Tutorial on the flow past a circular cylinder using nekStab

Ricardo A S Frantz

March 1, 2023

1 Preparing the computational environment

1.1 macOS

To install brew[1], just paste the installation command in a new terminal shell. For changes to take effect, close and re-open your current shell. Then ensure you have the latest version of XCode Command Line Tools. **Do not download** the full 20 GB XCode from the App Store!

```
xcode-select --install
```

If you have problems with XCode follow this steps[2] update and install packages:

```
brew update && brew upgrade
brew install mpich gfortran wget git cmake htop vim
```

if you have problems try:

```
brew cleanup
brew doctor
```

or eventually replace the package `mpich` by `open-mpi`. To have python3 do:

```
python -m pip install -U pip
python -m pip install -U matplotlib scipy
```

1.2 Ubuntu (including Windows Subsystem for Linux)

In Ubuntu, instead of using brew, update and install packages using apt:

```
sudo apt update && apt upgrade
sudo apt -y install open-mpi libopenblas-dev cmake m4 htop vim build-essential
sudo apt autoremove
```

[1] https://brew.sh/
[2] https://gist.github.com/Justintime50/2349ff5e62555aa097acbf519bbc27af

Note that using *mpich* in WSL does not seem to work.

Also, to be able to use python to plot the scripts you need:

```
sudo apt install python3-scipy python3-matplotlib
```

you also should add to your *.bashrc*:

```
export DISPLAY=:0
```

1.3 Conda (experimental)

Install Anaconda from `https://www.anaconda.com/`, then create and activate a new environment:

```
conda create --name nek
conda activate nek
```

Install packages using *conda*:

```
conda install -c conda-forge gfortran openmpi
```

or eventually replace the package `openmpi` by `mpich`.

```
conda remove openmpi
conda install -c conda-forge mpich
```

You can later remove the environment:

```
conda deactivate
conda remove --name nek --all
conda env list
```

2 Cloning the repositories from GitHub

It is recommended you clone the repositories directly to your HOME folder. To make it faster we can copy only the latest revision. From a terminal window:

```
cd
git clone --depth=1 https://github.com/nekStab/nekStab.git

cd nekStab
git clone --depth=1 https://github.com/nekStab/Nek5000.git
```

If you are not able to clone and have to manually download the compressed repositories, you might encounter permission issues that can be fixed with:

```
chmod +x myscript.sh
```

2.1 Adding the sources

To automatically add the sources to your *.bashrc* (or equivalent), just run it (one time, une fois, uma vez!):

```
cd ~/nekStab
./cat_exports.sh
```

Close the terminal and re-open!

3 Compiling the case

Navigate to to the tutorial folder and compile the code:

```
cd ~/nekStab/tutorials/cylinder
makeneks 1cyl
```

the code might ask: *do you want to rebuild all 3rd party dependencies too?* press *Y* or *Enter*. The compiler works in parallel and should take a few seconds to complete:

```
############################################################
#                  Compilation successful!                 #
############################################################
```

You are good to proceed to the next steps!

3.1 Troubleshooting

If you receive the error **command not found: nekbmpi**, just do:

```
source ~/nekStab/path.sh
```

4 Using *vim*

You can edit the file in the next session using either a graphical text editor (such as Sublime Text, gedit, Atom, or Visual Studio Code) or a command-line text editor like Vim. Vim is a powerful and versatile text editor available on most Unix-like systems. Here's a basic guide on how to edit a file, save, and exit:

1. Open a file with Vim:

   ```
   vim filename.txt
   ```

   Replace filename.txt with the name of the file you want to edit. If the file doesn't exist, Vim will create a new one.

2. Vim starts in "normal mode" by default. To enter "insert mode" and begin editing the file, press i. You can now edit the file like you would in any other text editor.

3. Once you've made your changes, press Esc to return to normal mode.

4. To save your changes, type :w and press Enter. This writes your changes to the file.

5. To exit Vim, type :q and press Enter. If you've made changes and want to save and exit in one command, type :wq instead.

6. If you want to exit Vim without saving your changes, type :q! and press Enter. This will discard any changes made since the last save.

