A practical review on linear and nonlinear global approaches to flow instabilities

D. Fabre$^{(a)}$, **V. Citro**$^{(b,a)}$, **D. Ferreira Sabino**$^{(a)}$, **P. Bonnefis**$^{(a)}$, **J. Sierra** $^{(a)}$, **F. Giannetti**$^{(b)}$ **& M. Pigou**$^{(a)}$
$^{(a)}$ Institut de Mécanique des fluides de Toulouse (IMFT), University of Toulouse
$^{(b)}$ Dipartimento di Ingegneria (DIIN), Universitá di Salerno.

This paper aims at reviewing linear and nonlinear approaches to study the stability of fluid flows. We provide a concise but self-contained exposition of the main concepts and specific numerical methods designed for global stability studies, including the classical linear stability analysis, the adjoint-based sensitivity and the most recent nonlinear developments. Regarding numerical implementation, a number of ideas making resolution particularly efficient are discussed, including mesh adaptation, simple shift-invert strategy instead of the classical Arnoldi algorithm, and a simplification of the recent nonlinear self-consistent approach proposed by Mantič-Lugo et al. (2014). An open-source software implementing all the concepts discussed in the present paper is provided. The software is demonstrated for the reference case of the two-dimensional flow around a circular cylinder, in both incompressible and compressible cases, but is easily customisable to a variety of other flow configurations or flow equations.

1 Introduction

The concept of stability bears on the response of a system to small perturbations of its state. If the generic disturbance grows in time, the system is unstable. The concept of stability can be simply formulated for a system of Ordinary Differential Equations (ODE). Such systems can be at equilibrium, where the state does not depend on time, or can present a periodic state, with all components returning to the same values, after every period.

The stability of fluid flows usually depends on the value of a given parameter. A bifurcation occurs when a critical value is reached and the original solution becomes linearly unstable, the system then starts evolving towards a new state, either steady or unsteady. In the second part of the 19th century, specific analytical and numerical methods have emerged to study these bifurcations and have continuously evolved up to the present days. A crucial point, that drove the development in this field, is the availability of significantly increasing and improving computing resources. Initially, the linear stability theory focused on fluid flows that are homogeneous in two spatial directions, *e.g.* plane Poiseuille flow [1]. In this case, thanks to a Fourier decomposition in the two homogeneous directions, it is possible to reduce the stability problem to a one-dimensional problem, an approach usually referred to as *local stability approach*. On the other hand, when there are at least two spatial variables, the class of methods suited to solve such problems are generally called *global stability approaches*. A classical example of such behaviour in fluid dynamics is the instability occurring in the wake of a circular cylinder. At low Reynolds number (precisely for $Re < 46.7$) the flow is steady and symmetric, but for larger values of Re a global instability arises in the flow field leading to the well-known von Kármán vortex alley. This flow configuration has served as a benchmark in the development of this class of methods. If one is only interested in predicting whether a flow is stable, it is sufficient to conduct a *linear stability analysis* which is the fundamental brick of global stability approaches. Beyond this simple question, in the past two decades, a number of extensions have been developed and popularized. *Adjoint methods* are an important extension [2], [3]; they can give insight into the sensitivity of the flow to intrinsic or extrinsic contributions. *Nonlinear stability approaches* [4] [5] have also been developed in order to extend the range of applicability of the numerical methods towards large amplitude perturbations.

The objective of the present work is to contribute to the popularization of such methods in two ways:

- First, we give a concise but self-contained exposition of the main concepts and specific numerical methods pertaining to global stability, including basic linear stability, adjoint-based sensitivity, as well as the most recent nonlinear developments.
- Secondly we offer an open-source and user-friendly software called StabFem [1] to perform such calculations. The software combines programs written in both FreeFem++ [6] and Octave/Matlab languages. FreeFem++ is used to generate and adapt meshes and to solve the various linear problems arising in the analysis. Octave/Matlab is used as a driver/wrapper to monitor the computations, perform the required loops

[1] The StabFem software may be obtained at the following url: https://www.gitlab.com/stabfem/StabFem.

over parameters, and post-process the results. The software is developed as a collaborative project, it aims a multi-platform support and is easily customizable to a variety of cases.

In the present paper the concepts are introduced and the software is demonstrated for the reference case of the incompressible, two-dimensional flow around a cylinder, but the software is easily customizable to a variety of other situations (free-surface, three-dimensional, etc..). We show also the application to compressible flows for the same geometrical configuration.

Although we do not claim inventing any radically new method, our exposition and implementation contains a number of novelties making the computation particularly efficient in terms of computational time and memory[2]. The most notable originalities are the systematic use of mesh adaptation (§2 and 3), the use of simple shift-invert instead of Arnoldi (§3), and a reformulation and simplification of the self-consistent approach of Mantič-Lugo in the framework of the Harmonic Balance formalism (§4).

2 Linear stability analysis: equations and methods

2.1 Computing a base-flow with Newton iteration

In this part, we expose the main concepts and methods of linear stability analysis for the case of a incompressible flow. The extension to compressible flow will be considered in section **??**.

Navier–Stokes equations and weak form We start from the general problem of a flow field $[\mathbf{u},p]$ satisfying the incompressible Navier–Stokes equations on a domain Ω,

$$\partial_t[\mathbf{u},p] = \mathcal{NS}([\mathbf{u},p]) \equiv -\mathbf{u}\cdot\nabla\mathbf{u} - \nabla p + \frac{2}{Re}\nabla\cdot \mathsf{D}(\mathbf{u}), \quad (1)$$

$$\nabla\cdot\mathbf{u} = 0, \quad (2)$$

with suitable boundary conditions on the frontier $\partial\Omega$ of the domain. Here $\mathsf{D}(\mathbf{u})$ is the rate-of-strain tensor defined as

$$\mathsf{D}(\mathbf{u}) = 1/2\left(\nabla\mathbf{u} + \nabla^T\mathbf{u}\right).$$

In the framework of finite element methods, we need to write the equations in weak form. Prior to this we define a scalar product as follows, for both scalar or vectorial quantities

$\langle\phi_1,\phi_2\rangle$:

$$\langle\phi_1,\phi_2\rangle = \int_\Omega \overline{\phi}_1\cdot\phi_2 \, d\Omega.$$

The weak form of the Navier–Stokes equations is readily defined by introducing test functions $[\mathbf{v},q]$ associated with momentum and continuity equations, and by integrating over the domain[3]

$$\forall[\mathbf{v},q], \quad \partial_t\langle\mathbf{v},\mathbf{u}\rangle = \big\langle\mathbf{v},\mathcal{NS}(\mathbf{u},p)\big\rangle + \langle q,\nabla\cdot\mathbf{u}\rangle. \quad (3)$$

Newton iteration We look for a steady base-flow $[\mathbf{u}_b,p_b]$ satisfying the steady Navier–Stokes equations, *i.e.* $\mathcal{NS}(\mathbf{u}_b,p_b)=0$. Suppose that we have a *guess* for the base flow $[\mathbf{u}_b^g,p_b^g]$ which almost satisfies the equations. We look for a better approximation under the form

$$[\mathbf{u}_b,p_b] = [\mathbf{u}_b^g,p_b^g] + [\delta\mathbf{u}_b,\delta p_b]. \quad (4)$$

Injecting (4) into the weak form (3) of the Navier–Stokes equations and linearizing leads to $\mathcal{NS}(\mathbf{u}_b^g,p_b^g) + \mathcal{LNS}_{\mathbf{u}_b^g}(\delta\mathbf{u}_b,\delta p_b) = 0$, which can also be written in weak form as:

$$\begin{aligned}&\big\langle\mathbf{v},\mathcal{NS}(\mathbf{u}_b^g,p_b^g)\big\rangle + \langle q,\nabla\cdot\mathbf{u}_b^g\rangle\\ &+\Big\langle\mathbf{v},\mathcal{LNS}_{\mathbf{u}_b^g}(\delta\mathbf{u}_b,\delta p_b)\Big\rangle + \langle q,\nabla\cdot\delta\mathbf{u}_b\rangle = 0,\end{aligned} \quad (5)$$

where $\mathcal{LNS}$ is the linearised Navier–Stokes operator, defined by its action on a flow field $[\mathbf{u},p]$ as follows

$$\mathcal{LNS}_{\mathbf{U}}(\mathbf{u},p) = -\mathcal{C}(\mathbf{U},\mathbf{u}) - \nabla p + \frac{2}{Re}\nabla\cdot\mathsf{D}(\mathbf{u}), \quad (6)$$

and $\mathcal{C}$ is the convection operator defined by

$$\mathcal{C}(\mathbf{U},\mathbf{u}) = (\mathbf{U}\cdot\nabla)\,\mathbf{u} + (\mathbf{u}\cdot\nabla)\,\mathbf{U}. \quad (7)$$

In the framework of finite elements, this problem can now be discretized by projecting upon a basis of Taylor–Hood $(u,v,p)\to(P2,P2,P1)$ elements. Noting δX the discretization of $[\delta\mathbf{u}_b,\delta p_b]$ this eventually leads to a matricial problem of the form $A\cdot\delta X = Y$. The procedure of Newton iteration is to solve iteratively this set of equations up to convergence. In our implementation, the algorithm for a 2D incompressible flow is written in the Freefem++ solver Newton_2D.edp (an extract of this code featuring the implementation of the Newton loop is given in appendix E). This FreeFem++ solver is wrapped by the Octave/Matlab driver SF_BaseFlow.m.

[2] All the figures contained in the present paper can be processed by launching a single Octave/Matlab script SCRIPT_CYLINDER_ALLFIGURES.m . On a MacbookPro (2018, 2.5Ghz, 16Go Ram) in single-core sequential execution, and using mesh $\mathbf{M}_2$ as described in appendix B, the execution time is 203 seconds for the linear analysis (section 3), and 413 seconds for the nonlinear analysis.

[3] In the simple presentation given here we have omitted the issue of boundary conditions. Details on how to incorporate boundary conditions in the weak formulation through integration by parts can be found in appendix D.

2.2 Linear stability

Direct eigenvalue problem We study the onset of the instability within the linear theory considering perturbations with eigenmode form:

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_b(\mathbf{x}) + \varepsilon\hat{\mathbf{u}}(\mathbf{x})e^{\lambda t}, \;\; p(\mathbf{x},t) = p_b(\mathbf{x}) + \varepsilon\hat{p}(\mathbf{x})e^{\lambda t}, \quad (8)$$

where $\lambda = \sigma + i\omega$ is the eigenvalue, σ the amplification rate, ω the oscillation rate, $\hat{\mathbf{u}}, \hat{p}$ the eigenmode (also called direct eigenmode, as opposed to the adjoint eigenmode introduced below), and ε a small parameter. The eigenmodes and eigenvalues are the solution of the following eigenproblem:

$$\lambda\hat{\mathbf{u}} = \mathcal{LNS}_{\mathbf{u}_b}(\hat{\mathbf{u}}, \hat{p}), \quad (9)$$

or, in weak form:

$$\lambda\left\langle \mathbf{v}, \hat{\mathbf{u}} \right\rangle = \left\langle \mathbf{v}, \mathcal{LNS}_{\mathbf{u}_b}(\hat{\mathbf{u}}, \hat{p}) \right\rangle + \left\langle q, \nabla\cdot\hat{\mathbf{u}} \right\rangle . \quad (10)$$

After discretization, we end up with an eigenvalue problem with the matricial form

$$\lambda \mathsf{B}\hat{X} = \mathsf{A}\hat{X}, \quad (11)$$

where A is the matrix resulting from the discretization of $\mathcal{LNS}_{\mathbf{u}_b}$, *i.e.* the same matrix appearing in the Newton computation of the base flow, and B is a weight matrix associated to the scalar product $\langle v, u \rangle = \int \overline{\mathbf{v}} \cdot \mathbf{u} \, d\Omega$.

Adjoint eigenvalue problem, structural sensitivity and wavemaker characterization Developed in the two past decades, the concept of adjoint eigenmodes has now become an unavoidable complement to the linear global stability approach. We here give a short summary of the definition and usefulness of this concept and refer the reader to Luchini & Bottaro [7] for further details.

First of all, the *adjoint linearised Navier–Stokes operator* $\mathcal{LNS}^{\dagger}$ is defined thanks to the following property:

$$\begin{aligned} \forall(\mathbf{u}, p; \mathbf{v}, q), \; & \left\langle \mathcal{LNS}^{\dagger}_{\mathbf{U}}(\mathbf{v}, q), \mathbf{u} \right\rangle + \left\langle \nabla\cdot\mathbf{v}, p \right\rangle \\ & = \left\langle \mathbf{v}, \mathcal{LNS}_{\mathbf{U}}(\mathbf{u}, p) \right\rangle + \left\langle q, \nabla\cdot\mathbf{u} \right\rangle . \end{aligned} \quad (12)$$

We can then define the adjoint eigenvalues and eigenmodes as the solution of the eigenvalue problem

$$\forall(\mathbf{u}, p), \quad \lambda^{\dagger}\left\langle \hat{\mathbf{v}}, \mathbf{u} \right\rangle = \left\langle \mathcal{LNS}^{\dagger}_{\mathbf{U}}(\hat{\mathbf{v}}, \hat{q}), \mathbf{u} \right\rangle + \left\langle \nabla\cdot\hat{\mathbf{v}}, p \right\rangle . \quad (13)$$

It can be shown [8] that the adjoint eigenvalues $\lambda_k^{\dagger}$ are the complex conjugates of the direct eigenvalues λ_k.

