, \cdots, m \\ &\qquad\qquad h_i(\mathbf{x}) = 0,\ i = 1, \cdots, p, \end{aligned}$$

where the objective function $\mathcal{J}(\mathbf{x})$ and the inequality constraints functions $g_i(\mathbf{x})$ are convex. The conditions on the equality constraints functions $h_i(\mathbf{x})$ are more restrictive as they need to be affine functions, i.e. of the form $h_i(\mathbf{x}) = \mathbf{a}_i^T\mathbf{x} + b_i$. See the book by Boyd & Vandenberghe [9] for extensive details about convex optimization.

$$\underset{\mathbf{x},\mathbf{v},\mu}{\text{maximize}}\ \mathcal{L}(\mathbf{x},\mathbf{v},\mu), \tag{13}$$

where

$$\mathcal{L}(\mathbf{x},\mathbf{v},\mu) = \mathcal{J}(\mathbf{x}_0) + \int_0^T \mathbf{v}^T(\dot{\mathbf{x}} - \boldsymbol{\mathcal{A}}\mathbf{x})\,\mathrm{d}t + \mu\left(\|\mathbf{x}_0\|_2^2 - 1\right) \tag{14}$$

is known as the *augmented Lagrangian* function. The additional optimization variables $\mathbf{v}$ and μ appearing in the definition of the augmented Lagrangian $\mathcal{L}$ are called *Lagrange multipliers*. Solutions to problem (13) are identified by vanishing first variations of $\mathcal{L}$ with respect to our three optimization variables. The first variation of $\mathcal{L}$ with respect to $\mathbf{v}$ and μ are simply the constraints of our original problem (12). The first variation of $\mathcal{L}$ with respect to $\mathbf{x}$ on the other hand is given by

$$\delta_{\mathbf{x}}\mathcal{L} = [\nabla_{\mathbf{x}}\mathcal{J} + \mathbf{v}(T)]\cdot\delta\mathbf{x}(0) + \int_0^T \left[\dot{\mathbf{v}} - \boldsymbol{\mathcal{A}}^\dagger\mathbf{v}\right]\cdot\delta\mathbf{x}\,\mathrm{dt} + [2\mu\mathbf{x}_0 - \mathbf{v}(0)]\cdot\delta\mathbf{x}(0). \tag{15}$$

Eq. (15) vanishes only if

$$\dot{\mathbf{v}} = \boldsymbol{\mathcal{A}}^\dagger\mathbf{v} \ \text{ over } t\in(0,T), \tag{16}$$

and

$$\begin{aligned} \nabla_{\mathbf{x}}\mathcal{J} - \mathbf{v}(T) &= 0 \\ 2\mu\mathbf{x}_0 - \mathbf{v}(0) &= 0. \end{aligned} \tag{17}$$

Note that Eq. (16) is known as the adjoint system[3] of our original linear dynamical system, while Eq. (17) are called compatibility conditions. Maximizing $\mathcal{L}$ is then a problem of simultaneously satisfying (7), (16) and (17). This is in general done iteratively by gradient-based algorithms such as gradient ascent or the rotation-update gradient algorithm (see §3). For more details about adjoint-based optimization, see [9, 44].

Formulation using SVD

As stated previously, formulating the optimal perturbation analysis as a constrained maximization results in a non-convex optimization problem (12). Consequently, although a solution to (12) can easily be obtained by means of gradient-based algo-

[3] Given an appropriate inner product, the adjoint operator $\boldsymbol{\mathcal{A}}^\dagger$ is defined such that

$$\langle\mathbf{v}|\boldsymbol{\mathcal{A}}\mathbf{x}\rangle = \langle\boldsymbol{\mathcal{A}}^\dagger\mathbf{v}|\mathbf{x}\rangle,$$

where $\langle\mathbf{a}|\mathbf{b}\rangle$ denotes the inner product of $\mathbf{a}$ and $\mathbf{b}$. If one consider the classical Euclidean inner product, the adjoint operator is simply given by

$$\boldsymbol{\mathcal{A}}^\dagger = \boldsymbol{\mathcal{A}}^H$$

where $\boldsymbol{\mathcal{A}}^H$ is the Hermitian (i.e. complex-conjugate transpose) of $\boldsymbol{\mathcal{A}}$. It must be noted finally that the direct operator $\boldsymbol{\mathcal{A}}$ and the adjoint one $\boldsymbol{\mathcal{A}}^\dagger$ have the same eigenspectrum. This last observation is a key point when one aims at validating the numerical implementation of an adjoint solver.

rithms, one cannot rule out the possibility that this solution is only a local maximum rather than the global one. In this section, we will show that recasting problem (12) in the framework of linear algebra however allows us to obtain easily this global optimal.

Let us first redefine our optimization problem as

$$\underset{\mathbf{x}_0}{\text{maximize}} \ \frac{\|\mathbf{x}(T)\|_2^2}{\|\mathbf{x}_0\|_2^2} \tag{18}$$

so that rather than maximizing $\|\mathbf{x}(T)\|_2^2$ under the constraint that $\|\mathbf{x}_0\|_2^2 = 1$, we now directly aim to maximize the energy gain $\mathcal{G}(T) = {}^{\|\mathbf{x}(T)\|_2^2}/_{\|\mathbf{x}_0\|_2^2}$. Moreover, recalling from (8) that

$$\mathbf{x}(T) = \exp(\boldsymbol{\mathcal{A}}T)\,\mathbf{x}_0,$$

our energy gain maximization problem can finally be written as

$$\begin{aligned} \mathcal{G}(T) &= \max_{\mathbf{x}_0} \frac{\|\exp(\boldsymbol{\mathcal{A}}t)\,\mathbf{x}_0\|_2^2}{\|\mathbf{x}_0\|_2^2} \\ &= \|\exp(\boldsymbol{\mathcal{A}}T)\|_2^2 \end{aligned} \tag{19}$$

where $\|\exp(\boldsymbol{\mathcal{A}}T)\|_2$ is a vector-induced matrix norm taking care of the optimization over all possible initial conditions $\mathbf{x}_0$. Introducing singular value decomposition (SVD), i.e.

$$\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{U}}\boldsymbol{\Sigma}\boldsymbol{\mathcal{V}}^H,$$

it is relatively easy to demonstrate that the optimal energy gain $\mathcal{G}(T)$ is given by

$$\mathcal{G}(T) = \sigma_1^2, \tag{20}$$

where σ_1 is the largest singular value of the exponential propagator $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T)$. The optimal initial condition $\mathbf{x}_0$ is then given by the principal right singular vector (i.e. $\mathbf{x}_0 = \mathbf{v}_1$), while the associated response is given by $\mathbf{x}(T) = \sigma_1\mathbf{u}_1$, where $\mathbf{u}_1$ is the principal left singular vector.

Illustration

As to illustrate linear optimal perturbations, let us consider the incompressible flow of a Newtonian fluid induced by two flat plates moving in-plane in opposite directions as sketched on figure 4(a). The resulting flow, known as *plane Couette flow*, is given by

$$U(y) = y.$$

Note that it is a linearly stable fixed point of the Navier-Stokes equations no matter the Reynolds number considered. Despite its linear stability, subcritical transition to turbulence can occur for Reynolds numbers as low as $Re = 325$ [55].

Without getting too deep into the mathematical and physical details of such subcritical transition, part of the explanation can be given by linear optimal perturbation analysis. The dynamics of an infinitesimal perturbation $\mathbf{x} = \begin{bmatrix} \mathbf{v} & \eta \end{bmatrix}^T$, characterized by a certain wavenumber $\mathbf{k} = \alpha \mathbf{e}_x + \beta \mathbf{e}_z$, evolving in the vicinity of this fixed point are governed by

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \mathbf{v} \\ \eta \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{OS} & 0 \\ \mathcal{C} & \mathcal{A}_S \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \eta \end{bmatrix} \tag{21}$$

where $\mathbf{v}$ is the wall-normal velocity of the perturbation and η its wall-normal vorticity, $\mathcal{A}_{OS}$ is the Orr-Sommerfeld operator, while $\mathcal{A}_S$ is the Squire one. The operator $\mathcal{C}$ describes the existing coupling between the wall-normal velocity $\mathbf{v}$ and the wall-normal vorticity η. For certain pairs of wavenumbers, this Orr-Sommerfeld-Squire operator is highly non-normal and perturbations can exhibit very large transient growth. This is illustrated on figure 4(a) where the evolution of the optimal gain $\mathcal{G}(T)$ as a function of the target time T is depicted for different pairs of wavenumbers (α, β) at $Re = 300$. The maximum amplification achievable over all target times T and wavenumbers pairs (α, β) is $\mathcal{G}_{\mathrm{opt}} \simeq 100$. The initial perturbation $\mathbf{x}_0$ corresponding to this optimal energy gain is depicted on figure 4(b). It corresponds to streamwise-oriented vortices that eventually give rise to streamwise velocity streaks due to the lift-up effect [48, 10], see figure 4(b). While this perturbation eventually decays exponentially rapidly in a purely linear framework, it has been shown that, even for a moderately large initial amplitude, it may eventually trigger transition to turbulence when used as initial condition in a non-linear direct numerical simulation of the Navier-Stokes equations [50]. For more details about subcritical transition and extension of optimal perturbation analysis to non-linear operators, interested readers are referred to [45].

2.3.2 Resolvent analysis

The optimal perturbation analysis (see §2.3.1) aims at finding the initial condition $\mathbf{x}_0$ that maximizes the transient amplification of energy of the response $\mathbf{x}(T) = \exp(\mathcal{A}T)\mathbf{x}_0$ at the target time $t = T$. It is thus an initial-value problem that can be investigated in the time domain. Rather than considering the response of the system to different initial conditions, one may instead wonder how the system reacts to external noise. For that purpose, let us now consider a forced linear dynamical system

$$\dot{\mathbf{x}} = \mathcal{A}\mathbf{x} + \mathbf{f} \tag{22}$$

where the forcing $\mathbf{f}$ now models the system's input such as the external noise. As before, we moreover assume that all of the eigenvalues of $\mathcal{A}$ lie within the stable half of the complex plane. As for the optimal perturbation analysis, one may now consider a worst-case scenario, i.e. what is the forcing $\mathbf{f}$ that maximizes the asymptotic response of the system? Because we consider a linear dynamical system, this question can naturally be addressed in the frequency domain.

In the most general case, the response of the system to the forcing $\mathbf{f}(t)$ is given by

$$\mathbf{x}(t) = \int_0^t \exp\left(\boldsymbol{\mathcal{A}}(t-\tau)\right)\mathbf{f}(\tau)\,\mathrm{d}\tau \tag{23}$$

which is a convolution integral. Note that, in the above expression, we assumed a zero initial condition, i.e. $\mathbf{x}_0 = 0$. Such a convolution integral is also known as a memory integral and highlights that the current state $\mathbf{x}(t)$ of the system depends on the entire history of the forcing $\mathbf{f}$. Because we consider linear stable systems, the influence of the forcing on the current state decays exponentially according to the least stable eigenvalue. Let us assume furthermore a harmonic external forcing

$$\mathbf{f}(t) = \Re\left(\hat{\mathbf{f}}e^{i\omega t}\right) \tag{24}$$

where $\omega \in \mathbb{R}$ is the circular frequency of the forcing. The convolution integral can now be easily computed in the frequency domain. Given our assumptions, the asymptotic response of the system at the frequency ω is given by

$$\hat{\mathbf{x}} = \left(i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}}\right)^{-1}\hat{\mathbf{f}}. \tag{25}$$

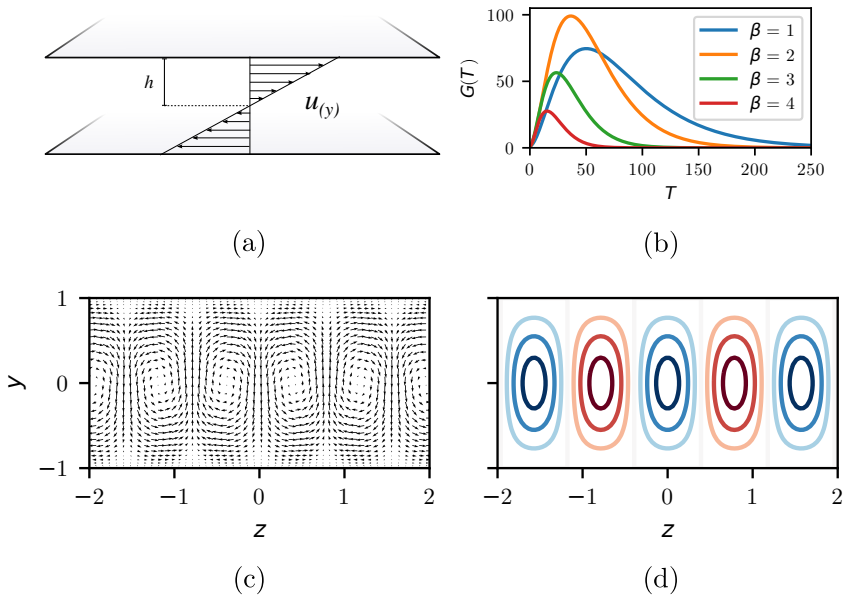

Fig. 4 Illustration of optimal perturbation analysis for the plane Couette flow at $Re = 300$. In all cases, the streamwise wavenumber of the perturbation is set to $\alpha = 0$. (a) Optimal gain curve for different spanwise wavenumbers β. (b) Optimal perturbation (left) and optimal response (right) for $\beta = 2$. Note that optimal perturbation consists of streamwise oriented vortices, while the associated response at time T consist in high- and low-speed streaks.