To summarize, the most common Vim commands are:

- i → Enter insert mode
- Esc → Return to normal mode
- :w → Save changes
- :q → Quit Vim
- :wq → Save changes and quit Vim
- :q! → Discard changes and quit Vim

5 Introduction

The flow past a circular cylinder is a classic problem in fluid dynamics and has been the subject of numerous studies over the years. The dynamics of the flow past a circular cylinder is complex and can exhibit a variety of interesting phenomena, including vortex shedding, flow separation, and turbulence.

When a fluid flows past a circular cylinder, it can either stay attached to the cylinder or separate from it, depending on the speed of the flow and the viscosity of the fluid. At low flow speeds, the fluid flows smoothly around the cylinder, and there is no separation. However, as the flow speed increases, the fluid begins to separate from the cylinder and form vortices on either side of the cylinder. These vortices are shed alternately from each side of the cylinder and produce a characteristic pattern of vortices in the wake of the cylinder. This phenomenon is known as vortex shedding and is a key feature of the flow past a circular cylinder.

As the flow speed increases further, the vortex shedding becomes more complex, and multiple vortices are shed simultaneously. This leads to a series of bifurcations in the flow, where the flow undergoes sudden changes in behavior as the speed of the flow increases. These bifurcations can be seen as a series of transitions from laminar flow to periodic vortex shedding, and then to chaotic flow.

5.1 Primary instability

At low Reynolds numbers, the flow is two-dimensional, steady and symmetric with respect to the cross-stream direction. The most well-known bifurcation in the flow past a circular cylinder is the Hopf bifurcation, which occurs as the first bifurcation in the flow and is connected to the loss of temporal symmetry as the periodic vortex shedding becomes unstable initiating the process of transition to turbulence.

The flow becomes unstable at a critical Reynolds number, $Re_{c,1} = UD/\nu \approx 46.6$. This bifurcation is marked by the emergence of periodic fluctuations in the cylinder's wake, which can be represented by a single frequency in Fourier space, characterized by a Strouhal number $St_c \simeq 0.116$. This primary instability serves as a classic example of a supercritical Hopf-type bifurcation. Numerous wake flows display instabilities leading to the onset of periodic vortex shedding, which results in their classification as *flow oscillators*. In figure 1 and figure 2 one can observe the base flow and the evolution of the growth rate and frequency as a function of the Reynolds number.

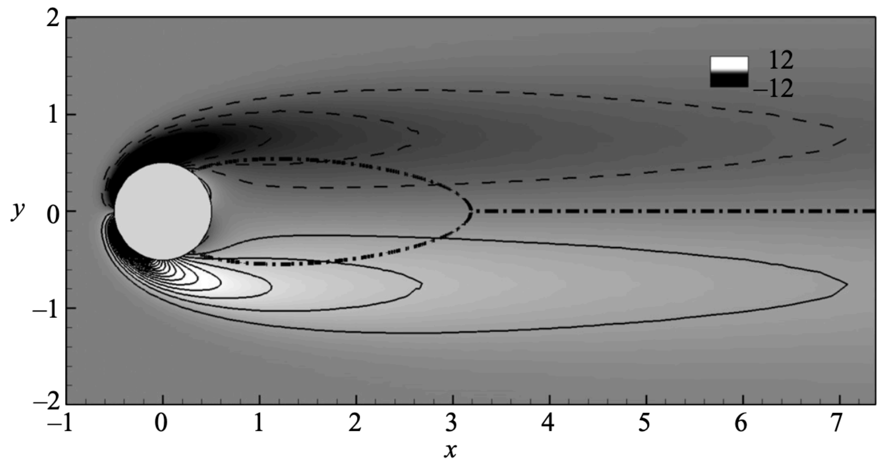

Figure 1: Cylinder base flow: spatial distribution of the vorticity for the Reynolds number $Re = 46.8$. The dash-dotted line is the dividing streamline delimiting the recirculation flow. Only a small portion of the computational domain is shown. Reproduced from Marquet et al. [2008]