Although the concept of adjoint operator may sound complicated, the resolution of the adjoint problem using finite elements methods is actually extremely easy. Indeed, as the scalar product used in the definition of the weak formulation (3) and of the adjoint (13) is the same, the weak formulations of both problems happen to be identical when swapping the test and the unknown functions . Therefore, the matricial form of the discretized version of (13) is deduced from the one of the direct problem by a simple (Hermitian) transpose of the matrix:

$$\overline{\lambda}^{\dagger} B\hat{X}^{\dagger} = A^{T}\hat{X}^{\dagger}. \quad (14)$$

The adjoint field is a powerful tool for investigating problems such as receptivity, transient growth, control and sensitivity (see the reviews of [9], [10] and [7]). The simplest physical interpretation of an adjoint eigenmode is as follows: it corresponds to the initial condition which has maximum projection along the direction of the corresponding eigenmode. Thus, the adjoint of the most amplified mode corresponds to the optimal perturbation which will maximize the growth of energy in the limit of large time. In effect, one can prove that for $t \to \infty$ the asymptotic behaviour of a solution with initial condition $\mathbf{u}_i$ is given as:

$$\mathbf{u}(t) \approx \frac{\left\langle \hat{\mathbf{u}}^{\dagger}, \mathbf{u}_i \right\rangle}{\left\langle \hat{\mathbf{u}}^{\dagger}, \hat{\mathbf{u}} \right\rangle} e^{\lambda t}\hat{\mathbf{u}}.$$

The choice $\mathbf{u}_i = \hat{\mathbf{u}}^{\dagger}$ is the initial condition of norm unity which maximizes the first factor in this expression.

Investigating the structure of the adjoint eigenmodes gives access to a number of information about the instability mechanism which are not visible when examining only the direct eigenmodes. First, a simple examination of the dissimilarity between direct and adjoint eigenmodes is an indication of the non-normality of the linear operator, mostly associated to the convection term in open flows [9]. Secondly, in a more elaborated way, the product of both fields allows defining the so-called *structural sensitivity tensor* defined as

$$\mathbf{S}(\mathbf{x}) = \frac{\hat{\mathbf{u}}^{\dagger} \otimes \hat{\mathbf{u}}}{\left\langle \hat{\mathbf{u}}^{\dagger}, \hat{\mathbf{u}} \right\rangle}. \quad (15)$$

As formalized by [2] (following ideas previously proposed by [11]), this metrics quantifies how an eigenvalue is affected by the introduction of localized feedback of a flow perturbation onto itself. It thereby indicates the spatial regions where closed-loop control using a localized activator will be most efficient. Giannetti & Luchini (2007) conjectured that regions where such an extrinsic feedback induces the strongest change of the eigenvalue are also the most significant regions considering the intrinsic feedback mechanisms that underpins the genuine eigenmode dynamics. Following this idea, structural sensitivity has thus become a popular way to characherize the *wavemaker* *region* from where the instability mechanism originates. The structural sensitivity being a tensor, is is convenient to simply represent its norm S_w, often simply called 'the wavemaker':

$$S_w(\mathbf{x}) = ||\mathbf{S}(\mathbf{x})||_\infty \equiv \frac{||\hat{\mathbf{u}}^\dagger(\mathbf{x})||\;||\hat{\mathbf{u}}(\mathbf{x})||}{\langle \hat{\mathbf{u}}^\dagger, \hat{\mathbf{u}} \rangle}, \qquad (16)$$

Iterative methods for eigenvalue computations The numerical resolution of generalized eigenvalue problems such as $AX = \lambda BX$ (or its adjoint version (14)), can be performed using several methods. Direct methods to compute the whole spectrum are both costly prohibitive and their usefulness is disputable. A popular alternative is the use of iterative methods to compute a limited set of eigenvalues located in the vicinity of a *shift* value λ_{shift}. The simplest version of this method is the simple shift-invert iteration which consists in solving iteratively the system

$$X^n = (A - \lambda_{shift} B)^{-1} B X^{n-1}.$$

It is easy to show that this iterative procedure quickly asymptotes to $X^{n+1} \approx (\lambda^{*-1})^n \hat{X}$ where $\hat{X}$ is the eigenmode with largest λ^{*-1} (*i.e.* the one with eigenvalue λ closest to the shift).

When a good estimation of the eigenvalue is available, this method is very efficient and converges very rapidly. On the other hand, it can only provide a single eigenvalue. If we want to compute a larger number of eigenvalues, we can revert to a generalized version of iterative methods, called Arnoldi methods [12]. The shift-invert version of the Arnoldi method is in fact the most commonly used method of the current time and is at the basis of both the popular Octave/Matlab function `eigs` and the standard eigenvalue solver of FreeFem (*i.e.* ARPACK++). For the case of the 2D incompressible flow, resolution of the eigenvalue problem is done by the FreeFem++ solver Stab2D.edp; an excerpt of this solver featuring the implementation of the shift-invert algorithm is displayed in appendix E. The latter is wrapped by the Octave/Matlab driver SF_Stability.m which accepts a number of optional parameters. For instance, the selection between shift-invert and Arnoldi is made according to the parameter `nev` (*i.e.* number of requested eigenvalues) transmitted to the driver. (an excerpt of this code featuring the implementation of the Newton loop is given in appendix E).

2.3 Mesh adaptation procedure

As for any numerical method, a crucial aspect for numerical efficiency is the design of the mesh. The finite element method allows using unstructured mesh and hence to have a local adaptation. The most common procedure is to decompose the domain into several parts with different grid densities; for instance for the wake of a cylinder, we will design a near-wall region with very small size, a wake region with intermediate mesh size, and an outer region with large mesh size. The drawback of such mesh generation guidelines is that the design relies on an a priori expectation of the regions where gradients will be large.

In our implementation, we use an automated mesh adaptation method. The implementation relies on the AdaptMesh procedure of the FreeFem++ software. This procedure is detailed in ref. [13]. In short, the classical Delaunay-Voronoi algorithm produces a mesh with gridpoint distribution specified by a *Metric* matrix $\mathcal{M}$. The AdaptMesh algorithm consists in using as a metric the *Hessian* (second-order spatial derivatives) of an objective function u_h defined over the domain, *i.e.* $\mathcal{M} = \nabla\nabla u_h$. The precision can be controlled by specifying an objective value for the interpolation error of the function on the new mesh.

To build an optimal mesh for the base-flow calculation, the idea is to use as the objective function u_h the solution $\mathbf{u}_b$ itself, as computed on a previous mesh. The base flow is then recomputed on the adapted mesh, providing a better approximation of the solution. The procedure can be repeated over a few steps to ensure a right convergence.

The mesh generated in the previous way may not be optimal for the stability calculations as the structure of the eigenmode may be more complex than that of the base flow. To remedy with this, the idea is to subsequently adapt the mesh to both the mesh flow and the results of the stability calculation. This is easily done with FreeFem, as the AdaptMesh procedure can be used with several objective functions. We have experimented four different strategies. The first (D strategy) is to adapt the mesh to the base flow and the structure of the leading direct eigenmode. A second strategy (A) is to use the adjoint mode instead of the direct mode for mesh adaptation. However, as the sensitivity of the eigenvalue to perturbations of the operator (including discretization errors) is more closely linked to structural sensitivity concepts, it sounds a better idea to use the 'wavemaker' S_w to adapt the mesh, leading to the last experimented strategy, called S. Mesh adaptation using the scalar product of the direct and adjoint eigenmodes (called E-strategy) was also proposed in [14] (this quantity has also beed discussed by [15]). A detailed comparison of the four strategies is reported in appendix A. It is shown that for the case of a cylinder, all strategies give the same values for base-flow drag, eigenvalues and several properties of the nonlinear limit cycle (see sec. 4) with less than $0,3\%$ deviation, but that the A, E and especially the S strategy lead to significantly lower number of grid points compared to the D strategy. Note that a valid alternative (not yet available in the StabFem project) is offered by the Error Sensitivity to Refinement recently proposed by [16].

In our implementation, the whole process of mesh adaptation (including projection and recomputation of the base flow on the new mesh) is monitored using the Octave/Matlab driver SF_AdaptMesh.m ; the kind of adaptation (to base flow only (BF), or D, A, S , E strategies) is decided by the the nature of the objects given as input to this function. In the current implementation it is possible to adapt the mesh to as much as 8 fields of diverse nature (for instance, multiple eigenmodes along with their adjoint fields,

```
mesh = SF_Mesh('Mesh_Cylinder_Large.edp');
bf = SF_BaseFlow(mesh,'Re',1);
bf = SF_BaseFlow(bf,'Re',10);
bf = SF_BaseFlow(bf,'Re',60);
bf = SF_Adapt(bf,'Hmax',10,'InterpError',0.01);
[ev,em] = SF_Stability(bf,'shift',0.04+0.74i,'type','S')
bf = SF_Adapt(bf,em,'Hmax',10,'InterpError',0.01);
SF_Plot(bf,'p','contour','psi','clevels',[-.02 0 .2 1 2 5],'cstyle','patchdashedneg',...
      'xlim',[-1.5 4.5],'ylim',[0 3],'cbtitle','p','colormap','redblue','colorrange',...
      'centered','boundary','on','bdlabels',2,'bdcolors','k'); % figure 2

Re_Range = [2 : 2: 50]; Drag_tab = []; Lx_tab = [];
    for Re = Re_Range
        bf = SF_BaseFlow(bf,'Re',Re);
        Drag_tab = [Drag_tab,bf.Drag];
        Lx_tab = [Lx_tab,bf.Lx];
    end
plot(Re_Range,Drag_tab,'b+-'); %Figure 3a
plot(Re_Range,Lx_tab,'b+-'); % Figure 3b

Re_Range = [40 : 2: 100];lambda_branch=[];
bf=SF_BaseFlow(bf,'Re',40);
[ev,em] = SF_Stability(bf,'shift',-.03+.72i,'nev',1,'type','D');
    for Re = Re_Range
        bf = SF_BaseFlow(bf,'Re',Re);
        [ev,em] = SF_Stability(bf,'nev',1,'shift','cont','guess',em);
        lambda_branch = [lambda_branch ev];
    end
plot(Re_Range,real(lambda_branch),'b+-'); %Figure 4a
plot(Re_Range,imag(lambda_branch)/(2*pi),'b+-'); % Figure 4b
```

Fig. 1. Illustration of the usage of the StabFem software to produce an adapted mesh and study the base flow and the linear stability properties of the wake flow around a cylinder (extracted from script SCRIPT_CYLINDER_ALLFIGURES.m)

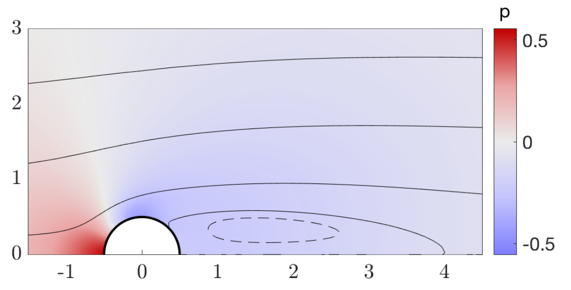

Fig. 2. Base flow for the flow over a cylinder at $Re = 60$. Pressure field (color levels) and streamlines (iso-levels of the streamfunction ψ).

harmonic-balance Fourier components, etc...)