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{f}}$ are the Fourier transforms of $\mathbf{x}$ and $\mathbf{f}$, respectively. The operator $\boldsymbol{\mathcal{R}}(\omega) = (i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}})^{-1}$ appearing in Eq. (25) is known as the *Resolvent operator* and is related to the exponential propagator $\boldsymbol{\mathcal{M}}(t) = \exp(\boldsymbol{\mathcal{A}}t)$ via Laplace transform. This operator, acting in the frequency domain, maps the input harmonic forcing $\hat{\mathbf{f}}(\omega)$ to the output harmonic response $\hat{\mathbf{x}}(\omega)$.

Finding the forcing frequency ω that maximizes the asymptotic response $\mathbf{x}$ of the system can now be formalized as

$$\begin{aligned}\mathcal{R}(\omega) &= \max_{\hat{\mathbf{f}}} \frac{\| (i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}})^{-1}\hat{\mathbf{f}}\|_2^2}{\|\hat{\mathbf{f}}\|_2^2} \\ &= \|\boldsymbol{\mathcal{R}}(\omega)\|_2^2.\end{aligned} \tag{26}$$

Going from the time domain to the frequency domain, the norm of the exponential propagator is replaced with that of the resolvent in order to quantify the energy amplification between the input forcing and the output response. As before, the optimal resolvent gain at the frequency ω is given by

$$\mathcal{R}(\omega) = \sigma_1^2,$$

where σ_1 is the largest singular value of $\boldsymbol{\mathcal{R}}(\omega)$. The associated optimal forcing $\hat{\mathbf{f}}_{\text{opt}}$ and response $\hat{\mathbf{x}}_{\text{opt}}$ are then given by the corresponding right and left singular vectors, respectively.

Illustration

Let us illustrate resolvent analysis using the linearized complex Ginzburg-Landau equation, a typical model for instabilities in spatially-evolving flows. The equation reads

$$\frac{\partial u}{\partial t} = -\nu\frac{\partial u}{\partial x} + \gamma\frac{\partial^2 u}{\partial x^2} + \mu(x)u. \tag{27}$$

The spatial dependency of the solution result from the parameter $\mu(x)$ which is defined as

$$\mu(x) = (\mu_0 - c_\mu^2) + \frac{\mu_2}{2}x^2.$$

The same expression has been used in [36, 7, 16]. We take $\mu_0 = 0.23$ and all other parameters are set to the same values as in [7]. The resulting model is linearly stable but is susceptible to large non-modal growth. We use the same code as [16]. The problem is discretized on the interval $x \in [-85, 85]$ using 220 points with a pseudo-spectral approach based on Hermite polynomials.

Figure 5(a) depicts the evolution of the first four resolvent gains σ_j^2 as a function of the forcing frequency ω. Although the system is linearly stable for the set of parameters considered, a unit-norm harmonic forcing $\hat{\mathbf{f}}(\omega)$ can trigger a response $\hat{\mathbf{u}}(\omega)$ whose energy has been amplified by a factor almost 1000. The optimal forcing and associated response for the most amplified frequency ($\omega \simeq -0.55$) are depicted

on figure 5(b). It can be observed that their spatial support are disjoint. The optimal forcing is mostly localized in the downstream region $-20 \leq x \leq 0$, while the associated response is mostly localized in the upstream region $0 \leq x \leq 20$. This difference in the spatial support of the forcing and the response is a classical feature of highly non-normal spatially evolving flows. Such a behavior, which has been observed in a wide variety of open shear flows, has a lot of implications when it comes to flow control, see [7] for more details.

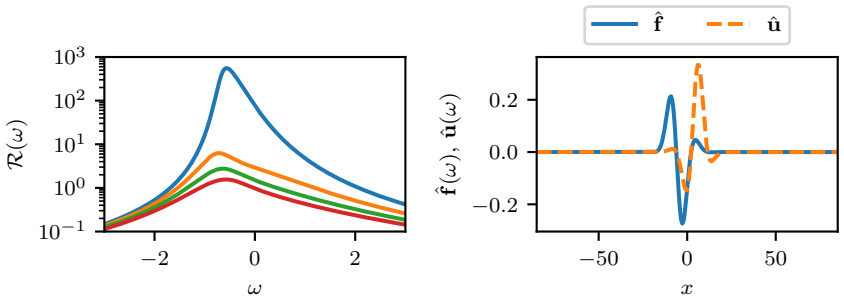

Fig. 5 (left) Evolution of the first four resolvent gains σ_j^2 as a function of the forcing frequency ω for the complex Ginzburg-Landau equation (27). (right) Optimal forcing $\hat{\mathbf{f}}(\omega)$ and associated optimal response $\hat{\mathbf{u}}(\omega)$ for the most amplified frequency. Note that only the real parts are shown.

3 Numerical methods

In this section, different techniques will be presented to solve modal and non-modal stability problems for very large-scale dynamical systems. Such very large-scale systems typically arise from the spatial discretization of partial differential equations, e.g. the Navier-Stokes equations in fluid dynamics. Throughout this section, the two-dimensional shear-driven cavity flow at various Reynolds numbers will serve as an example. The same configuration as [72] is considered. The dynamics of the flow are governed by

$$\begin{aligned} \frac{\partial \mathbf{U}}{\partial t} + (\mathbf{U}\cdot\nabla)\,\mathbf{U} &= -\nabla P + \frac{1}{Re}\nabla^2\mathbf{U} \\ \nabla\cdot\mathbf{U} &= 0, \end{aligned} \tag{28}$$

where $\mathbf{U}$ is the velocity field and P is the pressure field. Figure 6 depicts a typical vorticity snapshot obtained from direct numerical simulation at a supercritical Reynolds number.

Given a fixed point $\mathbf{U}_b$ of the Navier-Stokes equations (28), the dynamics of an infinitesimal perturbation $\mathbf{u}$ evolving on top of it are governed by

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\,\mathbf{U}_b + (\mathbf{U}_b\cdot\nabla)\,\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} \\ \nabla\cdot\mathbf{u} &= 0. \end{aligned} \tag{29}$$

Once projected onto a divergence-free vector space, Eq. (29) can be formally written as

$$\dot{\mathbf{u}} = \boldsymbol{\mathcal{A}}\mathbf{u}, \tag{30}$$

where $\boldsymbol{\mathcal{A}}$ is the linearized Navier-Stokes operator. After being discretized in space, $\boldsymbol{\mathcal{A}}$ is a $n \times n$ matrix. For our example, the computational domain is discretized using 158 400 grid points, resulting in a total of 475 200 degrees of freedom. From a practical point of view, explicitly assembling the resulting matrix $\boldsymbol{\mathcal{A}}$ would have relatively large memory footprint. Using explicitly the matrix $\boldsymbol{\mathcal{A}}$ to investigate the stability properties of this two-dimensional flow is thus hardly possible on a simple laptop at the moment despite the simplicity of the case considered. It has to be noted however that, given an initial condition $\mathbf{u}_0$, the analytical solution to Eq. (30) reads

$$\mathbf{u}(T) = \exp(\boldsymbol{\mathcal{A}}T)\,\mathbf{u}_0,$$

where $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T)$ is the exponential propagator introduced previously. Although assembling explicitly this matrix $\boldsymbol{\mathcal{M}}$ is even harder than assembling $\boldsymbol{\mathcal{A}}$, its application onto the vector $\mathbf{u}_0$ can easily be computed using a classical time-stepping code solving the linearized Navier-Stokes equations (29). Such a *time-stepper* approach has been popularized by [22, 6]. In the rest of this section, the different algorithms proposed for fixed point computation, linear stability and non-modal stability analyses will heavily rely on this time-stepper strategy. The key point is that they require only minor modifications of an existing time-stepping code to be put into use.

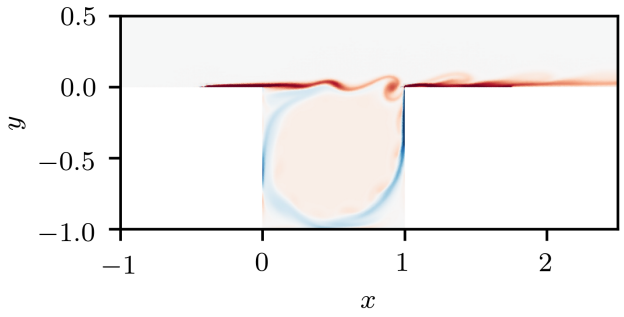

Fig. 6 Instantaneous vorticity field of the shear-driven cavity flow at $Re = 7500$ (based on the cavity's depth).

3.1 Fixed points computation

The starting point when investigating a nonlinear dynamical system it to determine its fixed points. As discussed in §2.1, for a continuous-time dynamical system, such points are solution to

$$\boldsymbol{\mathcal{F}}(\mathbf{X}) = 0, \tag{31}$$

while one needs to solve

$$\mathbf{X} - \boldsymbol{\mathcal{G}}(\mathbf{X}) = 0 \tag{32}$$

for a discrete-time nonlinear dynamical system. In this section, three different fixed point solvers will be presented.

3.1.1 Selective Frequency Damping

Selective frequency damping is a fixed point computation technique proposed by Åkervik *et al.* [2] in 2006 and largely adapted from the original work of Pruett *et al.* [62, 63] on temporal approximate deconvolution models for large-eddy simulations. It has since become one of the standard approaches for fixed point computation in fluid dynamics due to its ease of implementation. Note that various implementations of the original selective frequency damping method have been proposed over the years [39, 40, 19]. Moreover, it has since been extented to compute steady states of the Reynolds-Averaged-Navier-Stokes (RANS) equations [67] as well as for the computation of unstable periodic orbits [70]. In the rest of this section, only the original formulation by Åkervik *et al.* [2] will be described.

Let us consider a fixed point $\mathbf{X}^*$ of the nonlinear system

$$\dot{\mathbf{X}} = \boldsymbol{\mathcal{F}}(\mathbf{X}).$$

If $\mathbf{X}^*$ is linearly unstable, then any initial condition $\mathbf{X}_0 \neq \mathbf{X}^*$ will quickly depart from $\mathbf{X}^*$. Using standard regularization techniques from control theory, the aim of selective frequency damping is thus to stabilize the linearly unstable fixed point $\mathbf{X}^*$. For that purpose, one can use proportional feedback control so that the forced system now reads

$$\dot{\mathbf{X}} = \boldsymbol{\mathcal{F}}(\mathbf{X}) - \chi(\mathbf{X} - \mathbf{Y}), \tag{33}$$

where χ is the control gain and $\mathbf{Y}$ the target solution. This target solution is obviously the fixed point one aims to stabilize, i.e. $\mathbf{Y} = \mathbf{X}^*$, which is unfortunately not known *a priori*. It has to be noted however that, for a large range of situations, the instability of the fixed point $\mathbf{X}^*$ will tend to give rise to unsteady dynamics. In such cases, the target solution $\mathbf{Y}$ is thus a modification of $\mathbf{X}$ with *reduced temporal fluctuations*, i.e. a temporally low-pass filtered solution. This filtered solution is defined as

$$\mathbf{Y}(t) = \mathcal{H}(t, \Delta) * \mathbf{X}(t - \tau) \tag{34}$$

where $\mathcal{H}$ is the convolution kernel of the applied causal low-pass filter and Δ the filter witdh. Using such definitions, the forced system (33) can thus be rewritten as

$$\dot{\mathbf{X}} = \boldsymbol{\mathcal{F}}(\mathbf{X}) - \chi(\mathcal{I} - \mathcal{H}) * \mathbf{X}. \tag{35}$$

As $\mathbf{X}$ tends to the fixed point $\mathbf{X}^*$, the low-pass filtered solution $\mathbf{Y}$ tends to $\mathbf{X}$. Once a steady state has been reached, one has

$$\mathbf{X} = \mathbf{Y} = \mathbf{X}^*,$$

i.e. the fixed point of the controlled system (35) is the same as that of our original system. Moreover, as the system approaches its fixed point, the amplitude of the proportional feedback control term vanishes.