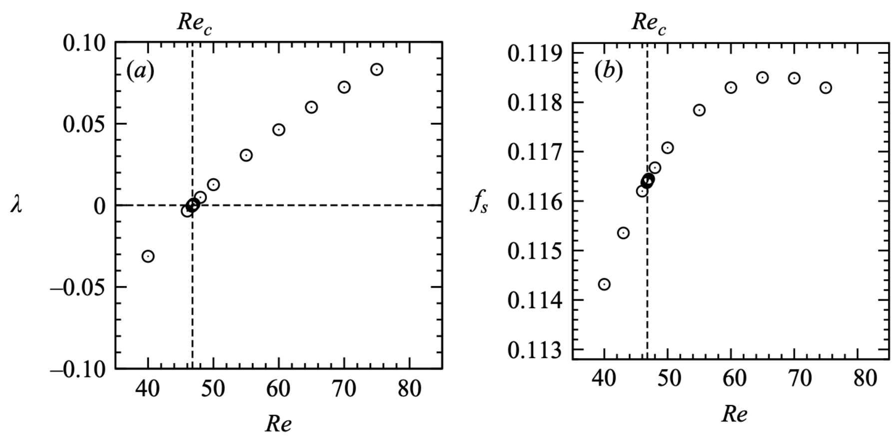

Figure 2: Cylinder flow. (a) Growth rate λ and (b) frequency $f_s = \omega/2\pi$ of the leading global mode as a function of the Reynolds number Re. The critical Reynolds number Re_c is shown by the vertical dashed line. Reproduced from Marquet et al. [2008]

6 Governing equations and numerical setup

The dynamics of the flow are governed by the incompressible Navier-Stokes equations (NSE), in the form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot (\mathbf{U} \otimes \mathbf{U}) = -\nabla P + \frac{1}{Re}\nabla^2 \mathbf{U}, \tag{1}$$

$$\nabla \cdot \mathbf{U} = 0. \tag{2}$$

where $\mathbf{U} = (U, V)^T$ is the velocity vector (with steamwise and cross-section velocity components) and P is the pressure field. The Reynolds number is defined as

$$Re = \frac{U_{in} D}{\nu}, \tag{3}$$

where U_{in} is the (constant) velocity at the inflow, D the cylinder diameter, and ν the (constant) kinematic viscosity coefficient of the fluid. The NSE are completed with the following boundary conditions: $\mathbf{U} = (1, 0)$ at the inlet, an outflow boundary condition at the outlet, (Neumann) symmetric boundary conditions on the lateral boundaries and (Dirichlet) no-slip conditions $\mathbf{U} = 0$ on the solid wall around the cylinder.

The NSE are solved using the incompressible flow solver Nek5000 which is based on the spectral element method (SEM). A non-staggered formulation has been used: the velocity field is discretized using Nth degree Lagrange interpolants, defined on the Gauss-Lobatto-Legendre quadrature points, as basis and trial functions, while the pressure field is discretized using Lagrange interpolants of degree $N - 2$ defined on the Gauss-Legendre quadrature points. Finally, time integration is performed using the BDF3/EXT3 scheme: integration of the viscous terms relies on backward differentiation (BDF3), while the convective terms are integrated explicitly using a third order accurate extrapolation (EXT3).

The domain is discretized by a two-dimensional mesh composed of 1996 spectral elements (66 in the flow direction and 32 in the cross-section). The cylinder has a diameter of $D = 1$ and the inflow velocity is set to $U_{in} = 1$, in a way that the Reynolds number reduces to the inverse of the kinematic viscosity coefficient. Note that here that the nondimensional frequency expressed by the Strouhal number $St = fD/U_{in}$ reduces to $St = f$. To limit the computational cost, we consider Lagrange interpolants of order $N = 5$, which shows good convergence compared to $N = 7$ which leads to a total of 71 856 grid points. We consider a third-order accurate temporal scheme and a time-step that satisfies Co ≤ 0.5 for all simulations. In the code, the user does not need to specify a time step as it will be computed on the fly to match the target Co specified by the user.