3 Illustration for the wake of a cylinder (incompressible case)

Problem description Here, we consider the two-dimensional flow of an incompressible fluid of density ρ past a circular cylinder. All flow quantities are

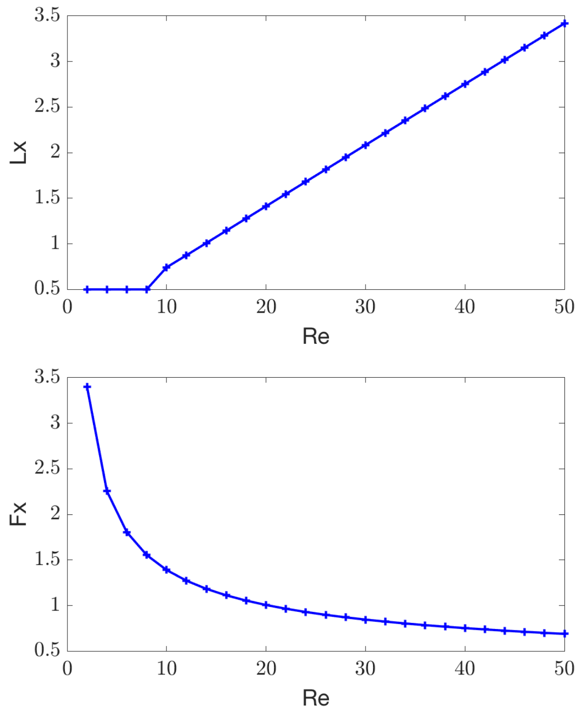

Fig. 3. Recirculation length L_x (a) and nondimensional drag F_x (b) of the base flow over a cylinder as function of Re.

normalized using the uniform incoming velocity U_∞ and the cylinder diameter D, which are the characteristic velocity and length scales used for the definition of the Reynolds number $Re = U_\infty D/\nu$. The origin of the Cartesian frame of reference is considered located on the cylinder axis, the x-axis is chosen to be parallel to the incoming free-stream velocity while the y-axis with the cross-stream velocity. The dimensions of the computational domain are the following: $-40 \leq x/D \leq 80$ and $0 \leq y/D \leq 40$ (boundary conditions and their implementation are detailed in appendix D). Note that we take advantage of the symmetry properties of the problem to only solve it on the positive y half-domain.

The hydrodynamic loads can be obtained by integrating the stress tensor over the cylinder surface. In particular, the hydrodynamic lift and drag forces read [4]

$$F_x = \mathcal{D}_{Re}(\mathbf{u},p) \equiv \int_{\Gamma_{cyl}} \left[-p\mathbf{n} + \frac{2}{Re}\mathsf{D}(\mathbf{u})\cdot\mathbf{n} \right] \cdot \mathbf{e_x} d\ell, \quad (17)$$

$$F_y = \mathcal{L}_{Re}(\mathbf{u},p) \equiv \int_{\Gamma_{cyl}} \left[-p\mathbf{n} + \frac{2}{Re}\mathsf{D}(\mathbf{u})\cdot\mathbf{n} \right] \cdot \mathbf{e_y} d\ell. \quad (18)$$

where Γ_{cyl} is the boundary of the cylinder.

Mesh adaptation procedure Let us now consider the Octave/Matlab code reported in figure 1. First we build an initial mesh (line 1), and compute base flow solutions for increasing values of the Reynolds number up to $Re = 60$ (lines 3-9). Then we perform the mesh adaptation with S strategy, as explained previously (line 13). The resulting mesh, depicted in figure 12, is used for the rest of the computations presented in this paper (except for plotting the structure of the direct eigenmode in figure 5(a) and computing the energy of the nonlinear perturbation displayed in figure 10(d) which adopted a finer mesh obtained with the strategy D). Appendix A presents additional tests regarding mesh convergence, and demonstrates that results obtained with the resulting mesh are reliable within 0.3% accuracy tolerance for the eigenvalue. It must be emphasized that the mesh generated through this adaptation process is very light, with only 2048 vertices, which is significantly less than reported in previous work (for instance, in their mesh convergence studies, [4] and [5] used respectively 190868 and 6731 vertices).

Base flow Having thus produced a convenient mesh, we can now illustrate the properties of the base flow as a function of Reynolds number. Figure 1 shows how to compute and plot with StabFem the two most commonly studied quantities, namely the recirculation length $Lx(Re)$, *i.e.* the location of the stagnation point at the rear of the recirculation region, and the drag force $F_x(Re)$. Note that the object `bf` is defined as a structure with fields `Fx` and `Lx`. The resulting plots are given in figure 3, and are in good agreement with known results for this classical problem. In particular, for low Reynolds, the recirculation $L_x(Re)$ is equal to 0.5 (which is the radius of the cylinder) indicating the absence of a recirculation region. The latter appears for $Re > 4.8$, in accordance with known results.

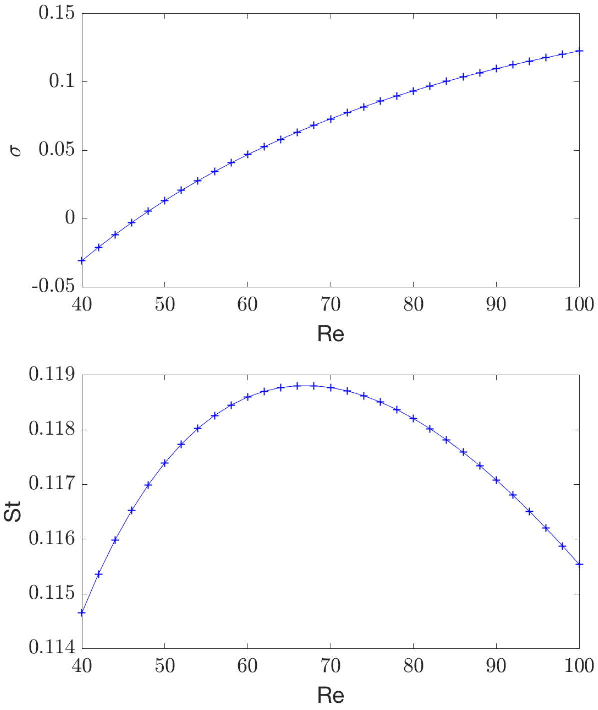

Fig. 4. Growth rate σ (a) and Strouhal number $St = \omega/2\pi$ (b) as function of the Reynolds number.

An illustration of the structure of the base flow is given in figure 2, for the case $Re = 60$. This figure is obtained using the Matlab/Octave function SF_Plot.m, which is built over the function ffpdeplot.m developed by M. Meister and also distributed on an open-source basis [5]. This function provides a number of possibilities for plotting data on unstructured meshes, including color plots, isolevels, streamlines, quiver plots, etc... The choice of parameters used here (lines 8-10 of the script in figure 1) allows plotting both the pressure field and selection of streamlines (through isocontours of the streamfunction ψ). This representation allows to easily visualize the extension of the recirculation region. In accordance with figure 3(b), for $Re = 60$ the recirculation length is $L_x \approx 4.07$.

Linear stability results We investigate the stability of the base flow field by performing a parametric study of the

[4] Note that F_x and F_y are actually nondimensional forces per unit length. For a cylinder of diameter D and length L (assuming $L \gg D$ so that the assumption of 2D flow makes sense), the corresponding dimensional forces are $F_x^* = \rho U_\infty^2 DLF_x$ and $F_y^* = \rho U_\infty^2 DLF_y$. Alternatively, one may characterize the forces through the drag and lift coefficients C_x and C_y. With the usual convention, The connection between the nondimensional forces and the force coefficients is $C_x = 2F_x$; $Cy = 2F_y$.

[5] https://github.com/samplemaker/freefem_matlab_octave_plot

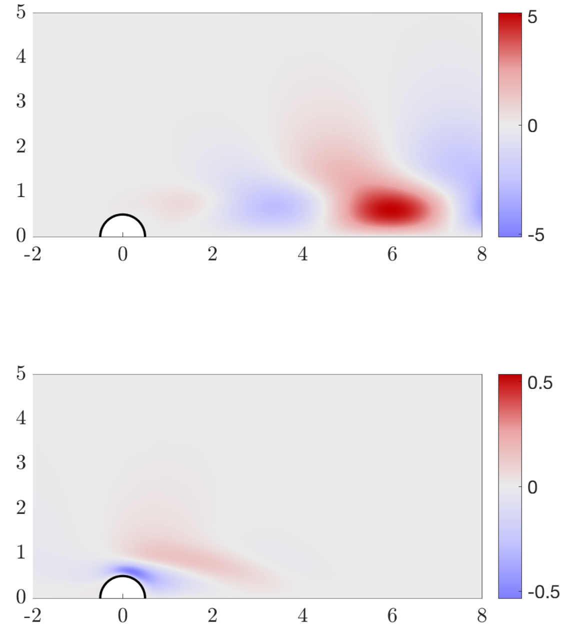

Fig. 5. Contour plot of the streamwise velocity component: (a) (Direct) Eigenmode; (b) Adjoint mode at $Re = 60$.

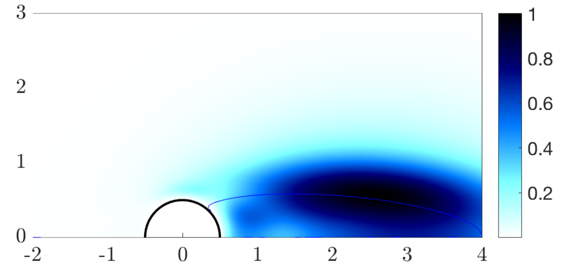

Fig. 6. Structural sensitivity S_w for the cylinder's wake at $Re = 60$. The red line represents the streamline bounding the recirculation region.

eigenproblem (9). In this way, we determine the critical Reynolds number Re_c from which the steady base flow first becomes unstable: to this end it is useful to remember that a flow state is linearly unstable when the real part of the leading eigenvalue, *i.e.* the growth rate, is positive. In our implementation, a parametric study is performed by looping over increasing Reynolds, corresponding to lines 21-30 of the script in figure 1. Note that at line 26 the options ’shift, ’cont’ ,’guess’, em transmitted to the driver SF_Stability allow to use an optimum value for the shift interpolated using the previous computations, and a guess value to initialize the shift-invert procedure corresponding to the previously computed eigenmode. Both these ideas very efficiently accelerate the calculations (the shift-invert procedure typically converges after only 2-3 iterations).

Figure 4 shows growth rate and the Strouhal number $St = a\omega/2\pi U_\infty$ as a function of the Reynolds number. It is easy to check that the critical Reynolds number is about 47 for the first mode. The associated direct eigenmode is depicted in figure 5. The spatial structure of this mode extends downstream of the bluff body and is characterized by streamwise extended spatial disturbances. On the other hand, the adjoint mode is highly localized near the cylinder on the upper (and lower) side of the body surface. We recall that the adjoint field provides useful information about the mechanism to flow receptivity to momentum forcing and mass injection. We note also that this receptivity decays rapidly both upstream and downstream of the bluff body.

The structural sensitivity S_w introduced in section 2.2 is displayed in figure 6. As discussed in section 2.2, both these quantities allow identifying the ’active’ flow regions responsible for the instability mechanism. The real quantity S_w indicates that the most active region roughly coincides with the recirculation bubble.

4 Extension : compressible flows

The main concepts of global linear stability analysis, exposed in section 2 for the simplest case of incompressible flows, are directly generalizable to more complex situations involving compressibility, fluid-structure interactions, deformable free surfaces, etc... Such more complex situations can be studied using the same class of numerical methods and the same methodology. One of the ambitions of the StabFem software if to provide a unified procedure to perform such stability studies, so that despite the fact that the underlying equations may be different, the corresponding Octave/Matlab scripts will be almost the same in all cases. In this section, we will illustrate this point for the case of a compressible flow, keeping the geometry as a 2D cylinder, and will reproduce the results of the recent study of [17].

4.1 Analysis and implementation

In the compressible case, the Navier-Stokes equations have to be formulated in terms of a state-vector $[\mathbf{u}, p, \rho, T]$ with larger dimension. The equations involve a larger number of terms but can be written in a compact way as follows :

$$\partial_t \mathcal{B}[\mathbf{u}, p, \rho, T] = \mathcal{NS}[\mathbf{u}, p, \rho, T] \tag{19}$$

Here $\mathcal{B}$ is a ”weight” operator specifying the coefficients in front of the time-derivatives of $[\mathbf{u}, p, \rho, T]$, and the operator $\mathcal{NS}[\mathbf{u}, p, \rho, T]$ specifies the time-evolution of $[\mathbf{u}, \rho, T]$ coming from the momentum, energy and mass balances as well as the state equation linking $[p, T, \rho]$. The detailed form of these equation is given in [17].

Starting with this form, the two main steps of the analysis, namely computation of a base-flow and eigenvalue

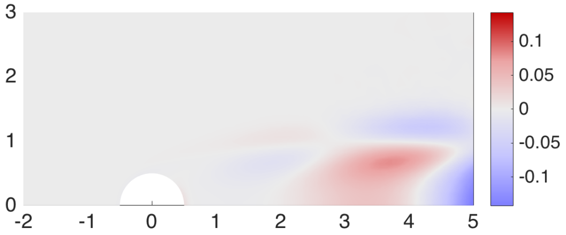

Fig. 7. Structure of the unstable eigenmode (streamwise velocity component) for the compressible flow around a cylinder $(Re = 150, M = 0.2)$.