As it is formulated, computing the low-pass filtered solution (34) requires the evaluation of the following convolution integral

$$\mathbf{Y}(t) = \int_{-\infty}^{t} \mathcal{H}(\tau - t, \Delta)\mathbf{X}(\tau)\mathrm{d}\tau. \tag{36}$$

Note that, to be admissible, the kernel $\mathcal{H}$ must be positive and properly normalized. Moreover, in the limit of vanishing filter width, it must approach the Dirac delta function. To the best of our knowledge, all implementations of the selective frequency damping thus relies on the exponential kernel

$$\mathcal{H}(\tau - t, \Delta) = \frac{1}{\Delta} \exp\left(\frac{\tau - t}{\Delta}\right). \tag{37}$$

The corresponding Laplace transform is given by

$$\hat{\mathcal{H}}(\omega, \Delta) = \frac{1}{1 + i\omega\Delta}. \tag{38}$$

The cutoff frequency of this filter is given by $\omega_c = 1/\Delta$. Figure 7 depicts the real part of $\hat{\mathcal{H}}$ as a function of the frequency ω for $\Delta = 1$. Naturally, this cutoff frequency needs to be tuned so that the frequency associated to the instability one aims to kill is quenched by the filter.

For real applications, evaluating the convolution integral (36) is impractical as it necessitates the storage of the complete time history of $\mathbf{X}$. Consequently, it is replaced by its differential form given by

$$\dot{\mathbf{Y}} = \frac{1}{\Delta}(\mathbf{X} - \mathbf{Y}) \tag{39}$$

which can be integrated in time using classical integration schemes, e.g. second-order Euler. Combining (39) and (33) finally yields to the following extended system

$$\begin{cases} \dot{\mathbf{X}} = \boldsymbol{\mathcal{F}}(\mathbf{X}) - \chi(\mathbf{X} - \mathbf{Y}) \\ \dot{\mathbf{Y}} = \dfrac{1}{\Delta}(\mathbf{X} - \mathbf{Y}). \end{cases} \tag{40}$$

Implementing (40) into an existing time-stepping code requires only minor modifications, hence making it an easy choice for fixed point computation. It must be emphasized however that, because it relies on a low-pass filtering procedure, this selective frequency damping method is unable to quench non-oscillating instabilities, e.g. instabilities arising due to a pitchfork bifurcation. This particular point is one of its major limitations.

3.1.2 Newton-Krylov methods

While we relied on the continuous time representation of our system in §3.1.1, we now turn to its discrete-time counterpart. For that purpose, consider the following nonlinear system

$$\mathbf{x}_{k+1} = \boldsymbol{\mathcal{G}}(\mathbf{x}_k). \tag{41}$$

Our goal is thus to find a fixed point $\mathbf{x}^*$ of this system. Newton-Raphson method is a natural choice, provided the dimension of $\mathbf{x}$ is not too large. For large-scale dynamical systems, one may turn to the class of Newton-Krylov methods instead. These encompass a wide variety of different approaches, part of which have been reviewed in [46]. In the rest of this section, a variant of the recursive projection method (RPM) originally proposed by Shroff & Keller [71] will be presented.

Iteration (41) converges if all the eigenvalues $\{\mu_k\}_1^n$ of the Jacobian of $\boldsymbol{\mathcal{G}}$ lie in the unit disk and the initial iterate $\mathbf{x}_0$ is sufficiently close to the actual fixed point $\mathbf{x}^*$. It will however fail even if a single eigenvalue of the Jacobian lies outside the unit disk. Note that, for our purposes, the Jacobian of $\boldsymbol{\mathcal{G}}$ is given by the exponential propagator

$$\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T). \tag{42}$$

The basic Newton iteration reads

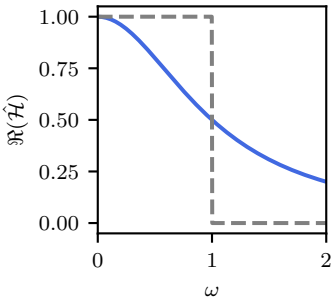

Fig. 7 Evolution of $\Re(\hat{\mathcal{H}})$ (—), i.e. the real part of the Laplace transform of the exponential filter, as a function of the frequency ω for $\Delta = 1$. The gray dashed line depicts the ideal spectral cutoff filter.

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k - (\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{M}})^{-1} (\tilde{\mathbf{x}}_k - \boldsymbol{\mathcal{G}}(\tilde{\mathbf{x}}_k)) \tag{43}$$

with

$$\lim_{k \to +\infty} \tilde{\mathbf{x}}_k = \mathbf{x}^*,$$

that is, as $k \to +\infty$, the Newton iterate $\tilde{\mathbf{x}}_k$ converges toward the fixed point $\mathbf{x}^*$ of the system under consideration. Note however that this Newton iteration requires the inversion of a large $n \times n$ matrix, something that may be quite impractical if n is large.

Let us now suppose that a small number m of eigenvalues lies outside the disk

$$K_\delta = \{|z| \leq 1 - \delta\},$$

that is

$$|\mu_1| \geq \cdots \geq |\mu_m| > 1 - \delta > |\mu_{m+1}| \geq \cdots \geq |\mu_n|.$$

and denote by $\mathbb{P}$ the maximal invariant subspace of $\boldsymbol{\mathcal{M}}$ belonging to $\{\mu_k\}_1^m$ while $\mathbb{Q}$ denotes its orthogonal complement, i.e. $\mathbb{P} + \mathbb{Q} = \mathbb{R}^n$. Introducing $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ as the orthogonal projectors onto these two subspaces, we have, for each $\mathbf{x} \in \mathbb{R}^n$, the unique decomposition

$$\mathbf{x} = \mathbf{p} + \mathbf{q}, \qquad \mathbf{p} \equiv \boldsymbol{\mathcal{P}}\mathbf{x} \in \mathbb{P}, \qquad \mathbf{q} \equiv \boldsymbol{\mathcal{Q}}\mathbf{x} \in \mathbb{Q}. \tag{44}$$

Using these two projectors, the Lyapunov-Schmidt decomposition of Eq. (41) finally reads

$$\mathbf{p}_{k+1} = \boldsymbol{f}(\mathbf{p}_k, \mathbf{q}_k) \equiv \boldsymbol{\mathcal{P}}\boldsymbol{\mathcal{G}}(\mathbf{p}_k + \mathbf{q}_k) \tag{45}$$

$$\mathbf{q}_{k+1} = \boldsymbol{g}(\mathbf{p}_k, \mathbf{q}_k) \equiv \boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{G}}(\mathbf{p}_k + \mathbf{q}_k). \tag{46}$$

As shown in [71], even though Eq. (41) may diverge, Eq. (46) is locally convergent on $\mathbb{Q}$ in the vicinity of the fixed point $\mathbf{x}^* = \mathbf{p}^* + \mathbf{q}^*$. The key idea of the *recursive projection method* is thus to stabilize Eq. (41) by using a Newton method within the low-dimensional unstable subspace $\mathbb{P}$ while continuing to use the classical fixed-point iteration within its orthogonal complement $\mathbb{Q}$. The stabilized system then reads

$$\begin{aligned} \mathbf{p}_{k+1} &= \mathbf{p}_k + \left(\boldsymbol{\mathcal{I}} - \boldsymbol{f}_p\right)^{-1} (\boldsymbol{f}(\mathbf{p}_k, \mathbf{q}_k) - \mathbf{p}_k) \\ \mathbf{q}_{k+1} &= \boldsymbol{g}(\mathbf{p}_k, \mathbf{q}_k), \end{aligned} \tag{47}$$

where $\boldsymbol{f}_p$ is the restriction of the Jacobian $\boldsymbol{\mathcal{M}}$ (evaluated at the current $\mathbf{x}_k$) onto the unstable subspace $\mathbb{P}$.

Solving directly the stabilized system (47) is quite impractical as it still requires the inversion of the $n \times n$ matrix $\boldsymbol{\mathcal{I}} - \boldsymbol{f}_p$. It must be noted however that $\boldsymbol{f}_p$ being the restriction of the Jacobian $\boldsymbol{\mathcal{M}}$ onto the low-dimensional unstable subspace $\mathbb{P}$, it has a low-rank structure. Consequently, given an orthonormal set of vectors $\boldsymbol{\mathcal{U}} \in \mathbb{R}^{n \times m}$ than spans $\mathbb{P}$, one can write

$$\begin{aligned} \mathbf{p} &= \mathcal{U}\mathbf{z} \\ \mathbf{q} &= \left(\mathcal{I} - \mathcal{U}\mathcal{U}^T\right)\mathbf{x}, \end{aligned} \tag{48}$$

where $\mathbf{z} \in \mathbb{R}^m$ (with $m \ll n$) is the projection of $\mathbf{p}$ onto the span of $\mathcal{U}$. Different procedures have been proposed to obtain the orthonormal set of vectors $\mathcal{U}$. Here, we use the Arnoldi or Krylov-Schur decompositions of $\mathcal{M}$ described in §3.2.1 and §3.2.2, respectively. One major benefit of these decompositions is that they do not require explicitly the matrix $\mathcal{M}$ but only a function that computes the corresponding matrix-vector product.

Starting from the original system $\mathbf{x}_{k+1} = \mathcal{G}(\mathbf{x}_k)$, the basic RPM update $\tilde{\mathbf{x}}_{k+1}$ can finally be expressed as

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1} - \mathcal{U}\mathcal{U}^T\left(\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k\right) + \mathcal{U}\left(\mathbf{I} - \mathcal{H}\right)^{-1}\mathcal{U}^T\left(\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k\right), \tag{49}$$

where $\mathcal{H} = \mathcal{U}^T\mathcal{M}\mathcal{U}$ is the projection of the high-dimensional Jacobian matrix onto the low-dimensional unstable subspace $\mathbb{P}$. By doing so, one only needs to invert a small $m \times m$ matrix at each iteration of the Newton-RPM solver.

Finally, looking at Eq. (49), RPM can be understood as a predictor-corrector. First, a new prediction $\mathbf{x}_{k+1}$ is obtained from the original system. Then, it is corrected by RPM in a two-step procedure:

1. the unstable part of the residual, $\mathcal{U}\mathcal{U}^T(\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k)$, is subtracted from the predicted iterate $\mathbf{x}_{k+1}$,
2. it is then replaced by its Newton correction, $\mathcal{U}\left(\mathbf{I} - \mathcal{H}\right)^{-1}\mathcal{U}\left(\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k\right)$, hence resulting in the new RPM iterate $\tilde{\mathbf{x}}_{k+1}$.

Although the present fixed-point computation strategy requires substantially more modifications of an existing time-stepper solver than the selective frequency damping procedure described in §3.1.1, it nonetheless has a number of key benefits. First, while selective frequency damping cannot compute the linearly unstable fixed point of a system if the associated instability is non-oscillating (i.e. associated to a real eigenvalue), the recursive projection method can. More importantly, the recursive projection method improves its approximation of the unstable subspace of the Jacobian matrix $\mathcal{M}$ at each iteration. Consequently, as the procedure converges to the fixed point $\mathbf{x}^*$, one obtains as a by-product really good approximations of the leading eigenvalues and associated eigenvectors of the matrix $\mathcal{M}$. Finally, the recursive projection method can relatively easily be extended to compute linearly unstable periodic orbits or to perform branch continuation. For more details about RPM and illustrations, interested readers are referred to [71, 38, 15, 66] and references therein.

3.1.3 BoostConv

The Newton-Krylov method presented in §3.1.2 is a valid alternative to selective frequency damping (see §3.1.1), particularly if a steady bifurcation occurs so that a non-oscillatory instability needs to be quenched as to recover the unstable stationary solution. Nevertheless, implementing RPM is not straightforward as it requires the

Jacobian matrix (or a good approximation) of the discrete-time system considered. Recently, [18] have introduced *BoostConv*, a new fixed point computation technique somehow related to the recursive projection method.