6.1 Linear Stability Analysis

Examining the global stability of a system is crucial for detecting bifurcations and identifying stable or unstable eigenpairs (combinations of eigenvalues and eigenvectors) or modes, which are associated with the transition to turbulence. We will analyze small disturbances, denoted by $\mathbf{u}$, that occur in the vicinity of a solution (specifically, a steady solution) of the Navier-Stokes equations (NSE), represented by the vector $\mathbf{U}_b$. This solution is calculated using a Newton method, which will be discussed in the following section. The linearized Navier-Stokes equations (LNSE) describe the behavior of these disturbances in relation to the base flow:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\, \mathbf{U}_b + (\mathbf{U}_b \cdot \nabla)\, \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}, \tag{4}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{5}$$

where the base flow vector $\mathbf{U}_b = (U_b, V_b)^T$ and the fluctuations vector are given by $\mathbf{u} = (u, v)^T$. By projecting the LNSE onto a divergence-free vector space, we obtain a more concise form:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{A}\mathbf{u}.$$

In this equation, $\mathbf{A}$ represents the projection of the LNSE operator onto the divergence-free vector space. The long-term behavior of a small disturbance $\mathbf{u}$ is determined by the eigenspectrum of $\mathbf{A}$.

However, computing the leading eigenvalues of $\mathbf{A}$ directly is challenging due to its large dimensions after discretization of the problem (even using SEM). In practice, we do not even attempt to construct $\mathbf{A}$. Instead, we use a time-stepper method that calculates approximations for the leading eigenvalue pairs of the exponential propagator:

$$\mathbf{M}(\Delta t) = e^{\mathbf{A}\Delta t}.$$

in doing so we only need the action of $\mathbf{A}$ on a vector (matrix-vector product) which is equivalent to time-stepping the LNSE. The action of this propagator on an initial vector (that can be composed by random noise) is computed by marching the LNSE from $t = 0$ to $t = \Delta t$. Then, iterative eigenvalue solvers project the propagator $\mathbf{M}$ onto an orthonormal set of vectors that span a Krylov subspace of size k, resulting in a low-dimensional $k \times k$ Hessenberg matrix $\mathbf{H}$.

Due to the smaller size of $\mathbf{H}$, its eigenvalues μ can be easily computed and are related to the eigenvalues of the LNSE operator $\mathbf{M}$ via the transformation

$$\lambda_k = \frac{\log(\mu_k)}{\tau}$$

where τ is the integration period (*endTime*). The complex eigenvalues of $\mathbf{M}$ (approximated by the eigenvalues of $\mathbf{H}$) are $\lambda_k = \sigma_k + i\omega_k$, where the real part corresponds to the growth rate and the imaginary part to the angular frequency $\omega = 2\pi f$.

We obtain $\mathbf{H}$ with the Arnoldi iteration, which generates an orthonormal basis for the Krylov subspace and forms an upper Hessenberg matrix whose eigenvalues approximate those of the original matrix.

7 Computing the Base Flow (BF)

In this example we are going to compute the base flow (a vector field that is a solution to the NSE) at the Reynolds number $Re = 100$. We will use a Newton-Krylov method. Details on the method are given in Frantz et al. [2023]. To speed up computations, the base flow at $Re = 40$ is provided in the folder and it should be used as an initial condition.

Start by editing `1cyl.par`, and setting an initial condition (*startFrom*), integration time (*endTime*) and *userParam* to Newton-Krylov mode:

```
startFrom = BFRe40_1cyl0.f00001
endTime = 1.0
userParam01 = 2
```

You can change (absolute) solver tolerances, to obtain more precise solutions:

```
[PRESSURE]
residualTol = 1.0E-06
[VELOCITY]
residualTol = 1.0E-06
```

Here we consider the minimum value to reduce the computational cost, but you can choose between 10^{-6} and 10^{-11}. nekStab will automatically set as target for the base flow computation the larger value between the two solvers.

The Reynolds number is specified in the end of `1cyl.par`, note that it should be entered as a **negative value** for Nek5000 to consider the problem as non-dimensional:

```
[VELOCITY]
viscosity = -100.0 # Reynolds number (given as negative value)
```

You can now execute the code with:

```
nekbmpi 1cyl 4
```

and follow the evolution of the output with:

```
tail -f logfile
```

Use “Ctrl+c” to exit.