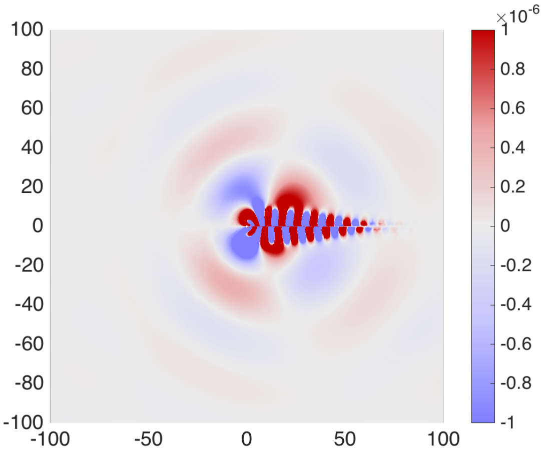

Fig. 8. Structure of the unstable eigenmode (pressure component) for the compressible flow around a cylinder $(Re = 150, M = 0.2)$. Upper half : real part ; lower half : imaginary part.

computation, can be done in the same way as explained in sec. 2. Namely :

- The base-flow $[\mathbf{u}_b, p_b, \rho_b, T_b]$ is the solution of $\mathcal{NS}[\mathbf{u}_b, p_b, \rho_b, T_b] = 0$ (eqs. $4(a-d)$ of [17]) which can readily be computed using Newton iteration.
- Eigenvalues λ and Eigenvalues $[\hat{\mathbf{u}}, \hat{p}, \hat{\rho}, \hat{T}]$ are solutions of the eigenvalue problem $\lambda \mathcal{B}[\hat{\mathbf{u}}, \hat{p}, \hat{\rho}, \hat{T}]^T = \mathcal{LNS}[\hat{\mathbf{u}}, \hat{p}, \hat{\rho}, \hat{T}]^T$ (eqs. $5(a-d)$ of [17]) which can be solved using either single-mode shift-invert iteration or multiple-mode Arnoldi iteration.

In our implementation, these calculations are done by the FreeFem++ solvers Newton2DComp.edp and Stab2DComp.edp, which are quite different from their incompressible counterparts. However these solvers are wrapped by the same generic drivers SF_BaseFlow.m and SF_Stability.m. The selection of which solver to use is made according to the parameters transmitted to them.

4.2 Example : 2D compressible flow around a cylinder

As an illustration, we consider the compressible flow around a 2D cylinder (same geometry as in previous section). The reader may find on the website of the project a script [6] reproducing the main results of the study of [17]. We restrict here to the illustration of the structure of the unstable eigenmode for $Re = 150; M = 0.2$. Figure 7 shows the axial velocity associated to the eigenmode in a domain centered on the body. Figure 8 displays the stucture of the pressure field on a much larger domain, allowing to detail the structure of the radiated acoustic field. The structure is in perfect agreement with the results of [17] (see $5(a)$ and 8 of this paper). The value of the corresponding eigenvalue is also perfectly reproduced, namely $\lambda = 0.15273 + 0.64482i$.

5 Nonlinear global stability approaches

In the past decade, efforts have been devoted to extend the range of validity of global approaches into the nonlinear regime for $Re > Re_c$, with the double objective of (i) describing the properties of the limit cycle reached after saturation and (ii) deriving amplitude equations describing the transient dynamics towards this cycle. The two main milestones in this direction are the *weakly nonlinear model* (WNL) of Sipp & Lebedev [4] and the *self-consistent model* (SC) of Mantič-Lugo et al. [5]. In this review, we will only address the first of the two questions, namely the description of the saturated cycle, and leave aside the question of transient dynamics. We will successively review the two aforementioned models, in a simplified formulation devoted to describe only the saturated cycle. We also provide a simple implementation of these two models. Finally, in line with the previous sections, this analysis is illustrated with results obtained for the wake of a cylinder.

In this part, to simplify the notations, we symbolically write the Navier–Stokes equations as $\partial_t \mathbf{u} = \mathcal{NS}(\mathbf{u})$, therefore dropping the systematic reference to the incompressibility constraint and associated pressure field. The same is done with the linearized operator $\mathcal{LNS}_{\mathbf{U}}(\mathbf{u})$.

5.1 General definitions in the nonlinear regime

In the nonlinear regime, the *base flow* introduced in the linear theory is no longer relevant, especially when the oscillation amplitudes become large. Instead, one may define a *mean flow* by using a time-average:

$$\mathbf{u}_m(\mathbf{x}) = \frac{1}{T} \int_0^T \mathbf{u}(\mathbf{x}, t) dt, \tag{20}$$

where $T = 2\pi/\omega$ is the period of the oscillation cycle. The difference between the instantaneous solution and the mean

[6] https://gitlab.com/stabfem/StabFem/blob/master/STABLE_CASES/compCYLINDER/SRIPT_Fani.m

flow is then called the *nonlinear perturbation*, defined as

$$\mathbf{u}'(\mathbf{x},t) = \mathbf{u}(\mathbf{x},t) - \mathbf{u}_m(\mathbf{x}). \tag{21}$$

A convenient measure of the unsteady part of the flow, which has been adopted in both the WNL and SC models, is the energy-amplitude, defined as the square-root of the total energy associated to the nonlinear perturbation:

$$A_E = \sqrt{\frac{1}{T}\int_0^T \left(\int_\Omega |\mathbf{u}'|^2\, dS\right) dt}. \tag{22}$$

In the next sections we will document the predictions of the WNL and SC models regarding the quantities A_E and L_x adopted in past studies. In addition, we will document two other quantities of practical interest: the Drag and Lift forces F_x and F_y exerted on the cylinder. Both are periodic functions of time and, owing to symmetry consideration, the drag contains only even harmonics and the lift only odd harmonics:

$$F_x = F_{x,0} + \sum_{n=1}^{\infty} \left(F_{x,2n,c}\cos(2n\omega t) + F_{x,2n,s}\sin(2n\omega t)\right), \tag{23}$$

$$\begin{aligned} F_y = \sum_{n=1}^{\infty} & \left(F_{y,2n-1,c}\cos((2n-1)\omega t)\right. \\ & \left. + F_{y,(2n-1),s}\sin((2n-1)\omega t)\right). \end{aligned} \tag{24}$$

In the sequel we will focus on the *mean drag* $F_{x,0}$ and on the fundamental components of the lift $F_{y,1,c}$ and $F_{y,1,s}$. These quantities are easily retrievable from a numerical simulation or an experiment, and we will show how they can be predicted from the nonlinear global approaches.

5.2 The weakly nonlinear model

We first review the weakly nonlinear model of Sipp & Lebedev [4], also discussed by Gallaire et al. [18]. The initial derivation of [4] makes use of a multiple scales method in order to obtain an amplitude equation. This complete analysis is reproduced in appendix B. In the present paragraph, we give a simplified derivation of this model restricted to the description of the periodic saturated cycle. The starting point can be taken as the following expansion of the velocity flow field:

$$\begin{aligned} \mathbf{u} = \mathbf{u}_{bc} &+ \varepsilon\left[A_{wnl}\hat{\mathbf{u}}e^{i(\omega_c+\varepsilon^2\omega_\varepsilon)t} + c.c.\right] \\ &+ \varepsilon^2\left[\mathbf{u}_\varepsilon + |A_{wnl}|^2\mathbf{u}_{2,0} + \left(A_{wnl}^2\mathbf{u}_{2,2}e^{2i(\omega_c+\varepsilon^2\omega_\varepsilon)t} + c.c.\right)\right] \\ &+ O(\varepsilon^3), \end{aligned} \tag{25}$$

This expansion is built as an asymptotic expansion in terms of the small parameter $\varepsilon = \sqrt{1/Re_c - 1/Re}$ corresponding to the distance to the critical Reynolds number. The zero-order term $\mathbf{u}_{bc}$ is the base flow at the threshold Re_c.

In the first-order term, $\hat{\mathbf{u}}$ is the neutral eigenmode at Re_c conveniently normalized (see discussion in appendix B), A_{wnl} an amplitude which can be assumed as real, ω_c the frequency predicted by the linear approach at $Re = Re_c$ and ω_ε a small deviation on the frequency.

The second-order term contains three contributions: $\mathbf{u}_\varepsilon$ is the modification to the base flow related to the increase of Re, $\mathbf{u}_{2,0}$ represents the nonlinear interaction of $\hat{\mathbf{u}}$ with its conjugate, $\mathbf{u}_{2,2}$ the nonlinear interaction of $\hat{\mathbf{u}}$ with itself. These three terms are computed as solutions of nonsingular linear systems (see eqs. (39), (40) and (41) in appendix B).

At $O(\varepsilon^3)$, compatibility conditions need to be enforced to ensure that the problem is correctly posed. These conditions lead to an *amplitude equation* which relates the amplitude A_{wnl} to three parameters Λ, ν_0 and ν_2 which depend uniquely on $\mathbf{u}_\varepsilon$, $\mathbf{u}_{2,0}$ and $\mathbf{u}_{2,2}$, respectively (see equations (43), (44) and (45) in appendix B). Restricting to the description of the limit cycle, the *amplitude equation* takes the form:

$$i\omega_\varepsilon A_{wnl} = \Lambda A_{wnl} - (\nu_0 + \nu_2)|A_{wnl}|^2 A_{wnl}. \tag{26}$$

The amplitude A_{wnl} and the correction to the frequency are then determined by considering the real and imaginary parts of this equation, leading to $A_{wnl} = \sqrt{\frac{\Lambda_r}{\nu_{0,r}+\nu_{2,r}}}$ and $\omega_\varepsilon = \Lambda_i - \Lambda_r\frac{\nu_{0,i}+\nu_{2,i}}{\nu_{0,r}+\nu_{2,r}}$ where the subscripts r and i represent the real and imaginary parts. Reintroducing the scaling, the amplitude $A = \varepsilon A_{wnl}$ and the frequency ω of the limit cycle are thus predicted as

$$A = \varepsilon A_{wnl} = \sqrt{\frac{\Lambda_r}{\nu_{0,r}+\nu_{2,r}}}\sqrt{\frac{1}{Re_c} - \frac{1}{Re}}, \tag{27}$$

$$\omega \equiv \omega^{(wnl)} = \omega_c + \left(\Lambda_i - \Lambda_r\frac{\nu_{0,i}+\nu_{2,i}}{\nu_{0,r}+\nu_{2,r}}\right)\left(\frac{1}{Re_c} - \frac{1}{Re}\right) \tag{28}$$

Finally, as specified above, we explain how the mean drag $F_{x,0}$ and fundamental components $(F_{y,1,c}, F_{y,1,s})$ of the oscillating lift can be predicted by the WNL approach. The mean drag can be obtained by $F_{x,0} = \mathcal{D}_{Re}(\mathbf{u}_m, p_m)$, where $\mathcal{D}$ is the drag operator defined in (17) and $[\mathbf{u}_m, p_m]$ is the *mean flow* which corresponds to the time-average of expansion (25), namely: $[\mathbf{u}_m, p_m] = [\mathbf{u}_{bc}, p_{bc}] + \varepsilon^2\left([\mathbf{u}_\varepsilon, p_\varepsilon] + A_{wnl}^2[\mathbf{u}_{2,0}, p_{2,0}]\right)$. Developing the terms as an asymptotic expansion leads to:

$$F_{x,0}(Re) \approx F_{x,0,Re_c} + F_{x,0,\varepsilon}\left(\frac{1}{Re_c} - \frac{1}{Re}\right) \tag{29}$$

```
 1  % WNL model
 2  [bf,em]=SF_FindThreshold(bf,em);  Omegac=imag(em.lambda);
 3  [ev,em] = SF_Stability(bf,'shift',1i*Omegac,'nev',1,'type','A');
 4  [wnl,meanflow,mode] = SF_WNL(bf,em,'Retest',47.,'Normalization','L');
 5
 6  epsilon2_WNL = -0.003:.0001:.005;
 7  Re_WNL = 1./(1/Rec-epsilon2_WNL);
 8  A_WNL = wnl.Aeps*real(sqrt(epsilon2_WNL));
 9
10  % HB model
11  Re_SC = [Rec 47 47.5 48 49 50 55 60 65 70 75 80 85 90 95 100];
12  Aenergy_SC  = [0];
13
14  for Re = Re_SC(2:end)
15      [meanflow,mode] = SF_HB1(meanflow,mode,'Re',Re);
16      Aenergy_SC  = [Aenergy_SC mode.AEnergy];
17  end
18
19  plot(Re_WNL,A_WNL,Re_SC,Aenergy_SC,'r+-'); % figure 7d.
```

Fig. 9. Illustration of the procedure for nonlinear calculations using StabFem (extract from script *SCRIPT_CYLINDER_ALLFIGURES.m*).

with $F_{x0,Re_c} = \mathcal{D}_{Re_c}(\mathbf{u}_{bc}, p_{bc})$, and $F_{x0,\varepsilon} = \mathcal{D}_{Re_c}(\mathbf{u}_\varepsilon, p_\varepsilon) - \mathcal{D}_1(\mathbf{u}_{bc}, 0) + A_{wnl}^2 \mathcal{D}_{Re_c}(\mathbf{u}_{2,0}, p_{2,0})$.