Let us rewrite Eq. (41) as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k \tag{50}$$

where $\mathbf{r}_k$ is the residual vector produced at each nonlinear iteration. It can be shown that the residual at time k and time $k+1$ are related by

$$\mathbf{r}_{k+1} \simeq \mathbf{r}_k - \mathcal{C}\mathbf{r}_k \tag{51}$$

where $\mathcal{C}$ is an unknown linear operator approximately governing the dynamics of the residual vector. The key idea of *BoostConv* is to define a corrected residual vector $\boldsymbol{\xi}_k$ such that the above equation becomes

$$\mathbf{r}_{k+1} \simeq \mathbf{r}_k - \mathcal{C}\boldsymbol{\xi}_k. \tag{52}$$

Clearly, the residual $\mathbf{r}_{k+1}$ is annihilated if one has

$$\mathcal{C}\boldsymbol{\xi}_k = \mathbf{r}_k. \tag{53}$$

Let us now consider the two Krylov sequence of residuals

$$\mathcal{X} = \text{span}\{\mathbf{r}_k\}_1^m \quad \text{and} \quad \mathcal{Y} = \text{span}\{\mathbf{r}_k - \mathbf{r}_{k+1}\}_1^m$$

so that

$$\mathcal{Y} \simeq \mathcal{C}\mathcal{X}.$$

Assuming that the corrected residual $\boldsymbol{\xi}_k$ is a linear combination of the previous residuals stored in $\mathcal{X}$, the least-square solution to Eq. (53) is given by

$$\boldsymbol{\xi}_k = \mathcal{X}\mathcal{Y}^\dagger \mathbf{r}_k, \tag{54}$$

where $\mathcal{Y}^\dagger$ is the Moore-Penrose pseudoinverse. Introducing this least-square solution into Eq. (52) yields the modified residual at time $k+1$

$$\tilde{\mathbf{r}}_{k+1} = (\mathcal{I} - \mathcal{Y}\mathcal{Y}^\dagger)\mathbf{r}_k. \tag{55}$$

Looking at the above equation, our least-square trick thus allows us to annihilate the residual within the subspace defined by the column span of $\mathcal{Y}$ while leaving it untouched in the orthogonal complement. Piecing everything together, the stabilized BoostConv iterate can finally be written as

$$\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1} - \mathcal{Y}\mathcal{Y}^\dagger (\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k) + \mathcal{X}\mathcal{Y}^\dagger (\mathbf{x}_{k+1} - \tilde{\mathbf{x}}_k). \tag{56}$$

Just like the recursive projection method, BoostConv can be understood as a predictor-corrector iterative scheme. First, a new prediction $\mathbf{x}_{k+1}$ is obtained from the original system. Then, it is corrected by BoostConv in a two-step procedure:

1. the unstable part of the residual $\mathcal{Y}\mathcal{Y}^{\dagger}\left(\mathbf{x}_{k+1}-\tilde{\mathbf{x}}_k\right)$ is first subtracted from the prediction $\mathbf{x}_{k+1}$
2. it is then replaced by its least-square correction $\mathcal{X}\mathcal{Y}^{\dagger}\left(\mathbf{x}_{k+1}-\tilde{\mathbf{x}}_k\right)$ which plays the same role as the low-dimensional Newton correction step in RPM.

Although RPM and BoostConv appear closely related, the latter does not require the Jacobian matrix $\mathcal{M}$ nor an estimate of it. It can moreover be implemented as a black box around an existing solver since it only requires the residual $\mathbf{r}_k$ as input and returns the corrected one $\boldsymbol{\xi}_k$ as output. Also, BoostConv can be easily adapted to stabilize periodic orbits [18].

3.1.4 Comparison of the different approaches

Computing the linearly unstable fixed point of the Navier-Stokes equations for the two-dimensional shear-driven cavity flow at $Re = 4150$ provides a typical benchmark to illustrate the performances of the three fixed points solvers presented, namely *selective frequency damping* (see §3.1.1), the *recursive projection method* (see §3.1.2) and *BoostConv* (see §3.1.3). The different methods have been setup as follows:

- *Selective frequency damping*: the cutoff frequency of the low-pass filter has been set to $\omega_c = 3.5$ while the gain is set to $\chi = 0.15$. These parameters, chosen based on trial and errors, provide the best performances for SFD that we have observed.
- *Recursive projection method*: the dimension of the Krylov subspace providing the orthonormal basis for the leading invariant subspace of the Jacobian matrix has been set to $k_{\text{dim}} = 10$. The outer RPM iteration $\mathbf{x}_{k+1} = \mathcal{G}(\mathbf{x}_k)$ has been setup so that it corresponds to 100 time-steps of the Navier-Stokes solver.
- *BoostConv*: it has been parametrized as RPM.

Each method iterates until the norm of the Navier-Stokes solver's residual is below $\varepsilon = 10^{-10}$. Figure 8 depicts the vorticity field of the linearly unstable solution to the Navier-Stokes equations computed by the recursive projection method. Although not shown, the other two methods converge toward the same unstable equilibrium solution. The evolution of the residual as a function of the number of iterations performed by the nonlinear Navier-Stokes solver is reported in figure 9. It appears quite clearly that the recursive projection method largely outperforms the selective frequency damping and BoostConv. On the other hand, BoostConv appears only marginally more efficient than the selective frequency damping procedure. This comparison is however biased as it does not include the computational cost of constructing the orthonormal projection basis needed in the RPM solver. Given how similar BoostConv and RPM are, this plot nonetheless highlights the importance of correctly approximating the leading unstable subspace of the Jacobian matrix. Although it has been partially addressed in the original paper [71] and in [66], this particular point currently focuses our efforts.

3.2 Linear stability and eigenvalue computation

The aim of linear stability analysis is to determine whether a perturbation $\mathbf{x}$, governed by

$$\dot{\mathbf{x}} = \boldsymbol{\mathcal{A}}\mathbf{x},$$

will grow or decay exponentially rapidly as $t \to \infty$. This asymptotic behavior is entirely governed by the eigenspectrum of the Jacobian matrix $\boldsymbol{\mathcal{A}}$: if at least one of its eigenvalues has a positive (resp. negative) real part, the linear system considered is unstable (resp. stable), see §2.2 for more details.

It must be emphasized that, within a time-stepper framework, one does not seek directly for the eigenpairs of the Jacobian matrix $\boldsymbol{\mathcal{A}}$ of the continuous-time problem. Instead, the problem considered is recast in the discrete-time framework as

$$\mathbf{x}_{k+1} = \boldsymbol{\mathcal{M}}\mathbf{x}_k, \tag{57}$$

where $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T)$ is the exponential propagator already introduced in §2.2, §2.3.1, and §3.1.2, and where T is the sampling period. The system is then linearly unstable if at least one eigenvalue μ of $\boldsymbol{\mathcal{M}}$ lies outside the unit disk, i.e. $|\mu| > 1$.

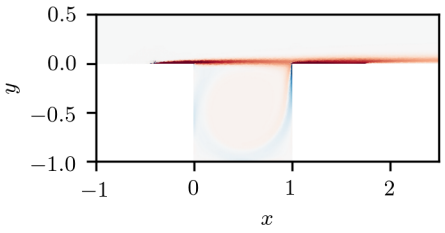

Fig. 8 Vorticity field of the linearly unstable solution to the Navier-Stokes equations obtained by the recursive projection method. Red denotes negative vorticity (i.e. clockwise rotation) while blue denotes positive vorticity (i.e. counter clockwise rotation).

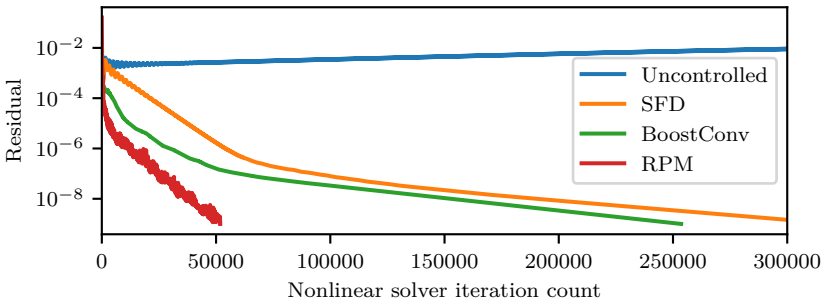

Fig. 9 Comparison of the residual evolution obtained by selective frequency damping (see §3.1.1), the recursive projection method (see §3.1.2) and BoostConv (see §3.1.3). The evolution of the uncontrolled residual is also reported for the sake of completeness. The benchmark considered is that of the two-dimensional shear-driven cavity flow at $Re = 4150$.

As discussed previously, although one cannot explicitly assemble the exponential propagator $\mathcal{M}$, its action onto a given vector $\mathbf{x}_k$ simply amounts to march in time the linearized system from $t = kT$ to $t = (k+1)T$. This ability to evaluate relatively easily the matrix-vector product given by (57) allows us to use iterative solvers in order to compute the eigenpairs of $\mathcal{M}$. The rest of this section is thus devoted to the presentation of two iterative eigenvalue solvers, namely the Arnoldi decomposition and the Krylov-Schur decomposition.

3.2.1 Arnoldi decomposition

Let us denote the following sequence of vectors

$$\mathcal{K}_m(\mathcal{M}, \mathbf{v}_0) = \left\{ \mathbf{v}_0, \mathcal{M}\mathbf{v}_0, \cdots, \mathcal{M}^{m-1}\mathbf{v}_0 \right\}. \tag{58}$$

Eq. (58) is known as a *Krylov sequence*. It eventually converges toward the eigenvector associated to the largest eigenvalue (in modulus) of $\mathcal{M}$ as $m \to \infty$. Generating this sequence to approximate the leading eigenpair of $\mathcal{M}$ is known as the *power iteration method*. Note that this simple method retains only the last vector of this sequence while discarding the information contained in the first $m-1$ vectors.

Contrary to the power iteration method, Arnoldi decomposition uses all of the information contained in the Krylov sequence (58) as to compute better estimates of the leading eigenvalues of $\mathcal{M}$. Readers can easily be convinced that the Krylov sequence (58) obeys

$$\mathcal{M}\mathcal{K}_m \simeq \mathcal{K}_m \mathcal{C},$$

where $\mathcal{C}$ is a $m \times m$ companion matrix representing the low-dimensional projection of $\mathcal{M}$ onto the span of the Krylov sequence (58). As such, the eigenpairs of $\mathcal{C}$ approximate the leading eigenpairs of $\mathcal{M}$. It must be emphasized however that, as m increases, the last vectors in the Krylov sequence become almost parallel. Consequently, the companion matrix $\mathcal{C}$ becomes increasingly ill-conditioned. In order to overcome the loss of information from the power iteration method and the increasingly ill-conditioned companion matrix decomposition, the Arnoldi method combines them with a Gram-Schmidt orthogonalization process. The basic Arnoldi iteration then reads

$$\mathcal{M}\mathcal{V}_m = \mathcal{V}_m \mathcal{H}_m + \mathbf{r}_m \mathbf{e}_m^T, \tag{59}$$

where $\mathcal{V}_m$ is an orthonormal set of vectors, $\mathcal{H}_m$ is a $m \times m$ upper Hessenberg matrix and $|\mathbf{r}_m \mathbf{e}_m^T|$ is the residual indicating how far $\mathcal{V}_m$ is from from an invariant subspace of $\mathcal{M}$. Because of its relatively small dimension, the eigenpairs $(\mu_H, \mathbf{y})$ of the Hessenberg matrix, also known as Ritz pairs, can be computed using direct eigensolvers. The Ritz pairs of $\mathcal{H}_m$ are related to the eigenpairs of $\mathcal{M}$ as follows

$$\begin{aligned} \mu_{\mathcal{M}} &\simeq \mu_{\mathcal{H}} \\ \hat{\mathbf{u}} &\simeq \mathcal{V}_m \mathbf{y}. \end{aligned} \tag{60}$$

A detailed presentation of the basic m-step Arnoldi factorization is given in algorithm (1) while figure 11 depicts its block-diagram representation to ease the understanding. As can be seen, Arnoldi decomposition is relatively simple to implement within an existing time-stepper code. One has to bear in mind however that, in order to capture (within a time-stepper framework) an eigenpair of the Jacobian matrix $\boldsymbol{\mathcal{A}}$ characterized by a circular frequency ω, one has to obey the Nyquist criterion and needs at least four snapshots to appropriately discretize the associated period.

Algorithm 1 The m-step *Arnoldi* factorisation.

Require: $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n\times n}$, starting vector $\mathbf{v} \in \mathbb{R}^n$.
$\mathbf{v}_1 = \mathbf{v}/\|\mathbf{v}\|$;
$\mathbf{w} = \boldsymbol{\mathcal{M}}\mathbf{v}_1$;
$\alpha_1 = \mathbf{v}_1^T\mathbf{w}$;
$\mathbf{f}_1 \leftarrow \mathbf{w} - \alpha_1\mathbf{v}_1$;
$\boldsymbol{\mathcal{V}}_1 \leftarrow (\mathbf{v}_1)$;
$\boldsymbol{\mathcal{H}}_1 \leftarrow (\alpha_1)$;
for $j = 1, 2, \cdots, m-1$ **do**
  $\beta_j = \|\mathbf{f}_j\|$;
  $\mathbf{v}_{j+1} \leftarrow \mathbf{f}_j/\beta_j$;
  $\boldsymbol{\mathcal{V}}_{j+1} \leftarrow \left(\boldsymbol{\mathcal{V}}_j, \mathbf{v}_{j+1}\right)$;
  $\hat{\boldsymbol{\mathcal{H}}}_j \leftarrow \begin{pmatrix} \boldsymbol{\mathcal{H}}_j \\ \beta_j \mathbf{e}_j^T \end{pmatrix}$
  $\mathbf{w} \leftarrow \boldsymbol{\mathcal{M}}\mathbf{v}_{j+1}$;
  $\mathbf{h} \leftarrow \boldsymbol{\mathcal{V}}_{j+1}^T\mathbf{w}$;
  $\mathbf{f}_{j+1} \leftarrow \mathbf{w} - \boldsymbol{\mathcal{V}}_{j+1}\mathbf{h}$;
  $\boldsymbol{\mathcal{H}}_{j+1} \leftarrow (\hat{\boldsymbol{\mathcal{H}}}_j, \mathbf{h})$;
end for