To discover how many core your computer have just use:

```
htop
```

If you need to stop the code:

```
killall nek5000
```

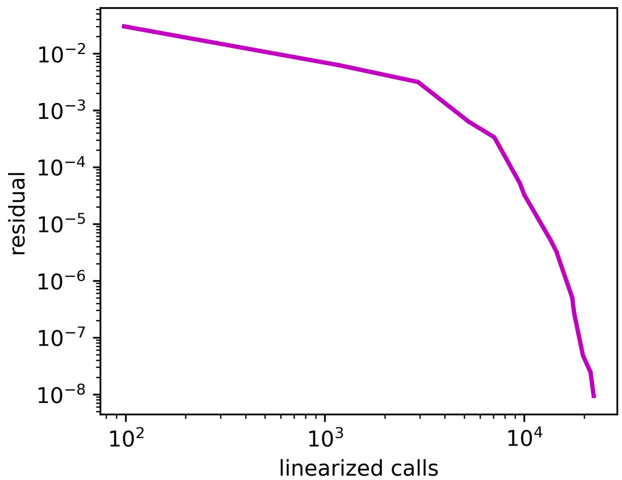

Figure 3: Residual evolution for the computation of the base flow at $Re = 100$ from the initial condition at $Re = 40$ using the Newton-Krylov method.

Otherwise, as the code completes you can plot the residual evolution that is written to *residu.dat*:

```
python3 plot_newton.py
```

You will see the residual evolution (decay) as a function of the number of the linearized calls (see figure 3). Note that we can observe the quadratic decay characteristic of the Newton method.

8 Computing the (direct) eigenvalue problem

After calculating the base flow, which is automatically stored in the file `BF_1cyl0.f00001`, one can proceed to compute the linear stability analysis to characterize the stability properties of the same. To achieve this, we need to modify `1cyl.par` once more, updating the initial condition (*startFrom*) to the computed base flow, setting the integration time (*endTime*), and configuring *userParam* for the direct eigenproblem mode:

```
startFrom = BF_1cyl0.f00001
endTime = 1.
userParam01 = 3.1
```

As in the previous section, you can execute the code using the following command:

```
nekbmpi 1cyl 4
```

To monitor the progress of the output, use:

```
tail -f logfile
```

To exit, press “Ctrl+c”.

Upon completion, you can plot both spectrum (shown in figure 4):

```
python3 plot_spectra.py
...
Reading Spectre_NSd.dat
Mode:  1
 sigma= 0.1275481
 omega= 0.7447225
     f= 0.1185263
```

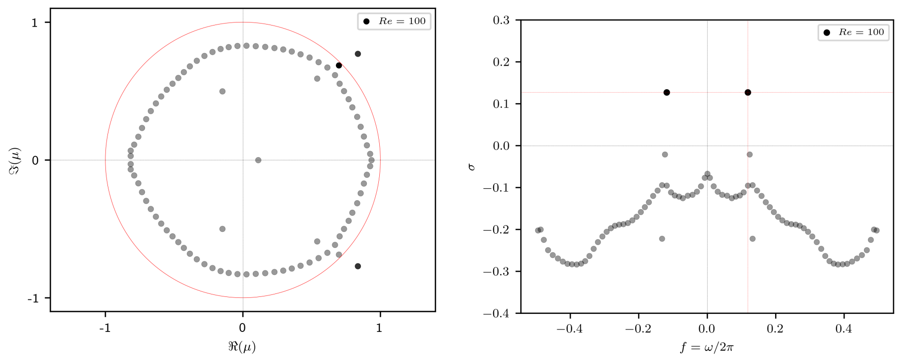

Figure 4: Spectral analysis of the LNSE: (left) Eigenspectrum of the Hessenberg matrix and (right) Eigenspectrum of the LNSE or analogous to the exponential propagator $\mathbf{M}$.

9 Run a DNS

Now that you have computed a base flow and its eigenvalues, we can run a DNS from the base flow to verify if the same will be stable or unstable and check the frequency of the vortex shedding.