Similarly the required components of the lift force are obtained by applying the lift operator $\mathcal{L}$ defined in (18) to the order-one component of the expansion, leading to:

$$F_{y,1,c} - iF_{y,1,s} = 2A_{wnl}\mathcal{L}_{Re_c}(\hat{\mathbf{u}}, \hat{p})\sqrt{\frac{1}{Re_c} - \frac{1}{Re}}. \qquad (30)$$

Implementation and results for the cylinder Lines 1-8 of the script shown in figure 9 (extracted from the script SCRIPT_CYLINDER_ALLFIGURES.m) illustrate the sequence of commands to perform the weakly nonlinear study for the cylinder wake. On line 2, we first determine the instability threshold, and the corresponding base flow and eigenmode[7]. On line 3 we solve the adjoint problem required for some terms in the WNL formulation. On line 4, we then compute all the terms and coefficients of the WNL model. Lines 6-8 then process the results in a way they can be plotted.

Figure 10 shows the predictions of the WNL approach regarding the frequency (expressed as a Strouhal number through $St = \omega/2\pi$), mean drag, amplitude of the oscillating lift, and energy-amplitude. As discussed in [4] and [18], when compared to DNS results, this approach gives good prediction in the immediate vicinity of Re_c, but deviation appears very rapidly and disagreement is already large at $Re = 48$ (although, as discussed by [18] and also observed by [19], a convenient definition of ε improves the prediction). These aspects are justified by the perturbative nature of the WNL approach. The description of the limit cycle can be extended to larger Re with the approach presented next.

5.3 The Self-Consistent model reformulated in the Harmonic-Balance formalism

Analysis We now briefly review the self-consistent model as introduced by Mantič-Lugo et al. [5]. In their original exposition, the authors adopted a pseudo-eigenmode expansion of the flow as follows:

$$\mathbf{u} = \mathbf{u}_m + A_{sc}\left[\tilde{\mathbf{u}}_1 e^{\sigma_{sc}t + i\omega_{sc}t} + \overline{\tilde{\mathbf{u}}_1} e^{\sigma_{sc}t - i\omega_{sc}t}\right], \qquad (31)$$

where $\mathbf{u}_m$ is the *mean flow* as previously defined, $\tilde{\mathbf{u}}_1$ is a pseudo-eigenvector which is normalized by the condition $||\tilde{\mathbf{u}}_1|| = 1/\sqrt{2}$, $\overline{\tilde{\mathbf{u}}_1}$ is its complex conjugate, A_{sc} is an amplitude parameter directly related to the energy of the oscillating flow, and $\lambda_{sc} = \sigma_{sc} + i\omega_{sc}$ is a pseudo-eigenvalue which depends upon the parameter A_{sc}.

The original version of the model (discussed in detail in appendix C) allowed to obtain an amplitude equation predicting the instantaneous growth rate σ as function of the amplitude A_{sc}. However, if we are simply interested in the properties of the limit cycle, not the transient, we may recast the self-consistent model in a simpler way.

We thus start with a truncated Fourier decomposition of the limit cycle under the form:

$$\mathbf{u} = \mathbf{u}_m + \mathbf{u}_{1,c}\cos(\omega t) + \mathbf{u}_{1,s}\sin(\omega t), \qquad (32)$$

where $\mathbf{u}_{1,c}$ and $\mathbf{u}_{1,s}$ are two *real* fields describing the nonlinear perturbation at two instants separated by a

[7] This routine uses Newton iteration to directly compute the base flow, the eigenmode, the frequency and the critical Reynolds. The algorithm is very similar to the one presented for the HB, with an additional unknown (Re) and an additional constraint (normalization of the mode). The interested reader should reconstruct easily the whole procedure from the code provided.

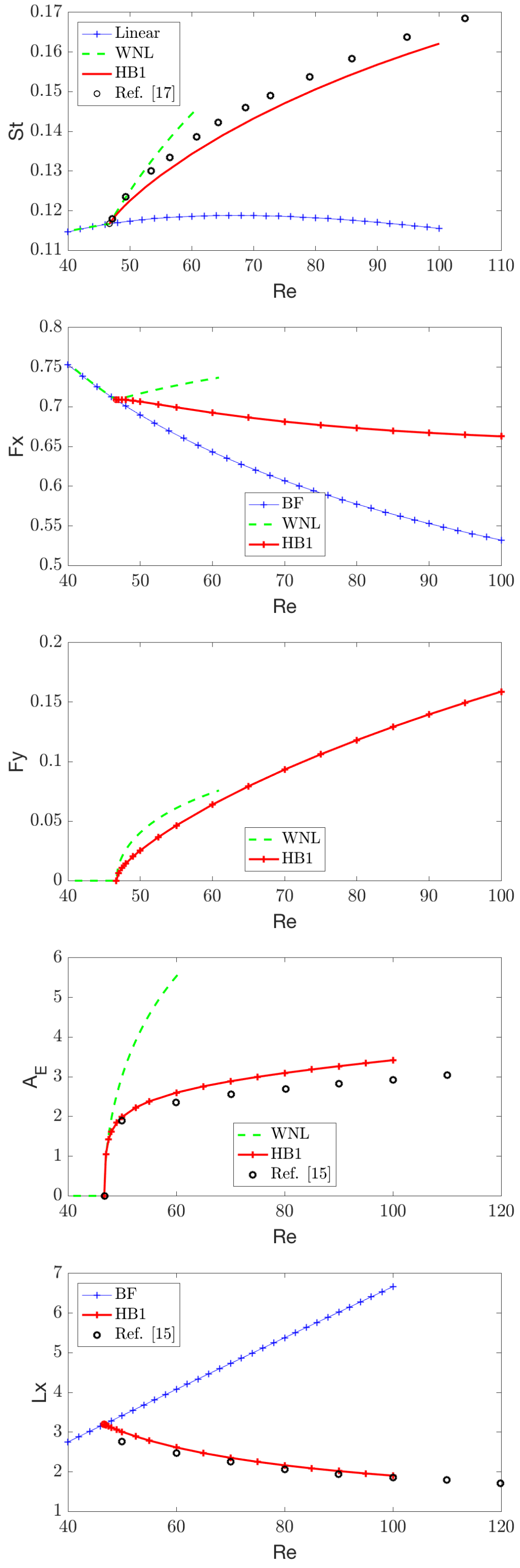

Fig. 10. Comparison between the weakly nonlinear results (WNL), the harmonic balance data (HB1), and baseflow/linear results: Strouhal number (a), Mean drag (b), amplitude of oscillating lift (c), Energy-amplitude of the nonlinear perturbation (d) and recirculation length of mean/base flows (e).

quarter-period of oscillation, and ω is the (real) oscillation frequency of the limit cycle (which is not known a priori).

Injecting this ansatz into the Navier–Stokes equations and taking the mean value and the first Fourier component leads to the following coupled equations:

$$\mathcal{NS}(\mathbf{u}_m) = \frac{\mathcal{C}(\mathbf{u}_{1,c},\mathbf{u}_{1,c})+\mathcal{C}(\mathbf{u}_{1,s},\mathbf{u}_{1,s})}{4}, \qquad (33a)$$

$$\omega\mathbf{u}_{1,s} = \mathcal{LNS}_{\mathbf{u}_m}(\mathbf{u}_{1,c}), \qquad (33b)$$

$$-\omega\mathbf{u}_{1,c} = \mathcal{LNS}_{\mathbf{u}_m}(\mathbf{u}_{1,s}). \qquad (33c)$$

Note that this system of equations contains three unknown fields (discretized on the Taylor–Hood basis) plus an extra scalar unknown, namely the frequency ω. In order to solve this coupled problem, we thus need an extra scalar equation. The latter is provided by fixing the *phase* of the cycle. Several choices are possible, but a convenient one is to decide that the instant $t = 0$ corresponds to a maximum of the lift force. This condition reads:

$$\mathcal{L}_{Re}(\mathbf{u}_{1s}, p_{1s}) = 0, \qquad (34)$$

so that the Fourier decomposition of the lift force (24) will contain only a cosine term, namely

$$F_y \approx F_{y,1,c}\cos\omega t, \quad \text{with} \quad F_{y,1,c} = \mathcal{L}_{Re}(\mathbf{u}_{1c}, p_{1c}). \qquad (35)$$

Note that the amplitude A_{sc}, which was considered as a key parameter in the original model of [5], does not appear in the simplified version discussed here. The connection between our parametrization and that of Mantič-Lugo et al. is as follows:

$$(\mathbf{u}_{1,c} - i\mathbf{u}_{1,s}) = 2A_{sc}\tilde{\mathbf{u}}_1. \qquad (36)$$

Considering their normalization choice, the parameter A_{sc} of [5] can be easily deduced from our results as follows: $A_{sc} = \sqrt{2\int_\Omega(|\mathbf{u}_{1,c}|^2+|\mathbf{u}_{1,s}|^2)dS}$.

SC/HB1 model: a direct algorithm When reformulated in this way, the nonlinear problem addressed by the self-consistent model actually reduces to the restriction to order one of a more general class of methods called harmonic balance methods (HB). One of the first applications of these methods in the domain of fluid mechanics has been done by Carte and Dušek [20,21]. The development of such methods, or related ones such as the time-spectral method ([22]) and the search for efficient numerical implementation to solve such problems is currently an active field of development in the fluid dynamics community. When considering HB development with a large order of terms, direct resolution by inversion of the full matrix is not achievable. Methods including iterative solvers and efficient preconditioning are currently under development (O. Marquet, personal communication).

On the other hand, when restricting as here to a truncation to order one (HB1 method), the use of adapted meshes makes the building and inversion of the matrix accessible on standard computers. We thus propose here a simple algorithm based on a direct resolution using a Newton iteration, just as explained for the base flow in section 2. This method is proposed as an efficient alternative compared to the double-loop resolution procedure of [5].

The algorithm assumes that we know a *guess* for the full state-vector defined as follows:

$$[\mathbf{u}_m,\mathbf{u}_{1,c},\mathbf{u}_{1,s},\omega] = [\mathbf{u}_m^g,\mathbf{u}_{1,c}^g,\mathbf{u}_{1,s}^g,\omega^g] + [\delta\mathbf{u}_m,\delta\mathbf{u}_{1,c},\delta\mathbf{u}_{1,s},\delta\omega].$$

Injecting and developing up to linear order leads to the following equations:

$$\mathcal{NS}(\mathbf{u}_m^g) - \frac{1}{4}[\mathcal{C}(\mathbf{u}_{1,c}^g,\mathbf{u}_{1,c}^g)+\mathcal{C}(\mathbf{u}_{1,s}^g,\mathbf{u}_{1,s}^g)] + \mathcal{LNS}_{\mathbf{u}_m^g}(\delta\mathbf{u}_m) - \frac{1}{2}[\mathcal{C}(\mathbf{u}_{1,c}^g,\delta\mathbf{u}_{1,c})+\mathcal{C}(\mathbf{u}_{1,s}^g,\delta\mathbf{u}_{1,s})] = 0, \tag{37a}$$

$$\mathcal{LNS}_{\mathbf{u}_m^g}(\mathbf{u}_{1,c}^g) - \omega^g\mathbf{u}_{1,s}^g - \mathcal{C}(\delta\mathbf{u}_m,\mathbf{u}_{1,c}^g) + \mathcal{LNS}_{\mathbf{u}_m^g}(\delta\mathbf{u}_{1,c}) - \omega^g\delta\mathbf{u}_{1,s} - \delta\omega\mathbf{u}_{1,s}^g = 0, \tag{37b}$$

$$\mathcal{LNS}_{\mathbf{u}_m^g}(\mathbf{u}_{1,s}^g) + \omega^g\mathbf{u}_{1,c}^g - \mathcal{C}(\delta\mathbf{u}_m,\mathbf{u}_{1,s}^g) + \mathcal{LNS}_{\mathbf{u}_m^g}(\delta\mathbf{u}_{1,s}) + \omega^g\delta\mathbf{u}_{1,c} + \delta\omega\mathbf{u}_{1,c}^g = 0, \tag{37c}$$

$$F_y(\mathbf{u}_{1,s}^g) + F_y(\delta\mathbf{u}_{1,s}) = 0. \tag{37d}$$

After discretization, this leads to a linear problem of the form $AX = Y$ where X is the discretized version of the unknowns $[\delta\mathbf{u}_m,\delta\mathbf{u}_{1,c},\delta\mathbf{u}_{1,s},\delta\omega]$, and A is a matrix of dimension $3N_{dof}+1$, where N_{dof} is the dimension of the Taylor–Hood basis of finite elements describing each of the three $[\mathbf{u},p]$ fields. This system is solved iteratively as explained in section 2.1 for the determination of the base flow. Mesh convergence issues for the nonlinear calculations are discussed in appendix A. It is shown that the mesh $\mathbf{M}_2$ designed using 'S' strategy is sufficient to yield converged results within 0.3% accuracy for all quantities considered here, with the exception of the energy-amplitude A_E which requires the more refined mesh $\mathbf{M}_4$ obtained using 'D' strategy.[8]

In the proposed implementation, resolution of the problem is done by the FreeFem++ program HB1_2D.edp which is wrapped by the Octave/Matlab function SF_HB1.m. Note that the interested reader will also find within the sources of StabFem project a driver SF_HB2.m implementing the same direct resolution method for the order-2 truncated harmonic balance. Generalization to higher orders is planned for future developments of the project.