3.2.2 Krylov-Schur decomposition

Let us consider the m-step Arnoldi factorization

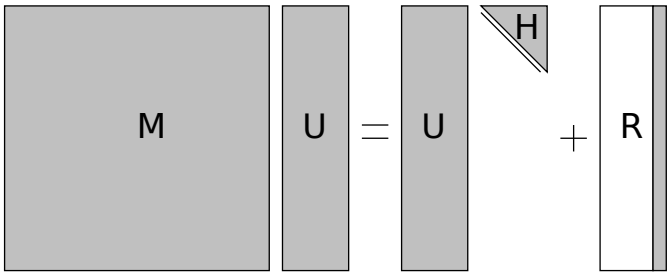

Fig. 10 *Arnoldi decomposition* – Given a matrix $\boldsymbol{\mathcal{M}} \in \mathbb{R}^{n\times n}$, construct an orthonormal set of vectors $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{n\times k}$ such that $\boldsymbol{\mathcal{H}} \in \mathbb{R}^{k\times k}$ is an upper Hessenberg matrix and only the last column of the residual matrix $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{n\times k}$ is nonzero. Figure has been adapted from [4].

$$\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m\boldsymbol{\mathcal{H}}_m + \beta\mathbf{v}_{m+1}\mathbf{e}_m^T \tag{61}$$

introduced in §3.2.1. As discussed previously, the Ritz pair $(\mu_H, \boldsymbol{\mathcal{V}}_m\mathbf{y})$ of $\boldsymbol{\mathcal{H}}_m$ provides a good approximation for the eigenpair $(\mu, \hat{\mathbf{u}})$ of the matrix $\boldsymbol{\mathcal{M}}$. One limitation of the Arnoldi decomposition is however that the dimension m of the Krylov subspace necessary to converge the leading Ritz pairs is not known *a priori*. It might hence be relatively large, thus potentially causing some numerical and/or practical problems (e.g. storage of Krylov basis $\boldsymbol{\mathcal{V}}_m$, forward instability of the Gram-Schmidt process involved in the Arnoldi decomposition, etc). Two different approaches have been proposed to overcome these limitations: the *Implicitly Restarted Arnoldi Method* introduced by Sorensen [73] in 1992 and the *Krylov-Schur decomposition* introduced by Stewart [75] in 2001. In the present work, the latter approach has been preferred because of its simplicity of implementation and its robustness.

The Krylov-Schur method is based on the generalization of the m-step Arnoldi factorization (61) to a *Krylov decomposition* of order m

$$\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{V}}_m = \boldsymbol{\mathcal{V}}_m\boldsymbol{\mathcal{B}}_m + \mathbf{v}_{m+1}\mathbf{b}_{m+1}^T \tag{62}$$

for which the matrix $\boldsymbol{\mathcal{B}}_m$ and the vector $\mathbf{b}_{m+1}$ have no restriction. The Arnoldi decomposition then appears as a special case of Krylov decomposition where $\boldsymbol{\mathcal{B}}_m$ is restricted to be in upper Hessenberg form and $\mathbf{b}_{m+1} = \mathbf{e}_m$. Another special case is the *Krylov-Schur* decomposition for which the matrix $\boldsymbol{\mathcal{B}}_m$ is in real Schur form (i.e. quasi-triangular form with its eigenvalues in the 1×1 or 2×2 diagonal blocks). It has been shown by Stewart [75] that Krylov and Arnoldi decompositions are equivalent (i.e. they have the same Ritz approximations). Moreover, by means of orthog-

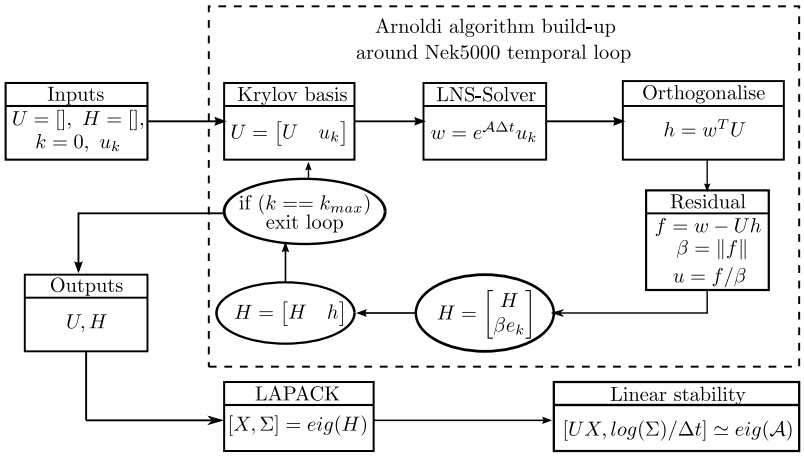

Fig. 11 Block-diagram representation of the basic m-step Arnoldi factorization. Note that, within a time-stepper framework, every matrix-vector product $\boldsymbol{\mathcal{M}}\mathbf{v}_i$ is evaluated by marching in time the linearized system considered.

onal similarity transformations, any Krylov decomposition can be transformed into an equivalent Krylov-Schur decomposition.

The core of the Krylov-Schur method is based on a two-steps procedure: (*i*) an expansion step performed using a m-step Arnoldi factorization, and (*ii*) a contraction step to a Krylov-Schur decomposition of order p retaining only the most useful spectral information from the initial m-step Arnoldi decomposition. Given an initial unit-norm vector $\mathbf{v}_1$, a subroutine to compute the matrix-vector product $\boldsymbol{\mathcal{M}}\mathbf{v}_i$, and the desired dimension m of the Krylov subspace, the Krylov-Schur method can be summarized as follows:

1. Construct an initial Krylov decomposition of order m using for instance the m-step Arnoldi factorization (61).
2. Check for the convergence of the Ritz eigenpairs. If a sufficient number has converged, then stop. Otherwise, proceed to step 3.
3. Compute the real Schur decomposition $\boldsymbol{\mathcal{B}}_m = \boldsymbol{\mathcal{Q}}\boldsymbol{\mathcal{S}}_m\boldsymbol{\mathcal{Q}}^T$ such that the matrix $\boldsymbol{\mathcal{S}}_m$ is in real Schur form and $\boldsymbol{\mathcal{Q}}$ is the associated matrix of Schur vectors. It is assumed furthermore that the Ritz values on the diagonal blocks of $\boldsymbol{\mathcal{S}}_m$ have been sorted such that the p "wanted" Ritz values are in the upper-left corner of $\boldsymbol{\mathcal{S}}_m$, while the $m-p$ "unwanted" ones are in the lower-right corner. At this point, we have the following re-ordered Krylov-Schur decomposition

$$\boldsymbol{\mathcal{M}}\tilde{\boldsymbol{\mathcal{V}}}_m = \tilde{\boldsymbol{\mathcal{V}}}_m \begin{bmatrix} \boldsymbol{\mathcal{S}}_{11} & \boldsymbol{\mathcal{S}}_{12} \\ \mathbf{0} & \boldsymbol{\mathcal{S}}_{22} \end{bmatrix} + \mathbf{v}_{m+1} \begin{bmatrix} \mathbf{b}_1^T & \mathbf{b}_2^T \end{bmatrix} \tag{63}$$

   with $\tilde{\boldsymbol{\mathcal{V}}}_m = \boldsymbol{\mathcal{V}}_m\boldsymbol{\mathcal{Q}}$ being the re-ordered Krylov basis, $\boldsymbol{\mathcal{S}}_{11}$ the subset of the Schur matrix containing the p "wanted" Ritz values, $\boldsymbol{\mathcal{S}}_{22}$ the subset containing the $m-p$ "unwanted" ones, and $\begin{bmatrix} \mathbf{b}_1^T & \mathbf{b}_2^T \end{bmatrix} = \mathbf{b}^T\boldsymbol{\mathcal{Q}}$.
4. Truncate the Krylov-Schur decomposition (63) of order m to a Krylov decomposition of order p,

$$\boldsymbol{\mathcal{M}}\tilde{\boldsymbol{\mathcal{V}}}_p = \tilde{\boldsymbol{\mathcal{V}}}_p\boldsymbol{\mathcal{S}}_{11} + \tilde{\mathbf{v}}_{p+1}\mathbf{b}_1^T \tag{64}$$

   with $\tilde{\boldsymbol{\mathcal{V}}}_p$ equal to the first p columns of $\tilde{\boldsymbol{\mathcal{V}}}_m$ and $\tilde{\mathbf{v}}_{p+1} = \mathbf{v}_{m+1}$.
5. Extend again to a Krylov decomposition of order m using a variation of the procedure used in the first step: the procedure is re-initialized with the starting vector $\mathbf{v}_{p+1}$ but all the vectors in $\tilde{\boldsymbol{\mathcal{V}}}_p$ are taken into account in the orthogonalization step.
6. Check the convergence of the Ritz values. If not enough Ritz values have converged, restart from step 3.

This algorithm has two critical steps. The first one is the choice of the "wanted" Ritz values in the re-ordering of the Schur decomposition in step 2. Since we are only interested in the leading eigenvalues of the linearized Navier-Stokes operator, all the Ritz pairs being classified as "wanted" must satisfy $|\mu_w| \geq 1-\delta$ (with $\delta = 0.05-0.1$ usually). Regarding the criterion assessing the convergence of a given Ritz pair, starting from the Krylov decomposition (61), one can write

$$\|\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{V}}_m\mathbf{y} - \boldsymbol{\mathcal{V}}_m\boldsymbol{\mathcal{B}}_m\mathbf{y}\| = \|\boldsymbol{\mathcal{M}}\boldsymbol{\mathcal{V}}_m\mathbf{y} - \mu_{\mathbf{B}}\boldsymbol{\mathcal{V}}_m\mathbf{y}\| = \left|\beta\mathbf{e}_m^T\mathbf{y}\right| \tag{65}$$

with $(\mu_{\mathbf{B}}, \mathbf{y})$ a given eigenpair of the matrix $\boldsymbol{\mathcal{B}}_m$. If the right hand side $\left|\beta \mathbf{e}_m^T \mathbf{y}\right|$ is smaller than a given tolerance, then the Ritz pair $(\mu_{\mathbf{B}}, \boldsymbol{\mathcal{V}}_m \mathbf{y})$ provides a good approximation to the eigenpair $(\mu, \hat{\mathbf{u}})$ of the original matrix $\boldsymbol{\mathcal{M}}$. A Ritz value is generally considered as being converged if the associated residual $\left|\beta \mathbf{e}_m^T \mathbf{y}\right| \leq 10^{-6}$.

3.2.3 Comparison of the two approaches

Following the comparison of the fixed points solvers in §3.1.4, let us now compare the efficiency of the time-stepper Krylov-Schur decomposition over the Arnoldi one when computing the leading eigenvalues and eigenmodes of the linearized Navier-Stokes operator for the shear-driven cavity flow at $Re = 4150$. For that purpose, the eigenspectrum obtained using the Arnoldi decomposition with a Krylov subspace of dimension $k_{\text{dim}} = 256$ will serve as our reference point. For the sake of comparison, three Krylov subspaces of various dimensions, namely k_{dim} =192, 128 and 64 have been considered for the Krylov-Schur decomposition. In all cases, twelve eigenvalues were required to have converged with a residual $\varepsilon \leq 10^{-6}$ before the computation could stop. Finally, the sampling period has been set to $T = 0.2$ non-dimensional time units so that the exponential propagator (whose action is approximated by time-marching the linearized Navier-Stokes equation) is given by $\boldsymbol{\mathcal{M}} = \exp(0.2\boldsymbol{\mathcal{A}})$, with $\boldsymbol{\mathcal{A}}$ being the linearized Navier-Stokes operator.

Left panel of figure 12 depicts the eigenspectra obtained using the Arnoldi decomposition with a Krylov subspace dimension $k_{\text{dim}} = 256$ and Krylov-Schur decomposition with $k_{\text{dim}} = 128$ and $k_{\text{dim}} = 64$, while its right panel shows the real part of the streamwise velocity component of some of the leading eigenmodes for the sake of completeness. These plots highlight the existence of two families of modes: (*i*) high-frequency shear layer modes (also known as *Rössiter* modes), and (*ii*) low-frequency inner-cavity modes similar to the ones existing in lid-driven cavities. A detailed description of the physical mechanisms underlying these instabilities is beyond the scope of the present work. Interested readers are referred to [8] regarding the shear layer instability modes and [24, 25] for the inner-cavity instabilities.

Table 1 reports the growth rate σ and circular frequency ω of the leading eigenvalue for all of the cases considered. Even with a Krylov subspace four times smaller than the reference one, it can be seen that the leading eigenvalue's growth rate computed by the Krylov-Schur decomposition differs by less than 0.02% compared to our reference solution while the circular frequency is left unchanged at least until the fifth digit. The major difference between $k_{\text{dim}} = 256$ and $k_{\text{dim}} = 64$ is the accuracy of the eigenvalues belonging to the branches of inner-cavity modes, see figure 12. These modes however turn out to be of limited interest in the dynamics of the flow.