To achieve this, we need to modify `1cyl.par` once more, updating the initial condition (*startFrom*) to the computed base flow, setting the integration time (*endTime*), and configuring *userParam* for the direct eigenproblem mode:

```
startFrom = BF_1cyl0.f00001
endTime = 500.
userParam01 = 0
```

You can execute the code using the following command:

```
nekbmpi 1cyl 4
```

To monitor the progress of the output, use:

```
tail -f logfile
```

To exit, press “Ctrl+c”.

Upon completion, you can plot the spectrum:

```
python3 plot_fft.py
```

To have a clear peak in the spectrum one can increase the skip time `tskps`.

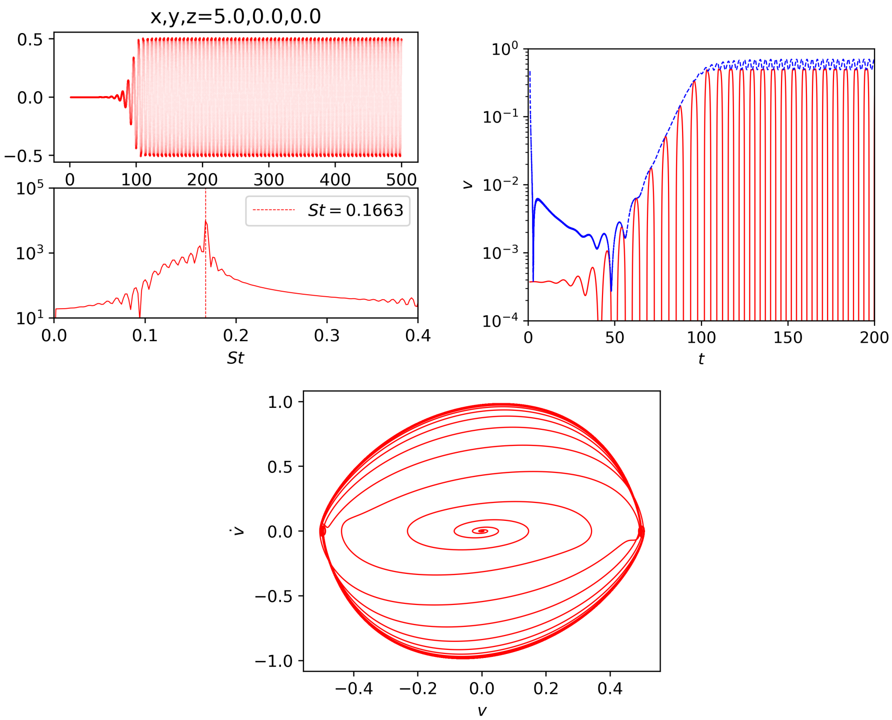

Figure 5: (upper, left) Vertical velocity and FFT spectrum; (upper, right) Log plot and Hilbert transform of the signal (in blue); (bottom) Phase space plot of the velocity signal.

In the following figure 5, one can observe the vertical velocity signal from a probe located in the center axis at $(x, y) = (5, 0)$ (note that at this position the mean of the vertical velocity is zero). We compute the Hilbert transform and extract the envelope of the signal, in blue, and observe the linear amplification region and the arrival of nonlinear saturation. Also, one can plot the phase space of the signal and observe that the dynamics settle in a periodic orbit.

Note that the linear stability predicted a $f = 0.118$ while the frequency found in the FFT spectrum peak is of $St = 0.166$. This discrepancy is explained by the nonlinear distortion of the baseflow that increases as we depart from the bifurcation point. This means that the two values should be very close just after bifurcation and the discrepancy should increase as we are far away from the bifurcation point.