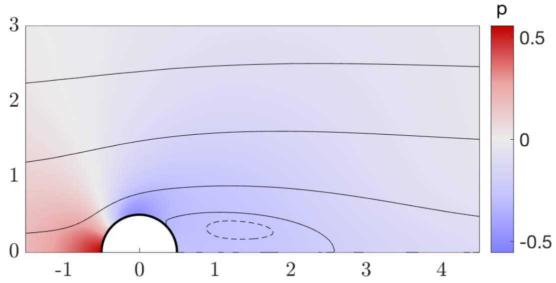

Fig. 11. Structure of the *mean flow* over a cylinder for $Re = 60$, as computed by the HB1 model. color levels: pressure ; streamlines.

Results for the cylinder's wake Lines 10-17 of the script shown in figure 9 explain how to use these programs to the SC/HB1 model for the wake of a cylinder in the range $Re \in [47-100]$. Note that a *guess* for the mean flow and the self-consistent mode for $Re = 47$, *i.e.* just above the threshold, has already been generated from the weakly nonlinear model at line 4 of the same script. This guess is used for the first step of the loop over Reynolds in lines 14-17. For next steps of the loop, continuation is done using the previous calculation results as guess.

Figure 11 illustrates the structure of the mean flow for $Re = 60$. As already identified by [5], the recirculation region associated with this mean flow is notably shorter than the one associated with the base flow (figure 2).

Figure 10 shows the comparison between the WNL (green) and HB1/SC (red) models for the quantities of interest identified in this paper, namely the Strouhal number, the mean drag, the maximum lift, the energy-amplitude, and the recirculation length associated to the mean flow. When relevant, results regarding the base flow and the linear approach are also displayed (blue) [9]. Concerning the quantities St, A_E and L_x, differences between HB1/SC, WNL models have already been commented in [5] and [18].

The present results are also compared with the available literature, namely with the experiments of [23] for the St number and with the DNS results of [5] for A_E and L_x. The predictions of the SC/HB1 model show an excellent agreement in the range $Re \in [Re_c, 100]$ with the references. However, we can remark that the predictions of the WNL model rapidly depart from the previous ones as soon as $Re - Re_c \gtrsim 1$.

6 Conclusion

The objective of this paper is twofold. First, we aimed at giving an up-to-date and comprehensive review on global stability approaches, both linear and nonlinear, including the most recent developments of the field. Secondly, we

[8] In terms of computational cost, on a standard laptop the program 9 requires approximatively 7 minutes using mesh $\mathbf{M}_2$ and 31 minutes using mesh $\mathbf{M}_4$.

[9] Figure 10 is obtained by line 19 of the script displayed in figure 9. The other figures are processed in a similar way. The entire set of commands to obtain all figures is provided in the script SCRIPT_CYLINDER_ALLFIGURES.m.

intended to provide an easy to use software performing all these computations from a single program. In accordance with this objective, all the figures presented in the paper can be produced by launching a single Octave/Matlab program available on the website of th project. This program handles all computation steps, from calls to FreeFem++ to figure generation.

Although the focus here was on the reference case of 2D incompressible and compressible flow around a cylinder, the *StabFem* software is designed to be easily customisable to a variety of other situations. The project is currently in constant development, and incorporates a growing number of other configurations. Under its present status, the project incorporates test-cases for the following classes of problems:

- Incompressible flows around 2D and axisymmetric blunt bodies of various geometries, including spheres and disks [19].
- Incompressible flows around 2D and 3D objects in free movement, including for instance the spring-mounted cylinder [24], and freely falling disks and spheres [25].
- Incompressible and compressible flow flow through apertures [26, 27],
- Compressible flow around 3D objects [28],
- Oscillations of hanging drops, sessile drops and liquid bridges [29],
- Rotating free surface flows [30].

Our ambition is to use the website of the project as a support to publish scripts allowing anyone to reproduce the main results of our past and future publications on such topics. Such scripts are already available concerning Refs. [19,27,29,30] and the list of available case will be growing in the coming months.The project is intended as collaborative, so anyone who wants to contribute is welcome !

A Mesh convergence: efficiency of mesh adaptation and effect of domain size

This appendix presents complementary results obtained using various mesh designs. All results can be obtained using the following program available on the StabFem website: SCRIPT_CYLINDER_MESHCONVERGENCE.m.

A.1 Efficiency of the mesh adaptation process

As discussed in section 3, the mesh adaptation proves to be an extremely efficient way to obtain significant and reliable results with very reasonable meshes. The objective of this paragraph is to demonstrate this point by comparing the efficiency of several mesh adaptation ways.

For this purpose, 6 meshes were generated. Mesh $\mathbf{M}_1$ was adapted on the *base flow* obtained for $Re = 60$. Mesh $\mathbf{M}_2$ was adapted to both the base flow and structural sensitivity (strategy S). Mesh $\mathbf{M}_2'$ was obtained by a subsequent refinement of mesh $\mathbf{M}_2$, splitting all triangles in four subtriangles, therefore doubling the effective density of the mesh. Mesh $\mathbf{M}_3$ was adapted to both the base flow and following [14] (strategy E). Meshes $\mathbf{M}_4$ and $\mathbf{M}_5$ were obtained by adapting to the structure of the direct eigenmode (strategy D) and of the adjoint eigenmode (strategy A). In each case, the adapt-mesh process was repeated twice to ensure a correct convergence.

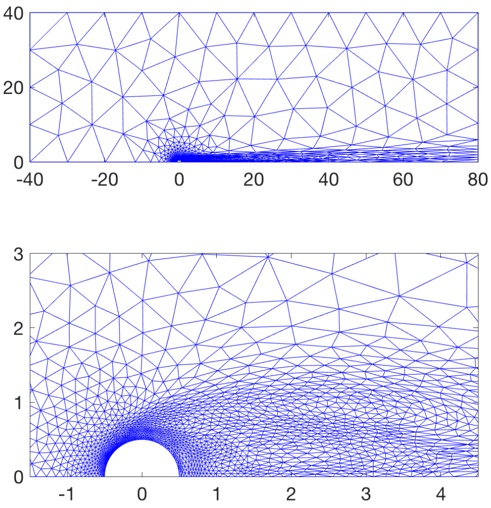

Fig. 12. Illustration of the stucture of mesh $\mathbf{M}_2$ (adapted to both the base flow and structural sensitivity).

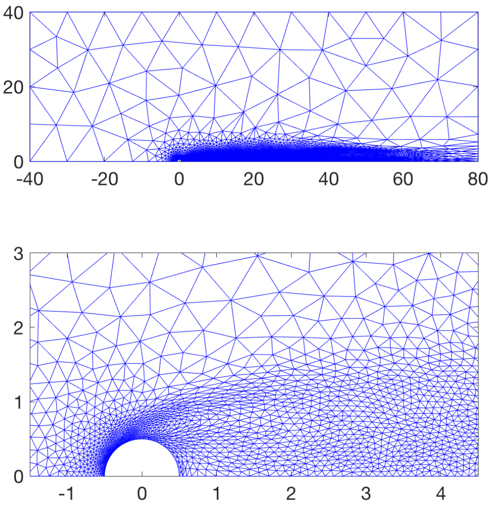

Fig. 13. Illustration of the stucture of mesh $\mathbf{M}_4$ (adapted to both the base flow and direct eigenmode).

Figures 12 and 13 illustrate the structure of meshes $\mathbf{M}_2$ and $\mathbf{M}_4$. It can be observed that these two mesh-adaptation strategies lead to meshes with comparable densities in the region close to the cylinder. On the other hand, when moving downwards in the wake, mesh $\mathbf{M}_2$ quickly gets rather coarse while mesh $\mathbf{M}_4$ maintains a significant density.

Table 1 gives numerical informations about the geometry of these meshes. In particular, we document the minimum and maximum cell size, as well as the cell size at four points A,B,C,D defined with by their coordinates

Mesh	N_p	N_{dof}	δ_{min}	δ_{max}	δ_A	δ_B	δ_C	δ_D
$\mathbf{M}_1$(Adapt on base flow)	1429	12545	0.0131	14.33	0.0259	0.514	0.819	1.067
$\mathbf{M}_2$(Adapt on sensitivity)	2038	17885	0.0155	14.17	0.02826	0.2046	0.3909	1.2014
$\mathbf{M}_2'$(Adapt on sensitivity, split)	7974	70651	0.00786	7.131	0.0104	0.0744	0.0975	0.614
$\mathbf{M}_3$(Adapt following [14])	3749	32655	0.01542	12.7109	0.09689	0.1236	0.2004	0.65775
$\mathbf{M}_4$(Adapt on mode)	12080	103564	0.00825	12.461	0.0229	0.143	0.0993	0.0934
$\mathbf{M}_5$(Adapt on adjoint)	3813	330715	0.0108	13.95	0.0176	0.114	0.127	1.177

Table 1. Description of meshes used for validation of mesh adaptation strategy: number of vertices N_p ; number of degrees of freedom of the P2-P2-P1 Taylor–Hood basis N_{dof} ; cell size (minimum and maximum value, and value at four characteristic point A,B,C,D as defined in the text).

Mesh	L_x	F_x	λ	$L_{x,HB1}$	$F_{x,HB1}$	ω_{HB1}	$F_{y,1,c}$	A_E
$\mathbf{M}_1$	4.0733	0.64310	$0.047056+0.74416i$	2.6073	0.69277	0.84334	0.1298	1.8301
$\mathbf{M}_2$	4.075	.0.64348	$0.046719+0.74489i$	2.6153	0.69246	0.84351	0.12789	1.7134
$\mathbf{M}_2'$	4.0772	0.64391	$0.04684+0.74511i$	2.6141	0.69338	0.84383	0.12799	2.048
$\mathbf{M}_3$	4.0749	0.64370	$0.04673+0.74538i$	2.6140	0.69317	0.84408	0.12808	2.064
$\mathbf{M}_4$	4.0748	0.64402	$0.04676+0.74502i$	2.6130	0.69034	0.84371	0.12809	2.5905
$\mathbf{M}_5$	4.0736	0.64470	$0.046782+0.74534i$	2.6142	0.69399	0.84390	0.12800	1.8234

Table 2. Results for mesh adaptation strategy ($Re = 60$): Base-flow characteristics L_x and F_x, linear eigenvalue λ, Nonlinear self-consistent model characteristics ω_{HB1}, $F_{y,1,c}$ and A_E. All the results can be optained using the Octave/Matlab script SCRIPT_CYLINDER_MESHCONVERGENCE.m.

as follows: $(x_A, y_A) = (0, 0.5)$ (within the boundary layer at the cylinder wall); $(x_B, y_B) = (2.5, 0.5)$ (in the region of maximum structural sensitivity); $(x_C, y_C) = (4, 0)$ (in the near wake); $(x_D, y_D) = (10, 0)$ (in the far wake). Results confirm that all meshes have similar densities in the near wake (cell size at points A,B,C are comparable) while $\mathbf{M}_4$ has maximum resolution in the far wake (in the vicinity of point D).

Table 2 compares the results obtained with the six meshes. For base-flow characteristics L_x and F_x, all values agree with a relative dispersion of less than 0.15%. This underlines that all adaptation strategies are successful to correctly compute the base flow. The performances of meshes for linear stability calculations can then be evaluated by comparing the eigenvalues. Meshes $\mathbf{M}_2$ to $\mathbf{M}_5$ all give values within less than 0.1% dispersion. The value obtained with mesh $\mathbf{M}_1$, which is not adapted to the eigenmode, is a bit farther from the others but still rather good. The table also displays results allowing to compare the performances of meshes for nonlinear self-consistent calculations. As for the frequency and the maximum lift, meshes $\mathbf{M}_2$ to $\mathbf{M}_5$ again give almost identical values within less than 0.1% dispersion. The dispersion concerning the energy-amplitude of the perturbation A_E displayed in the last column is however much larger. Only the mesh $\mathbf{M}_4$ is able to correctly compute this quantity, while all other meshes significantly underestimate it. This is not surprising since this quantity is an integral property which depends upon the structure of the nonlinear perturbation over the whole wake, not only the near-wake region.

From this study we can conclude that if we are only interested in predicting the frequency of the mode and the forces exerted on the cylinder (in both linear and nonlinear regimes), the strategies S (mesh $\mathbf{M}_2$) and E (mesh $\mathbf{M}_3$) are the most efficient and lead to a very light mesh (here only 2038 vertices for mesh $\mathbf{M}_2$). On the other hand, if we are interested in describing the structure of the perturbation in the whole domain (and being able to correctly evaluate its energy), mesh adaptation to the eigenmode structure is preferable. However this second strategy produces a much finer mesh (here 12080 point).