Finally, the last row of table 1 reports the total number of calls to the linearized Navier-Stokes solver necessary to converge the required twelve eigenvalues. Compared to our reference case (i.e. $k_{\text{dim}} = 256$), the total number of matrix-vector multiplications is inversely proportional to the reduction factor of the dimension of the Krylov subspace considered. Despite this increase of the number of matrix-vector

multiplications, Krylov-Schur decomposition with a moderate subspace dimension (e.g. $k_{\text{dim}} = 64$ or 128) has nonetheless two major benefits compared to the classical Arnoldi decomposition:

- *Reduced memory footprint*: having a smaller Krylov subspace implies that fewer Krylov vectors need to be stored. This may be of crucial importance if one consider very large-scale eigenvalue problems such as the ones appearing in fluid dynamics (see [37, 49, 17, 14] for illustrations) which need to be solved on high-performance computers.
- *Partially reduced computation complexity*: although table 1 underlines that a larger number of calls to the linearized time-stepper code is required as we decrease the size of the Krylov subspace, one must not overlook that Arnoldi decomposition necessitates modified Gram-Schmidt orthogonalization of the Krylov sequence to iteratively construct the upper Hessenberg matrix. For an $n \times k$ matrix (where n is the number of degrees of freedom and k the dimension of the Krylov subspace), the computational complexity of this step scales as nk^2. As a consequence, decreasing the size of the Krylov subspace by a factor 4 reduces the computational cost of the modified Gram-Schmidt orthogonalization by a factor 16. Such a reduction becomes particularly attractive if ones needs a very large Krylov subspace to converge the leading eigenvalues when using the classical Arnoldi decomposition.

Finally, although Krylov-Schur decomposition does have some benefits compared to the classical Arnoldi iteration, it must not be forgotten that the overall computational time is dictated by the linearized time-stepper solver used to evaluate the application of the exponential propagator $\mathcal{M}$ onto a given vector. Consequently, efficient and scalable temporal integrators are key enablers for very large-scale eigenvalue analysis arising from the discretization of partial differential equations. Discussion on efficient and scalable temporal and/or spatial discretization is however beyond the scope of this contribution.

Table 1 Growth rate σ and circular frequency ω of the leading eigenvalue computed for different dimensions of the Krylov subspace. Note that only the largest Krylov subspace (i.e. $k_{\text{dim}} = 256$) uses the basic Arnoldi decomposition. All other computations have been performed with Krylov-Schur. The total number of calls to the linearized Navier-Stokes solver (i.e. the number of Jacobian matrix-vector multiplications) is also reported for each case.

	$k_{\text{dim}} = 256$	$k_{\text{dim}} = 192$	$k_{\text{dim}} = 128$	$k_{\text{dim}} = 64$
σ	$4.56757 \cdot 10^{-3}$	$4.56757 \cdot 10^{-3}$	$4.56757 \cdot 10^{-3}$	$4.56673 \cdot 10^{-3}$
ω	± 7.4938	± 7.4938	± 7.4938	± 7.4938
Matrix-vector multiplications	256	384	512	832

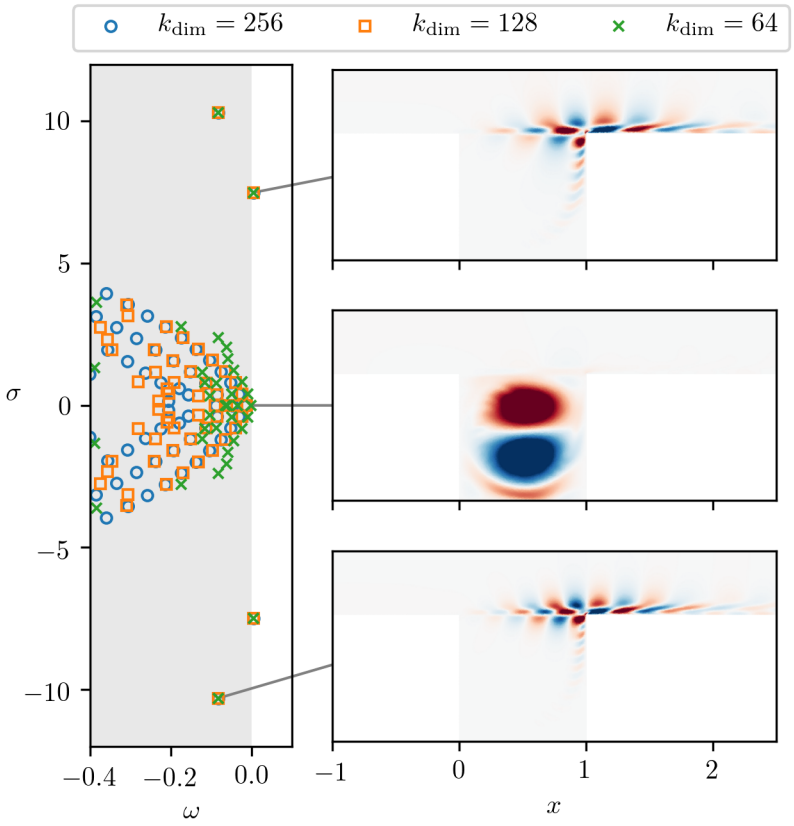

Fig. 12 Eigenspectrum and leading eigenmodes (streamwise velocity field) for the shear-driven cavity flow at $Re = 4150$. Blue circles (○) depict the eigenvalues obtained using the Arnoldi decomposition (see §3.2.1) with a Krylov subspace of dimension $k_{\mathrm{dim}} = 256$ while orange squares (□) and green crosses (×) depict the eigenvalues obtained using the Krylov-Schur decomposition (see §3.2.2) with a Krylov subspace of dimension $k_{\mathrm{dim}} = 128$ and $k_{\mathrm{dim}} = 64$, respectively. In both cases, the computation stopped once the twelve eigenvalues have been converged down to $\varepsilon = 10^{-6}$.

3.3 Non-modal stability and singular value decomposition

Given the linear time-invariant dynamical system

$$\dot{\mathbf{x}} = \boldsymbol{\mathcal{A}}\mathbf{x} + \mathbf{f},$$

it has been shown in §2.2 that, for $\mathbf{f} = \mathbf{0}$ (i.e. no external forcing), the asymptotic fate of a random initial condition $\mathbf{x}_0$ is dictated by the eigenpairs of the Jacobian matrix $\boldsymbol{\mathcal{A}}$. On the other hand, as shown in §2.3, its short-term dynamics are governed by the singular triplets of the exponential propagator $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T)$, where T is the

time horizon considered. Conversely, the asymptotic response of the (stable) system to an external forcing $\mathbf{f}$ is governed by the singular triplets of the so-called resolvent operator $\boldsymbol{\mathcal{R}} = (i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}})^{-1}$, where ω is the forcing frequency. The rest of this section is devoted to the presentation of two different time-stepper algorithms for the computation of the leading singular values and singular vectors of the exponential propagator $\boldsymbol{\mathcal{M}}$ or the resolvent operator $\boldsymbol{\mathcal{R}}$.

3.3.1 An optimization approach

It has been shown in §2.3 that optimal perturbation analysis could be formulated as an optimization problem given by

$$\begin{aligned} &\underset{\mathbf{x}_0}{\text{maximize}}\ \mathcal{J}(\mathbf{x}_0) = \|\mathbf{x}(T)\|_2^2 \\ &\text{subject to } \dot{\mathbf{x}} - \boldsymbol{\mathcal{A}}\mathbf{x} = 0 \\ &\qquad\qquad \|\mathbf{x}_0\|_2^2 - 1 = 0, \end{aligned} \tag{66}$$

where $\mathcal{J}(\mathbf{x}_0)$ is known as the *objective function*. Similarly, the optimal forcing problem can be formulated as

$$\begin{aligned} &\underset{\hat{\mathbf{f}}}{\text{maximize}}\ \mathcal{J}\left(\hat{\mathbf{f}}\right) = \|\hat{\mathbf{x}}(\omega)\|_2^2 \\ &\text{subject to } (i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}})\,\hat{\mathbf{x}} = \hat{\mathbf{f}} \\ &\qquad\qquad \|\hat{\mathbf{f}}\|_2^2 - 1 = 0, \end{aligned} \tag{67}$$

Though these optimization problems are non-convex, solutions to both of them can be obtained by means of standard gradient-based optimization algorithms. One of the most famous such algorithms is the *conjugate gradient* method originally introduced by [35], see [68] and [32] for more recent presentations. In this work we will however introduce the reader to the *rotation update* technique, a modification of the classical steepest ascent method based on geometric considerations. Figure 13 provides a schematic description of this algorithm. This approach has been used by [28, 29] and [23] in the context of *p-norms* optimization in fluid dynamics.

Both Eq. (66) and Eq. (67) are constrained maximization problems. As shown in §2.3, introducing Lagrange multipliers allows us to transform these constrained problems into equivalent unconstrained ones. For the optimal perturbation analysis, the unconstrained maximization problem thus reads

$$\underset{\mathbf{x},\mathbf{v},\mu}{\text{maximize}}\ \mathcal{L}(\mathbf{x},\mathbf{v},\mu), \tag{68}$$

where

$$\mathcal{L}(\mathbf{x},\mathbf{v},\mu) = \mathcal{J}(\mathbf{x}_0) + \int_0^T \mathbf{v}^T(\dot{\mathbf{x}} - \boldsymbol{\mathcal{A}}\mathbf{x})\,\mathrm{d}t + \mu\left(\|\mathbf{x}_0\|_2^2 - 1\right) \tag{69}$$

is known as the *augmented Lagrangian* function. The additional optimization variables $\mathbf{v}$ and μ appearing in the definition of the augmented Lagrangian $\mathcal{L}$ are the *Lagrange multipliers*. The gradient of the augmented Lagrange functional $\mathcal{L}$ with respect to the initial condition $\mathbf{x}_0$ reads

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_0} = 2\mu\mathbf{x}_0 - \mathbf{v}_0. \tag{70}$$

This expression explicitly depends on the Lagrange multiplier μ whose value is unfortunately unknown. One can however write down a mathematical expression of this gradient orthogonalized with respect to the input $\mathbf{x}_0$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp} = \frac{\partial \mathcal{L}}{\partial \mathbf{x}} - \frac{\langle \frac{\partial \mathcal{L}}{\partial \mathbf{x}}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}\mathbf{x} \tag{71}$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product. Note moreover that we dropped the subscript 0 for the sake of simplicity. It can now be expressed as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp} = (\mathbf{v} - 2\mu\mathbf{x}) - \frac{\langle (\mathbf{v} - 2\mu\mathbf{x}), \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}\mathbf{x} \tag{72}$$

After simplifications, the orthogonalized gradient finally reads

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp} = \mathbf{v} - \frac{\langle \mathbf{v}, \mathbf{x} \rangle}{\langle \mathbf{x}, \mathbf{x} \rangle}\mathbf{x} \tag{73}$$

This expression now solely depends on the direct variable $\mathbf{x}$ and the adjoint one $\mathbf{v}$, while the dependence on the unknown Lagrange multiplier μ has been completely removed from the optimization problem. Normalizing this new gradient such that

$$\mathbf{G}^n = \sqrt{\frac{\|\mathbf{x}_0\|_2^2}{\langle \frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp}, \frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp} \rangle}} \frac{\partial \mathcal{L}}{\partial \mathbf{x}}^{\perp} \tag{74}$$

now allows us to look for the update $\mathbf{x}^{n+1}$ as a simple linear combination of $\mathbf{x}^n$ and $\mathbf{G}^n$ given by

$$\mathbf{x}^{n+1} = \cos(\alpha)\mathbf{x}^n + \sin(\alpha)\mathbf{G}^n \tag{75}$$

Since $\mathbf{x}^n$ and $\mathbf{G}^n$ form an orthonormal set of vectors, the update $\mathbf{x}^{n+1}$ fulfills, directly by construction, the constraint on the amplitude of the initial perturbation. No quadratic equation in μ, as in the case of steepest ascent method, need to be solved anymore at each iteration of the optimization loop. To ensure the convergence of the method to the maxima of the augmented functional $\mathcal{L}$, a check needs however to be put on the value of the angle α used for the update of the solution. Every calculations presented in this work uses $\alpha = 0.5$ as the initial value. If the value of the cost function $\mathcal{J}$ computed at the $(n+1)^{\text{th}}$ iteration is smaller than the value of $\mathcal{J}$

at the previous one, then the update $\mathbf{x}^{n+1}$ is re-updated with a different value of α, typically $\alpha = \alpha/2$ until the new value of $\mathcal{J}$ is larger than the previous one.