Mesh	N	L_x	F_x	λ	$L_{x,SC}$	$F_{x,SC}$	ω_{SC}	$F_{Y,SC}$
$\mathbf{M}_6[-20,40]x[0,20]$	1954	4.1097	0.65048	$0.048335+0.74989i$	2.6014	0.69988	0.85032	0.13219
$\mathbf{M}_7[-80,160]x[0,80]$	2292	4.0618	0.64046	$0.046081+0.74286i$	2.6188	0.68924	0.84042	0.12631
$\mathbf{M}_8$(”slip” conditions)	2026	4.0701	0.64590	$0.047017+0.74790i$	2.6127	0.69492	0.84613	0.12836
$\mathbf{M}_9$(”inlet” conditions)	2043	4.0677	0.64557	$0.047077+0.74781i$	2.6152	0.69429	0.84562	0.12771

Table 3. Comparison of the performances of several meshes with variable dimensions and different boundary conditions

A.2 Effect of domain size and boundary conditions

As already identified in several previous studies, the size of the domain and the type of boundary conditions applied at the boundaries have a notable impact on results. To illustrate this, we designed four additional meshes. All were obtained through adaptation to base flow and sensitivity, just as mesh $\mathbf{M}_2$. Meshes $\mathbf{M}_6$ and $\mathbf{M}_7$ are respectively twice smaller and twice larger than the reference one. Meshes $\mathbf{M}_8$ and $\mathbf{M}_9$ have the same dimension but the boundary condition at the lateral boundary Γ_{lat} differs. Unlike the reference case $\mathbf{M}_2$ which uses a no-stress boundary condition (identical to that applied at the outlet), mesh $\mathbf{M}_8$ uses a ”slip” condition $u_y = 0; \partial u_x/\partial y = 0$ while mesh $\mathbf{M}_9$ uses an even more restrictive constant-flow condition (the condition at the lateral boundary is $u_x = 1, u_y = 0$, just as for the inlet).

Even though the domain size of the reference case (namely $[-40,80] \times [0,40]$) may appear large, the table shows that confinement effects are still present. The quantity which appears to be the the most sensitive to domain size and/or boundary conditions is the imaginary part of the linear eigenvalue. Interestingly, the nonlinear SC results appears to be more robust with respect to confinement effects than linear ones. In effect, values for nonlinear frequency ω_{SC} and the maximum lift $F_{y,1,c}$ obtained with meshes $\mathbf{M}_2$, $\mathbf{M}_7$ $\mathbf{M}_8$ and $\mathbf{M}_9$ agree with less than 0.3% dispersion.

B Additional details of the weakly nonlinear approach

B.1 Derivation of the amplitude equation using multiple-scale approach

The initial derivation of [4] makes use of a multiple scale method in order to obtain an amplitude equation. The starting point can be taken as the following expansion of the velocity flow field:

$$\begin{aligned}\mathbf{u} &= \mathbf{u}_{bc} + \varepsilon\left[A_{wnl}(\tau)\hat{\mathbf{u}}e^{i\omega_c t} + c.c.\right] \\ &+ \varepsilon^2\left[\mathbf{u}_\varepsilon + |A_{wnl}(\tau)|^2\mathbf{u}_{2,0} + \left(A_{wnl}(\tau)^2\mathbf{u}_{2,2}e^{2i\omega_c t} + c.c.\right)\right] \\ &+ O(\varepsilon^3),\end{aligned} \tag{38}$$

Note that compared to the simplified version given in the main text by eq. (25), the amplitude A_{wnl} depends upon a slow time scale $\tau = \varepsilon^2 t$.

Substituting the expansion (38) into the Navier–Stokes equations (19) and grouping terms multiplied by the same power of ε, a hierarchy of equations is obtained. The order ε^0 gives directly the base flow at Re_c. The order ε^1 corresponds to the linear neutral eigenmode as computed in the first part of this article. The order ε^2 contains three terms respectively computed as the solutions of the following linear problems:

$$\mathcal{LNS}_{\mathbf{u}_{bc}}(\mathbf{u}_\varepsilon) - 2\nabla\cdot\mathsf{D}(\mathbf{u}_{bc}) = 0, \tag{39}$$

$$\mathcal{LNS}_{\mathbf{u}_{bc}}(\mathbf{u}_{2,0}) = \mathcal{C}(\hat{\mathbf{u}}, \overline{\hat{\mathbf{u}}}), \tag{40}$$

$$\mathcal{LNS}_{\mathbf{u}_{bc}}(\mathbf{u}_{2,2}) - 2i\omega_c\mathbf{u}_{2,2} = \frac{1}{2}\mathcal{C}(\hat{\mathbf{u}}, \hat{\mathbf{u}}). \tag{41}$$

Finally, compatibility conditions, given by the Fredholm’s alternative, are imposed at order ε^3 to remove the secular terms, leading to an amplitude equation, known as Stuart-Landau equation:

$$\frac{\partial A_{wnl}}{\partial\tau} = \Lambda A_{wnl} - (\nu_0 + \nu_2)|A_{wnl}|^2 A_{wnl}, \tag{42}$$

where coefficients Λ, ν_0 and ν_2 are given by

$$\Lambda = -\frac{\left\langle\hat{\mathbf{u}}^\dagger, \left(\mathcal{C}(\mathbf{u}_\varepsilon, \hat{\mathbf{u}}) + 2\nabla\cdot\mathsf{D}(\hat{\mathbf{u}})\right)\right\rangle}{\langle\hat{\mathbf{u}}^\dagger, \hat{\mathbf{u}}\rangle}, \tag{43}$$

$$\nu_0 = \frac{\left\langle\hat{\mathbf{u}}^\dagger, \mathcal{C}(\mathbf{u}_{20}, \hat{\mathbf{u}})\right\rangle}{\langle\hat{\mathbf{u}}^\dagger, \hat{\mathbf{u}}\rangle}, \tag{44}$$

$$\nu_2 = \frac{\left\langle\hat{\mathbf{u}}^\dagger, \mathcal{C}(\mathbf{u}_{22}, \overline{\hat{\mathbf{u}}})\right\rangle}{\langle\hat{\mathbf{u}}^\dagger, \hat{\mathbf{u}}\rangle}., \tag{45}$$

B.2 Normalisation of the eigenmode

A key issue in the weakly nonlinear expansion is that the definition of the amplitude depends upon a normalization choice of the eigenmodes. Several choices are possible. In the literature three possibilities have been used:

First, [4] normalized the eigenmode by assuming a specified value to the y-component of the velocity at one point, namely:

$$\hat{u}_y(1,0) = 0.4612. \tag{46}$$

Norm.	λ	ν_0	ν_2	ω_ε	$F_{x,0,\varepsilon}$	$F_{y,1,c}/\varepsilon$	$F_{y,1,s}/\varepsilon$	$\lvert F_{y,1}\rvert/\varepsilon$
[4]	$9.10988+3.28004i$	$9.3996-32.0289i$	$-0.305116-0.866118i$	36.2307	5.2848	0.0349973	−0.536113	0.537254
[18]	$9.10988+3.28004i$	$(0.488701-1.66524i)\times 10^{-3}$	$(-1.58635-4.50309i)\times 10^{-5}$	36.2307	5.2848	0.537254	0	0.537254
[31]	$9.10988+3.28004i$	$32.62-111.152i$	$-1.05886-3.00574i$	36.2307	5.2848	0.537254	0	0.537254

Table 4. Results of the WNL approach for three different choices of eigenmode normalization.

The advantage of this choice is that the coefficients ν_0 and ν_2 have the same order of magnitude as the coefficient Λ.

Secondly, [18] proposed the following normalization choice:

$$\int_\Omega |\hat{\mathbf{u}}|^2 d\mathbf{x} = \frac{1}{2}. \tag{47}$$

This directly leads to $|A|=A_E$, so this normalisation seems equivalent to the previous one.

Thirdly, following [31], another convenient choice is to normalize the eigenmode with its lift force:

$$\mathcal{D}_{Re_c}(\hat{\mathbf{u}},\hat{p}) = \frac{1}{2}. \tag{48}$$

The advantage of this choice is that the amplitude $|A|$ is then a direct measure of the fundamental lift. In effect, eq. (30) directlty leads to $F_{y,1,c}=|A|$, $F_{y,1,s}=0$.

In our implementation of the WNL approach, we allowed choosing the normalization convention, as seen in line 4 of figure 9. In table 4 we give the predictions of the WNL approach using the three normalization choices. We can note that the coefficients ν_0 and ν_2 strongly depend on the normalization choice. On the other hand, the frequency deviation ω_ε, the λ coefficient, and the term $F_{x,0,\varepsilon}$ related to the dependency of mean drag with deviation from the threshold are independent upon the normalization.

Considering the lift force, columns 7 and 8 of the table show that the different choices of normalization give different values for the coefficients $F_{1,y,c}$ and $F_{1,y,s}$. However, the sin-cos expansion of the lift force can be recast as $F_y = (F_{y,1,c}\cos\omega t + F_{y,1,s}\sin\omega t) = |F_{y,1}|\cos(\omega t+\varphi)$ with $|F_{y,1}|=\sqrt{F^2_{y,1,c}+F^2_{y,1,s}}$. The last column of the table confirms that the three possible normalization choices effectively lead to the same values of $|F_{y,1}|$.

C Additional details about the self-consistent method

The objective of this appendix is to provide additional details about the SC model in its original form as given by [5], and to explain the connection with the simpler version discussed in section 4.2. The full model is obtained by introducing the decomposition (31) into the Navier–Stokes equations, leading to:

$$\mathcal{NS}(\mathbf{u}_m) - A^2\mathcal{C}(\tilde{\mathbf{u}}_1,\overline{\tilde{\mathbf{u}}_1}) = 0, \tag{49a}$$

$$(\sigma_{sc}+i\omega_{sc})\tilde{\mathbf{u}}_1 = \mathcal{LNS}_{\mathbf{u}_m}(\tilde{\mathbf{u}}_1). \tag{49b}$$

Equation (49a) provides the mean flow field $\mathbf{u}_m$ while the pseudo-eigenpairs $(\lambda_{sc},\tilde{\mathbf{u}}_1)$ can be computed by solving the eigenvalue problem (49b). [5] initially proposed a resolution method involving two nested loops, which is advantageously replaced by the direct Newton resolution of section 4.3.

The self-consistent model has the following properties:

- For $A \ll 1$, it is equivalent to the linear eigenvalue problem (9), and the generalized eigenvalue coincides with the one predicted by linear stability: $\sigma_{SC}+i\omega_{SC} = \sigma_{lin}+i\omega_{lin}$.
- For $\sigma_{SC}=0$ (corresponding to a specific choice of the amplitude $A=A_{sc}$), the expansion (31) is equivalent to the Fourier expansion (**??**) taken as the starting point in the present paper.
- For $0<A<A_{sc}$, the resolution leads to a relation $\sigma_{SC}(A);\omega_{SC}(A)$ such that $0<\sigma_{SC}(A)<\sigma_{lin}$. Although in this case the expansion (31) cannot represent the flow for all t, Mantič-Lugo et al [5] argued that the relation between σ_{SC} and A can be used to build an amplitude equation which captures the transient approach to the limit cycle.

Note that in our numerical implementation, the programs can actually be used to solve the SC model in the general case (with $\sigma\neq 0$). This can be controlled by assigning a nonzero value to the optional parameter `sigma` of the SF_SelfConsistent.m Octave/Matlab function. The interested reader will find on the website of the StabFem project a program SCRIPT_CYLINDER_NONLINEAR.m which computes A as function of σ for $Re=100$, yielding identical results as displayed in figure 3 of [5].

D Details of the weak formulation

In the presentation of the numerical methods in sections 2.1 and 2.2, and introduction of the *weak form*, we have omitted an important point, namely the issue of boundary conditions. In this appendix we explain more rigorously how the weak formulation is obtained. We consider here the

```
// definition of linear system to be solved at each step of the iteration
 problem NewtonIter([dux,duy,dup],[vx,vy,q]) =
       int2d(th)( -2*nu*(D(du):D(v)) + q*div(du) + div(v)*dup - Conv(up,du,v) )
    + int2d(th)( -2*nu*(D(up):D(v)) + q*div(up) + div(v)*upp -.5*Conv(up,up,v) )
    + BoundaryconditionsBaseFlow(du,up,v);

// Newton loop
 while((res>tolerance)&(res<50)&(iter<=Nitermax))
{
    upx[]=ux[];
    NewtonIter ;
    ux[] = ux[]+dux[];
    res=sqrt(int2d(th)(dux^2+duy^2)/th.area);
    cout << "$$ Iter = "<< iter+1 << "  Re = " << Re << " ; res = " << res << endl;
    iter++;
} ;
```

Fig. 14. Illustration of the implementation of the Newton algorithm for base-flow computation (extract from FreeFem++ program Newton2D.edp).

full time-dependent nonlinear Navier–Stokes equations, but the treatment of the base-flow equations and the linearised equations is essentially the same.

Noting Γ the boundary of the numerical domain, the latter can be decomposed in five parts: $\Gamma = \Gamma_{in} \cup \Gamma_{cyl} \cup \Gamma_{axis} \cup \Gamma_{out} \cup \Gamma_{lat}$. Noting $\sigma = 2Re^{-1}\mathbf{D}(\mathbf{u}) - p\mathbf{1}$ the stress tensor, the relevant boundary conditions are as follows:

On Γ_{in} (inlet): $\mathbf{u} = \mathbf{e}_x$ (Dirichlet).
On Γ_{cyl} (surface of the cylinder): $\mathbf{u} = \mathbf{0}$ (Dirichlet).
On Γ_{out} (outlet): $\sigma \cdot \mathbf{n} = \mathbf{0}$ (Neumann).
On Γ_{lat} (lateral boundary): $\sigma \cdot \mathbf{n} = \mathbf{0}$ (Neumann).
On Γ_{axis} (symmetry plane): $u_y = 0$ and $\sigma_{xy} = 0$ (Mixed).

We will introduce the following notation for integrals along any portion of the boundary Γ_i of the product of two quantities ϕ_1, ϕ_2 (either scalar or vectorial):

$$\langle \phi_1, \phi_2 \rangle_{\Gamma_i} = \int_{\Gamma_i} \overline{\phi}_1 \cdot \phi_2 \, d\ell,$$

Instead of the simplified version (3), the more precise form of the weak formulation can be first written as follows:

$$\begin{aligned} \forall [\mathbf{v}, q], \quad & \partial_t \langle \mathbf{v}, \mathbf{u} \rangle = \left\langle \mathbf{v}, \mathcal{NS}(\mathbf{u}, p) \right\rangle + \langle q, \nabla \cdot \mathbf{u} \rangle \qquad (50) \\ & + \frac{1}{\varepsilon} \left(\langle \mathbf{v}, \mathbf{u} \rangle_{\Gamma_{cyl}} + \langle \mathbf{v}, (\mathbf{u} - \mathbf{e}_x) \rangle_{\Gamma_{in}} + \langle v_y, u_y \rangle_{\Gamma_{axis}} \right) \\ & + \langle \mathbf{v}, \sigma \cdot \mathbf{n} \rangle_{\Gamma_{out} \cup \Gamma_{lat} \cup \Gamma_{axis}} \end{aligned}$$

where $\varepsilon = 10^{-30}$ is a small parameter used to impose the Dirichlet boundary conditions by penalization. An integration by parts of the pressure gradient and viscous stress terms of the Navier–Stokes equation eventually leads to the weak form effectively used in the programs Newton2D.edp and Stab2D.edp:

$$\begin{aligned} \forall [\mathbf{v}, q], \quad & \partial_t \langle \mathbf{v}, \mathbf{u} \rangle = -\langle \mathbf{v}, \mathcal{C}(\mathbf{u}, \mathbf{u})/2 \rangle - 2Re^{-1} \langle \mathsf{D}(\mathbf{v}) : \mathsf{D}(\mathbf{u}) \rangle \\ & + \langle \nabla \cdot \mathbf{v}, p \rangle + \langle q, \nabla \cdot \mathbf{u} \rangle \qquad (51) \\ & + \frac{1}{\varepsilon} \left(\langle \mathbf{v}, \mathbf{u} \rangle_{\Gamma_{cyl}} + \langle \mathbf{v}, (\mathbf{u} - \mathbf{e}_x) \rangle_{\Gamma_{in}} + \langle v_y, u_y \rangle_{\Gamma_{axis}} \right) \end{aligned}$$

Note that the Neumann boundary conditions do not appear anymore thanks to the integration by parts.

E Numerical implementation in FreeFem++

In this appendix, we provide pieces of codes illustrating how the basic algorithms are implemented in the FreeFem++ solvers. These explanations will be useful both for readers who wish to use directly FreeFem++ solvers without using the overlayer of Octave/Matlab drivers provided by the StabFem software, and also for the reader who wants to understand the logics of the implementation and to customize the software to implement their own cases. The full version of the codes is available in the web repository of the StabFem project. A full documentation of the software is also in progress [10].

Figure 14 details the implementation of the Newton algorithm for base-flow computation, as implemented in the FreeFem++ solver Newton2D.edp which is a generic solver usable for the whole class of 2D incompressible problems. Note that the syntax makes use of macros D, div, Conv, resulting in a very similar to the weak formulation written in the previous appendix. The boundary conditions are also implemented using a macro BoundaryconditionsBaseFlow. To allow an easy customization, this macro is not defined in the generic solver Newton2D.edp but is reported in a file Macros_StabFem.idp regrouping case-dependant

[10] https://gitlab.com/stabfem/StabFem/blob/master/99_Documentation/MANUAL/main.pdf

```
// Definition of operators and assembly of matrix OP = NLS - sigma_s B
varf   LNSE ([ux,uy,up],[vx,vy,q]) =
int2d(th)( -2*nu*(D(u):D(v)) + up*div(v) + div(u)*q - Conv(u,Ub,v) - shift*(ux*vx+uy*vy) )
 + BoundaryconditionsStability(u,v,symmetry);
OP=LNSE(XXMh,XXMh, solver=sparsesolver);
varf brhs([ux,uy,p],[vx,vy,q]) = int2d(th)( (ux0*vx+uy0*vy) );

/// Iteration loop
        for (iter=0; ((err>errmax)&&(iter<itmax)); iter++)
        {
        complex[int] rhs= brhs(0,XXMh);
        complex[int] w = OP^-1*rhs;
        ux[] = w;
        complex XnXn1 = int2d(th)(ux0*ux+uy0*uy);
        complex Xn1Xn1 =  int2d(th)(ux*ux+uy*uy);
        complex GG = Xn1Xn1/XnXn1;
// at each step the inverse shifted eigenval is approximated by <X_{n+1}, Y> / <X_n, Y>
//      ( the choice Y = X_{n+1} is convenient but we could choose something else )
        lambda = shiftOP+1/GG;
        err = abs(lambda-lambda0);
        cout << "$$ Iteration " << iter+1 << " : " << endl;
        cout << "$$ Estimated eigenvalue lambda = " << lambda << endl;
        ux0[] = 1/sqrt(abs(Xn1Xn1))*ux[];
        lambda0 = lambda;
        };
```

Fig. 15. Illustration of the implementation of the shift-invert algorithm for single eigenmode computation (extract from FreeFem++ program Stab2D.edp).

macros (essentially boundary conditions and post-processing options).

Figure 15 details the implementation of the shift-invert algorithm for eigenvalue computation, as implemented in the generic solver Stab2D.edp for 2D incompressible flows. Here again, boundary conditions (which may differ from a case to an other within the generic class of 2D incompressible flows) are defined by a macro BoundaryconditionsStability which has to be defined in the Macros_StabFem.idp file.

We do not provide here the listing for the Newton resolution of the HB1 model, but the interested reader is encouraged to look at the program HB1_2D.edp on the project site.

References

[1] Drazin, P., and Reid, W., 2004. *Hydrodynamic Stability*. Cambridge University Press.

[2] Giannetti, F., and Luchini, P., 2007. "Structural sensitivity of the first instability of the cylinder wake". *J. Fluid Mech.,* **581**, pp. 167–197.

[3] Marquet, O., Sipp, D., and Jacquin, L., 2008. "Sensitivity analysis and passive control of cylinder flow". *J. Fluid Mech.,* **615**, pp. 221–252.

[4] Sipp, D., and Lebedev, A., 2007. "Global stability of base and mean-flows: a general approach and its applications to cylinder and open cavity flows". *J. Fluid Mech.,* **593**, pp. 333–358.

[5] Mantič-Lugo, V., Arratia, C., and Gallaire, F., 2014. "Self-consistent mean flow description of the nonlinear saturation of the vortex shedding in the cylinder wake". *Phys. Rev. Lett.,* **113**, p. 084501.

[6] Hecht, F., 2012. "New development in freefem++". *J. Numer. Math.,* **20**(3-4), pp. 251–265.

[7] Luchini, P., and Bottaro, A., 2014. "Adjoint equations in stability analysis". *Annu. Rev. Fluid Mech.,* **46**, pp. 493–517.

[8] Schmid, P. J., and Henningson, D. S., 2001. *Stability and Transition in Shear Flows*. Springer-Verlag, New York.

[9] Chomaz, J.-M., 2005. "Global instabilities in spatially developing flows: Non-normality and nonlinearity". *Annu. Rev. Fluid Mech.,* **37**, pp. 357–392.

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

[11] Hill, D., 1992. "A theoretical approach for analyzing the restabilization of wakes". In 30th Aerospace Sciences Meeting and Exhibit, p. 67.

[12] Arnoldi, W. E., 1951. "The principle of minimized iterations in the solution of the matrix eigenvalue problem". *Quart. Appl. Math.,* **9**, pp. 17–29.

[13] Hecht, F., 1998. "The mesh adapting software: bamg". *INRIA report*.

[14] Kouhi, M., Onate, E., and Mavriplis, D., 2015. "Adjoint-based adaptive finite element method for the compressible euler equations using finite calculus". *Aerospace Science and Technology,* **46**, pp. 422–435.

[15] Marquet, O., and Lesshafft, L., 2015. "Identifying the active flow regions that drive linear and nonlinear

instabilities". *https://arxiv.org/abs/1508.07620*.

[16] Luchini, P., Giannetti, F., and Citro, V., 2017. "Error sensitivity to refinement: a criterion for optimal grid adaptation". *Theoretical and Computational Fluid Dynamics,* **31**(5), Dec, pp. 595–605.

[17] Fani, A., Citro, V., Giannetti, F., and Auteri, F., 2018. "Computation of the bluff-body sound generation by a self-consistent mean flow formulation". *Physics of Fluids,* **30**(3).

[18] Gallaire, F., Boujo, E., Mantic-Lugo, V., Arratia, C., Thiria, B., and Meliga, P., 2016. "Pushing amplitude equations far from threshold: application to the supercritical hopf bifurcation in the cylinder wake". *Fluid. Dyn. Res.,* **48**, p. 061401.

[19] Tchoufag, J., Fabre, D., and Magnaudet, J., 2015. "Weakly nonlinear model with exact coefficients for the fluttering and spiraling motion of buoyancy-driven bodies". *Phys. Rev. Lett.,* **115**, p. 114501.

[20] Carte, G., Dušek, J., and Fraunié, P., 1995. "A Spectral Time Discretization for Flows with Dominant Periodicity". *J. Comput. Phys.,* **120**(2), pp. 171–183.

[21] Carte, G., Dušek, J., and Fraunié, P., 1996. "Numerical Simulation of the mechanisms governing the onset of the Bénard-Von Kármán instability". *Int. J. Numer. Methods Fluids,* **23**(8), pp. 753–785.

[22] Hall, K. C., Thomas, J. P., and Clark, W. S., 2002. "Computation of unsteady nonlinear flows in cascades using a harmonic balance technique". *AIAA Journal,* **40**(5), pp. 879–886.

[23] Williamson, C. H., 1988. "Defining a universal and continuous Strouhal–Reynolds number relationship for the laminar vortex shedding of a circular cylinder". *Phys. Fluids,* **31**(10), pp. 2742–2744.

[24] Navrose, N., and Mittal, S., 2016. "Lock-in in vortex-induced vibration". pp. 565–594.

[25] Tchoufag, J., Fabre, D., and Magnaudet, J., 2014. "Global linear stability analysis of the wake and path of buoyancy-driven disks and thin cylinders". *Journal of Fluid Mechanics,* **740**, pp. 278–311.

[26] Fabre, D., Bonnefis, P., Charru, F., Russo, S., Citro, V., Giannetti, F., and Luchini, P., 2014. "Application of global stability approaches to whistling jets and wind instruments". In Proceedings of the International Symposium on Musical Acoustics ISMA'14.

[27] Longobardi, R., Fabre, D., Bonnefis, P., Citro, V., Giannetti, F., and Luchini, P., 2018. "Studying sound production in the hole-one configuration using compressible and incompressible global stability analyses". In Symposium on Critical flow dynamics involving moving/deformable structures with design applications, IUTAM, ed.

[28] Meliga, P., Sipp, D., and Chomaz, J.-M., 2010. "Effect of compressibility on the global stability of axisymmetric wake flows". *Journal of Fluid Mechanics,* **660**, pp. 499–526.

[29] Chireux, V., Fabre, D., Risso, F., and Tordjeman, P., 2015. "Oscillations of a liquid bridge resulting from the coalescence of two droplets". *Phys. Fluids,* **27**, p. 062103.

[30] Mougel, J., Fabre, D., Lacaze, L., and Bohr, T., 2017. "On the instabilities of a potential vortex with a free surface". *Journal of Fluid Mechanics,* **824**, pp. 230–264.

[31] Fabre, D., Auguste, F., and Magnaudet, J., 2008. "Bifurcations and symmetry breaking in the wake of axisymmetric bodies". *Phys. Fluids*, p. 051702.