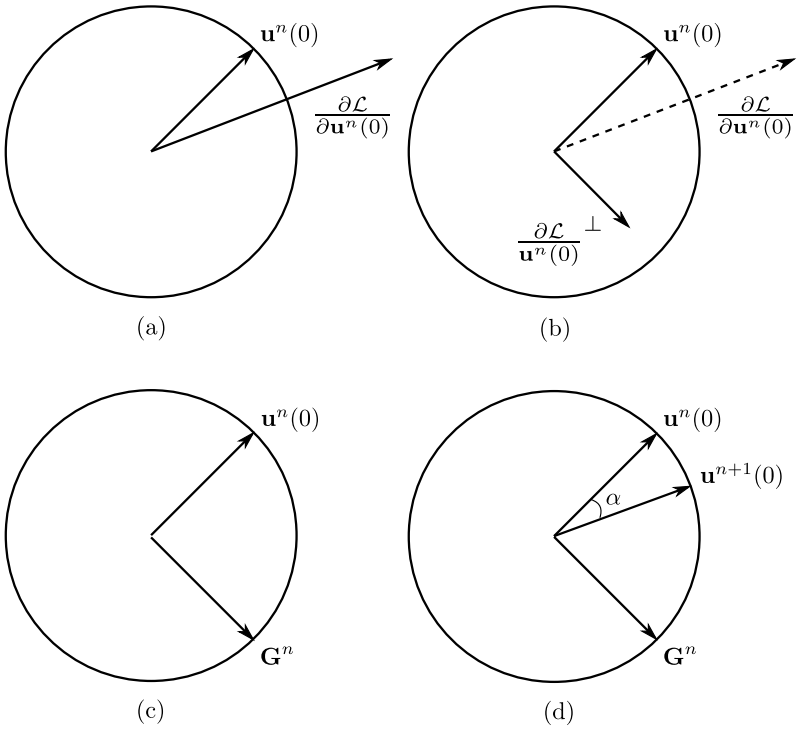

Fig. 13 Schematic representation of the *rotation update* method. (a) Compute the gradient $\partial\mathcal{L}/\partial\mathbf{x}$ of the augmented Lagrange functional. (b) Orthogonalise the gradient with respect to $\mathbf{x}^n(0)$. (c) Compute $\mathcal{G}^n$, i.e. the orthogonalized gradient normalized such that its energy is E_0. (d) Update $\mathbf{x}^{n+1}(0)$ using a linear combination $\mathbf{x}^n(0)$ and of the orthonormalized gradient $\mathbf{G}^n$.

3.3.2 Singular value decomposition

Formulating the optimal perturbation analysis as a maximization problem yields a non-convex optimization problem. As a consequence, one cannot rule out the possibility that gradient-based algorithms get stuck on a local maximum. Hopefully, it has been shown in §2.3 that the same problem could be formulated as a singular value decomposition of either the exponential propagator $\boldsymbol{\mathcal{M}} = \exp(\boldsymbol{\mathcal{A}}T)$ or the resolvent one $\boldsymbol{\mathcal{R}} = (i\omega\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{A}})^{-1}$, depending on the problem considered.

Let us consider once more the optimal perturbation problem for the sake of simplicity, although the derivation to be given naturally extend to the case to the resolvent. Given the singular value decomposition of the exponential propagator

$$\mathcal{M} \triangleq \exp(\mathcal{A}T) = \mathcal{U}\Sigma\mathcal{V}^H,$$

the optimal perturbation at $t = 0$ is given by the right singular vector $\mathbf{v}_1$, i.e. the first column of $\mathcal{V}$, while the associated response at time $t = T$ is given by the rescaled left singular vector $\sigma_1\mathbf{u}_1$, where σ_1 is the associated singular value characterizing the amplification of the perturbation. Computing directly the singular values and singular vectors of $\mathcal{M}$ is a challenging task for very large scale problems. Hopefully, introducing the adjoint exponential propagator $\mathcal{M}^\dagger = \exp\left(\mathcal{A}^\dagger T\right)$, readers can easily be convinced that our problem can be recast as the following equivalent eigenvalue problems

$$\mathcal{M}^\dagger\mathcal{M}\mathbf{v} = \sigma^2\mathbf{v} \quad \text{and} \quad \mathcal{M}\mathcal{M}^\dagger\mathbf{u} = \sigma^2\mathbf{u}. \tag{76}$$

From a practical point of view, evaluating the action of the matrix $\mathcal{M}^\dagger\mathcal{M}$ onto a vector $\mathbf{x}$ can be computed in a two-step procedure:

1. Integrate forward in time the original system, $\mathbf{x}(T) = \exp(\mathcal{A}T)\mathbf{x}$.
2. Integrate backward in time the adjoint problem using the output of the previous step as the initial condition, i.e. evaluating $\exp\left(\mathcal{A}^\dagger T\right)\mathbf{x}(T)$.

Provided an adjoint time-stepper code is available, one can thus readily solve the optimal perturbation problem using the eigenvalue solvers described in §3.2.1 or §3.2.2. Moreover, given that $\mathcal{M}^\dagger\mathcal{M}$ is a symmetric positive-definite matrix, the upper Hessenberg matrix in the Arnoldi iteration can be replaced by a tri-diagonal matrix, hence resulting in the so-called *Lanczos* iteration [47]. Note finally that, when applied to the resolvent operator $\mathcal{R}$, the matrix-vector product $\mathcal{R}(\omega)\hat{\mathbf{f}}$ can be evaluated as follows:

1. Integrate forward in time the system $\dot{\mathbf{x}} = \mathcal{A}\mathbf{x} + \mathbf{f}(\omega)$.
2. Perform a discrete Fourier transform of the asymptotic response to obtain $\hat{\mathbf{u}}(\omega)$.

The same procedure applies for the action of $\mathcal{R}^\dagger(\omega)$ where one now needs to integrate backward in time the adjoint system using $\hat{\mathbf{u}}(\omega)$ as the external forcing. For more details about the computation of the optimal forcing using a time-stepper approach, readers are referred to [56].

3.3.3 Illustration

As for modal stability (see §3.2.3), let us illustrate non-modal stability on the shear-driven cavity flow. For that purpose, the Reynolds number is set to $Re = 4100$, i.e. slightly below the critical Reynolds number for the onset of linear instability. Only the optimal perturbation analysis (time-domain) will be presented for the sake of simplicity. For more details about the resolvent analysis (frequency domain), readers are referred to [56, 13]. Figure 14 depicts the evolution in time of the optimal perturbation's kinetic energy. It can be seen that, although linear stability analysis predicts that the flow is stable, perturbations can be amplified by 4 to 5 orders of magnitude solely through non-modal effects. Once the perturbation has reached its maximum transient amplification at $t = 2$, its fate is eventually dictated by the least

stable eigenvalue of the Jacobian matrix. Note that in the present case, the Reynolds number considered being slightly below that for the onset of instability, the eventual decay rate of the perturbation is relatively small. The perturbation nonetheless eventually disappears at $t \to +\infty$. Figure 14 also clearly illustrates the different spatial support of the optimal initial perturbation (left panel) and the associated optimal response at $t = 2$ (right panel). These different spatial supports result from the strong convective effects, which are related mathematically to the degree of non-normality of the Jacobian matrix. Similar observation holds true regarding the optimal forcing and optimal response when performing a resolvent analysis. Analysis of these transient (non-normal) effects may be of crucial importance when studying subcritical transition or for control purposes.

Let us finally conclude this section by presenting the pros and cons of the SVD and optimization approaches. As discussed earlier, formulating the optimal perturbation and optimal forcing analyses in an optimization framework results in a non-convex optimization problem typically solved using gradient-based algorithms. Consequently, one cannot rule out the possibility that the solution returned by the optimization procedure actually corresponds to a local maxima of the problem at hand. On the other hand, formulating these two problems as singular value decompositions of the appropriate operator ensure that the solution obtained is indeed the optimal one by virtue of the Eckart-Young theorem. Moreover, singular value decomposition allows us to compute in one go not only the optimal perturbation but also the sub-optimal ones, something hardly possible within a classic optimization framework. Nonetheless, the optimization formulation offers much more flexibility than simply computing the optimal perturbation in the ℓ_2 sense. Indeed, one can choose the objective function $\mathcal{J}(\mathbf{x})$ and the associated constraints according to the specific problem he/she aims to solve, see for instance [28, 29, 23] for optimization based on the ℓ_1 norm of the perturabtion.

4 Conclusions and perspectives

With the ever increasing computational power available (roughly 20 to 25% increase annually) and the development of high-performances computing (HPC), investigating the properties of realistic very large-scale nonlinear dynamical systems has become reachable. In the field of fluid dynamics, computation of fixed points of two-dimensional flows and investigation of the spectral properties of the corresponding linearized Navier-Stokes operator are now routinely performed on workstations or even laptops. The traditional way to do so is to use a so-called *matrix-forming* approach where the Jacobian matrix of the system is explicitly assembled, whether one aims at computing a fixed point of the equations using Newton-like methods or at computing its leading eigenvalues and eigenmodes characterizing the linear stability properties of the fixed point considered. It must be noted however that the memory capabilities of computers increase at a slower rate than their computational capabilities. As a consequence, while simulations of very large-scale systems can

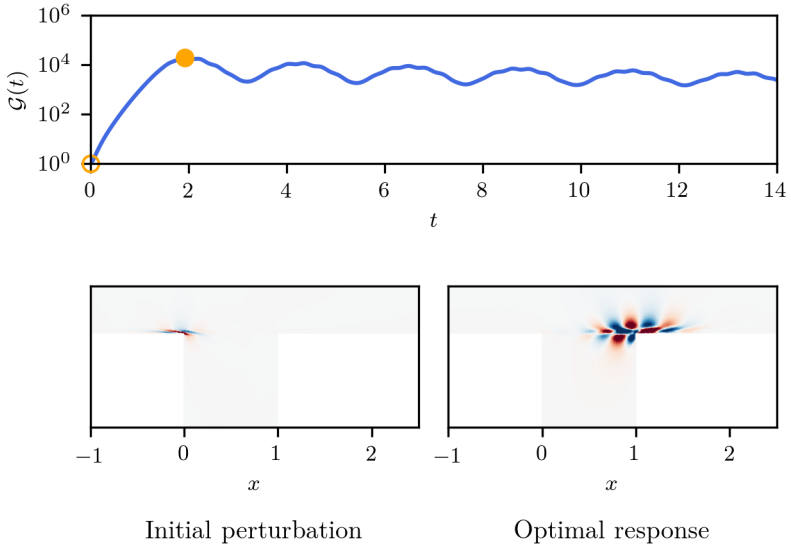

Fig. 14 Time-evolution of the optimal perturbation's kinetic energy for the shear-driven cavity flow at $Re = 4100$, i.e. below the critical Reynolds number for the onset of linear instability. The lower panels depict the streamwise velocity of the initial optimal perturbation (left) and associated optimal response (right) at $T = 2$.

now be performed (see [65] where the three-dimensional Navier-Stokes equations have been discretized using 10^{13} cells), using a matrix-forming approach to compute fixed points and study the stability properties of such systems becomes rapidly intractable. This gap between CPU and memory performances sprung the development of a new class of algorithms known as *matrix-free*.

In this chapter, the reader has been introduced to a number of such matrix-free algorithms for the computation of fixed points and eigenpairs of the linearized operator. Most of these algorithms rely on the observation that existing simulation codes do not solve explicitly the continuous-time problem

$$\dot{\mathbf{x}} = \boldsymbol{\mathcal{F}}(\mathbf{x})$$

but rather its discrete-time counterpart

$$\mathbf{x}_{k+1} = \boldsymbol{\mathcal{G}}(\mathbf{x}_k).$$

Moreover, these time-stepper simulation codes do not form explicitly the matrices but only require to be able to compute their applications onto a given set of vectors. Given this observation, only minor modifications of existing time-stepper codes are required as to transform them into black- functions evaluating matrix-vector products, hence enabling practitioners to wrap them into powerful matrix-free iterative fixed points and/or eigenvalues solvers. Once again in the field of fluid dynamics,

such an approach proved successful and allowed [37, 49, 17, 14] to investigate the stability properties of the fixed points of nonlinear dynamical systems characterized by almost 50 millions degrees of freedom.

Time-stepper matrix-free approaches nonetheless suffer from a number of limitations and drawbacks. First and foremost, these approaches rely onto an existing simulation code to emulate the matrix-vector products required. As a consequence, the overall performances of the fixed points and eigenvalue solvers presented herein are essentially dictated by the efficiency of the time-stepper code considered. Second, a number of analyses (non-modal stability, receptivity, sensitivity, ...) may require the definition of an adjoint. While such adjoint operator simply reduces to the transconjugate operation in matrix-forming approaches, a dedicated adjoint time-stepper solver needs to be developed within the matrix-free framework. Although this may be quite challenging if the system is defined by a complicated set of nonlinear partial differential equations, one must note that recent developments in automatic differentiation might prove helpful (see the software TAPENADE [33] for instance). Finally, because the eigenvalue solvers described herein rely on Krylov techniques, one must bear in mind that only the leading subset of eigenpairs of the Jacobian matrix can be accurately computed.

Despite these limitations, time-stepper matrix-free approaches offer a practical and efficient computing framework for the investigation of very large-scale nonlinear dynamical systems. Provided one has access to an efficient time-stepper solver, their relative ease of implementation make the approaches described in the present chapter a standard choice of tools whenever matrix-forming approaches are intractable. Finally, these approaches can easily be combined with finite-differences approximation of the Jacobian matrix-vector product whenever a linearized time-stepper solver is not available, hence proving extremely versatile and applicable a very broad class of systems.

References

1. *Intel Math Kernel Library. Reference Manual*. Intel Corporation, 2009.
2. E Åkervik, L Brandt, D S Henningson, J Hœpffner, O Marxen, and P Schlatter. Steady solutions of the navier-stokes equations by selective frequency damping. *Physics of fluids*, 18(6):068102, 2006.
3. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen. Lapack users' guide (3rd ed.). SIAM, 1999.
4. A C Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.
5. W.E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:1951, 1951.
6. S. Bagheri, E. Åkervik, L. Brandt, and D. S. Henningson. Matrix-free methods for the stability and control of boundary layers. *AIAA journal*, 47(5):1057–1068, 2009.
7. Shervin Bagheri, DS Henningson, J Hoepffner, and PJ Schmid. Input-output analysis and control design applied to a linear model of spatially developing flows. *Applied Mechanics Reviews*, 62(2):020803, 2009.

8. J. Basley, L. R. Pastur, F. Lusseyran, T. M. Faure, and N. Delprat. Experimental investigation of global structures in an incompressible cavity flow using time-resolved piv. *Exp Fluids*, 50:905–918, 2011.
9. S Boyd and L Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
10. L Brandt. The lift-up effect: The linear mechanism behind transition and turbulence in shear flows. *European Journal of Mechanics-B/Fluids*, 47:80–96, 2014.
11. T.J. Bridges and P.J. Morris. Differential eigenvalue problems in which the parameter appears nonlinearly. *J. Comput. Phys.*, 55:437–460, 1984.
12. T.J. Bridges and P.J. Morris. Boundary layer stability calculations. *Phys. Fluids*, 30:(11), 1987.
13. A. M. Bucci. *Subcritical and supercritical dynamics of incompressible flow over miniaturized roughness elements*. PhD thesis, École Nationale Supérieure d'Arts et Métiers-ENSAM, 2017.
14. MA Bucci, DK Puckert, C Andriano, J-Ch Loiseau, S Cherubini, J-Ch Robinet, and U Rist. Roughness-induced transition by quasi-resonance of a varicose global mode. *Journal of Fluid Mechanics*, 836:167–191, 2018.
15. Michele Sergio Campobasso and Michael B Giles. Stabilization of a linear flow solver for turbomachinery aeroelasticity using recursive projection method. *AIAA journal*, 42(9):1765–1774, 2004.
16. Kevin K Chen and Clarence W Rowley. H 2 optimal actuator and sensor placement in the linearised complex ginzburg–landau system. *Journal of Fluid Mechanics*, 681:241–260, 2011.
17. V Citro, F Giannetti, P Luchini, and F Auteri. Global stability and sensitivity analysis of boundary-layer flows past a hemispherical roughness element. *Physics of Fluids*, 27(8):084110, 2015.
18. V Citro, P Luchini, F Giannetti, and F Auteri. Efficient stabilization and acceleration of numerical simulation of fluid flows by residual recombination. *Journal of Computational Physics*, 344:234–246, 2017.
19. G Cunha, P-Y Passaggia, and M Lazareff. Optimization of the selective frequency damping parameters using model reduction. *Physics of Fluids*, 27(9):094103, 2015.
20. A. Davey and P. J. Drazin. The stability of poiseuille flow in a pipe. *J. Fluid Mech.*, 36:209–218, 1969.
21. Henk A Dijkstra, Fred W Wubs, Andrew K Cliffe, Eusebius Doedel, Ioana F Dragomirescu, Bruno Eckhardt, Alexander Yu Gelfgat, Andrew L Hazel, Valerio Lucarini, Andy G Salinger, et al. Numerical bifurcation methods and their application to fluid dynamics: analysis beyond simulation. *Communications in Computational Physics*, 15(1):1–45, 2014.
22. W. S. Edwards, L. S. Tuckerman, R. A. Friesner, and D. C. Sorensen. Krylov methods for the incompressible navier-stokes equations. *Journal of computational physics*, 110(1):82–102, 1994.
23. Mirko Farano, Stefania Cherubini, Jean-Christophe Robinet, and Pietro De Palma. Subcritical transition scenarios via linear and nonlinear localized optimal perturbations in plane poiseuille flow. *Fluid Dynamics Research*, 48(6):061409, 2016.
24. T. M Faure, P. Adrianos, F. Lusseyran, and L. Pastur. Visualizations of the flow inside an open cavity at medium range reynolds numbers. *Exp Fluids*, 42:169–184, 2007.
25. T. M Faure, L. Pastur, F. Lusseyran, Y. Fraigneau, and D. Bisch. Three-dimensional centrifugal instabilities development inside a parallelepipedic open cavity of various shape. *Exp Fluids*, 47:395–410, 2009.
26. P. Fischer, J. Kruse, J. Mullen, H. Tufo, J. Lottes, and S. Kerkemeier. Open source spectral element cfd solver. https://nek5000.mcs.anl.gov/index.php/MainPage., 2008.
27. D. P. G. Foures, C. P. Caulfield, and P. J. Schmid. Optimal mixing in two-dimensional plane Poiseuille flow at finite Péclet number. *J. Fluid Mech.*, 748:241–277, 2012.
28. D. P. G. Foures, C. P. Caulfield, and P. J. Schmid. Localization of flow structures using ∞-norm optimization. *J. Fluid Mech.*, 729:672–701, 2013.
29. D. P. G. Foures, C. P. Caulfield, and P. J. Schmid. Variational framework for flow optimization using seminorm constraints. *Physical Review E*, 86 (2):026306, 2014.
30. J. Gary and R. Helgason. A matrix method for ordinary differential eigenvalue problems. *J. Comput. Phys.*, 5:169–187, 1970.

31. M. Gaster and R. Jordinson. On the eigenvalues of the orr-sommerfeld equation. *J. Fluid Mech.*, 72:121–133, 1975.
32. Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
33. L. Hascoët and V. Pascual. The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. *ACM Transactions On Mathematical Software*, 39(3), 2013.
34. F. Hecht. New development in freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
35. Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.
36. RE Hunt and DG Crighton. Instability of flows in spatially developing media. In *Proc. R. Soc. Lond. A*, volume 435, pages 109–128. The Royal Society, 1991.
37. Miloš Ilak, Philipp Schlatter, Shervin Bagheri, and Dan S Henningson. Bifurcation and stability analysis of a jet in cross-flow: onset of global instability at a low velocity ratio. *Journal of Fluid Mechanics*, 696:94–121, 2012.
38. Vladimír Janovskỳ and O Liberda. Continuation of invariant subspaces via the recursive projection method. *Applications of Mathematics*, 48(4):241–255, 2003.
39. B E Jordi, C J Cotter, and S J Sherwin. Encapsulated formulation of the selective frequency damping method. *Physics of Fluids*, 26(3):034101, 2014.
40. B E Jordi, C J Cotter, and S J Sherwin. An adaptive selective frequency damping method. *Physics of Fluids*, 27(9):094104, 2015.
41. R. Jordinson. The flat plate boundary layer. part 1. numerical integration of the orr-sommerfeld equation,. *J. Fluid Mech.*, 43:801–811, 1970.
42. R. Jordinson. Spectrum of eigenvalues of the orr sommerfeld equation for blasius flow. *Phys. Fluids*, 14:2535, 1971.
43. G Karniadakis and S Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2nd edition edition, 2005.
44. R R Kerswell, C C T Pringle, and A P Willis. An optimization approach for analysing nonlinear stability with transition to turbulence in fluids as an exemplar. *Reports on Progress in Physics*, 77(8):085901, 2014.
45. R.R. Kerswell. Nonlinear nonmodal stability theory. *Ann. Rev. Fluid Mech.*, 50:319–345, 2018.
46. D A Knoll and D E Keyes. Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.
47. C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bureau Standards*, Sec. B, 45:255–282, 1950.
48. M T Landahl. A note on an algebraic instability of inviscid parallel shear flows. *Journal of Fluid Mechanics*, 98(2):243–251, 1980.
49. Jean-Christophe Loiseau, Jean-Christophe Robinet, Stefania Cherubini, and Emmanuel Leriche. Investigation of the roughness-induced transition: global stability analyses and direct numerical simulations. *Journal of Fluid Mechanics*, 760:175–211, 2014.
50. P. Luchini and A. Bottaro. Adjoint equations in stability analysis. *Ann. Rev. Fluid Mech.*, 46:493–517, 2014.
51. L. Mack. Boundary layer stability theory. Technical report 900-277, Jet Propulsion Laboratory, Pasadena, 1969.
52. L. Mack. A numerical study of the temporal eigenvalue spectrum of the blasius boundary layer. *J. Fluid Mech.*, 73:497–520, 1976.
53. M.R. Malik. Numerical methods for hypersonic boundary layer stability. *J. Comput. Phys.*, 86:376–413, 1990.
54. M.R. Malik, T.A. Zang, and M.Y. Hussaini. a spectral collocation method for the navier-stokes equations. *J. Comput. Phys.*, 61:64–88, 1985.
55. P. Manneville. Transition to turbulence in wall-bounded flows: Where do we stand? *Mechanical Engineering Reviews*, 3(2):15–00684, 2016.
56. Antonios Monokrousos, Espen Åkervik, Luca Brandt, and Dan S Henningson. Global three-dimensional optimal disturbances in the blasius boundary-layer flow using time-steppers. *Journal of Fluid Mechanics*, 650:181–214, 2010.

57. M. Nayar and U. Ortega. Computation of selected eigenvalues of generalized eigenvalues problems. *J. Comput. Phys.*, 108:8 – 14, 1993.
58. S. A. Orszag. Accurate solution of the orr-sommerfeld stability equation. *J. Fluid Mech.*, 50:659–703, 1970.
59. M. Pernice and H. F. Walker. Nitsol: A newton iterative solver for nonlinear systems. *SIAM J. Sci. Comput.*, 19(1):302–318, 1998.
60. R. Peyret. *Spectral Methods for Incompressible Viscous Flow*. New York, Springer, 2002.
61. R. Peyret and T. Taylor. *Computational methods for fluid flows*. New York, Springer, 1983.
62. C D Pruett, T B Gatski, C E Grosch, and W D Thacker. The temporally filtered navier–stokes equations: properties of the residual stress. *Physics of Fluids*, 15(8):2127–2140, 2003.
63. C D Pruett, B C Thomas, C E Grosch, and T B Gatski. A temporal approximate deconvolution model for large-eddy simulation. *Physics of Fluids*, 18(2):028104, 2006.
64. C. Yang R. B. Lehoucq, D. C. Sorensen. Arpack user's guide: Solution of large scale eigenvalue problems with implicitly restarted arnoldi methods. Technical Note, 1997.
65. U Rasthofer, F Wermelinger, P Hadijdoukas, and P Koumoutsakos. Large scale simulation of cloud cavitation collapse. *Procedia Computer Science*, 108:1763–1772, 2017.
66. Florent Renac. Improvement of the recursive projection method for linear iterative scheme stabilization based on an approximate eigenvalue problem. *Journal of Computational Physics*, 230(14):5739–5752, 2011.
67. F Richez, M Leguille, and O Marquet. Selective frequency damping method for steady rans solutions of turbulent separated flows around an airfoil at stall. *Computers & Fluids*, 132(Supplement C):51 – 61, 2016.
68. Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.
69. 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):024803, 2014.
70. L Shaabani-Ardali, D Sipp, and L Lesshafft. Time-delayed feedback technique for suppressing instabilities in time-periodic flow. *Physical Review Fluids*, 2(11):113904, 2017.
71. G M Shroff and H B Keller. Stabilization of unstable procedures: the recursive projection method. *SIAM Journal on numerical analysis*, 30(4):1099–1120, 1993.
72. D. Sipp and A. Lebedev. Global stability of base and mean flows: a general approach and its applications to cylinder and open cavity flows. *J. Fluid Mech.*, 593:333–358, 2007.
73. D C Sorensen. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, 13:357–385, 1992.
74. K. Stewarson and J. T. Stuart. A non-linear instability theory for a wave system in plane poiseuille flow. *J. Fluid Mech.*, 48:529–545, 1971.
75. G W Stewart. A Krylov-Schur algorithm for large eigenproblems. *SIAM J. Matrix Anal. Appl.*, 23:601–614, 2001.
76. J. Stoer and Bulirsch. *Introduction to Numerical Analysis*. Texts in Applied Mathematics, No 12. Springer-Verlag, Third edition, 2002.
77. V. Theofilis. Advances in global linear instability analysis of nonparallel and three-dimensional flows. *Progress in Aerospace Sciences*, 39:249–315, 2003.
78. V. Theofilis. Global linear instability. *Ann. Rev. Fluid Mech.*, 43:319352, 2011.
79. Laurette S Tuckerman and Dwight Barkley. Bifurcation analysis for timesteppers. In *Numerical methods for bifurcation problems and large-scale dynamical systems*, pages 453–466. Springer, 2000.