At $Re = 100$ one can observe a discrepancy between the linear stability frequency ($f = 0.118$) and the frequency found in the FFT (Fast Fourier Transform) spectrum peak ($St = 0.166$). This discrepancy can be attributed to the increase in the nonlinear distortion which can be seen as the amplitude of the difference between the instantaneous flow and the base flow. This becomes more

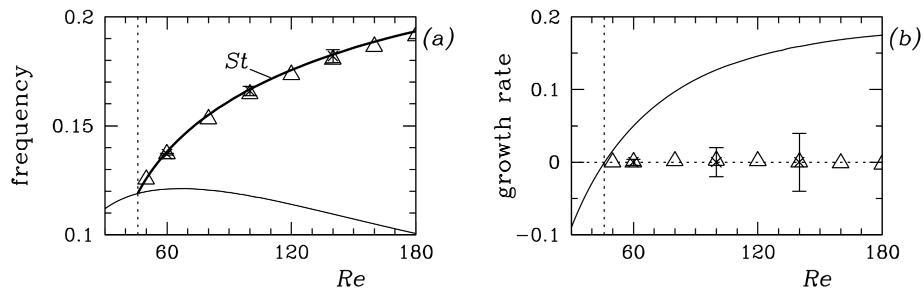

Figure 6: (a) Frequencies and (b) growth rates as a function of Reynolds number. In (a) the bold curve labelled St is frequency from the vortex shedding obtained by DNS. All other results in (a) and (b) are from linear stability computation. Thin curves are for the base flow. Triangles are for the mean flow. Note that the discrepancy between the frequency predicted by linear stability analysis and the frequency from the DNS is more pronounced as the Reynolds number increase. Note also that the linear stability analysis conducted on the mean flow (instead of the base flow) yields eigenvalues with zero growth rate, which is expected as the flow is saturated and nothing is growing or decaying. Vertical dashed lines indicate Re_c. Reproduced from Barkley [2006].

pronounced as we move further from the bifurcation point.

In a fluid dynamics context, the bifurcation point is the point where the flow transitions from one state to another, often due to a change in a control parameter such as the Reynolds number or flow speed. As we move away from this point, the flow becomes increasingly nonlinear, which can result in differences between the predictions made by linear stability analysis and the actual behavior observed in experiments or numerical simulations. Just after the bifurcation point, the two values should be very close, indicating that the flow is close to the linear regime. However, as we move further from the bifurcation point, the discrepancy between the two values should increase, highlighting the increasing influence of nonlinear effects on the flow. This is explored in figure 6.

References

Dwight Barkley. Linear analysis of the cylinder wake mean flow. *Europhysics Letters*, 75(5):750, 2006.

R. A. S. Frantz, J.-Ch. Loiseau, and J.-Ch. Robinet. Krylov Methods for Large-Scale Dynamical Systems: Application in Fluid Dynamics. *Applied Mechanics Reviews*, 75(3), 03 2023.

Olivier Marquet, Denis Sipp, and Laurent Jacquin. Sensitivity analysis and passive control of cylinder flow. *Journal of Fluid Mechanics*, 615:221–252, 2008.

Final Project

- Form a group and choose a unique Reynolds number value for analysis. Talk to other groups to ensure that each group investigates a distinct value of the Reynolds number.
- Start by computing the base flow and then move to assessing its stability at the selected Reynolds number. Create plots depicting the residual evolution and the eigenspectrum.
- Conduct a Direct Numerical Simulation (DNS) and compare the frequency found from the Fast Fourier Transform (FFT) with the linear stability prediction (eigenvalue frequency).
- Prepare a concise report explaining the obtained plots and results. Discuss the discrepancies between the DNS and linear stability predictions. Optionally, extend the analysis to other Reynolds number values.
- Use ParaView to open the .nek5000 files and add screenshots of the base flow and eigenmodes. (do not install ParaView with *apt* or *apt-get* as such version does not support Nek5000 files. You need to download the latest version from `https://www.paraview.org/download`)
- Explain what is happening on the flow. Is the base flow stable or unstable? Why? What type of bifurcation the flow experienced? What it means to the dynamics?
- Examine the role of non-linear effects on flow dynamics and stability. Describe how these effects become more significant as the Reynolds number increases.
- Try to compare your findings with experimental or other numerical studies in the literature. Assess the agreement between your predictions and these studies, identifying potential factors contributing to any discrepancies.
- Discuss the practical implications of your results and the significance of linear stability analysis for real-world applications, such as drag reduction, flow control, or energy harvesting. Explain the importance of understanding the mechanisms underlying the transition to turbulence and why engineers care about it.
- Deadline: Please email your group report in PDF format to `ricardo.schuh_frantz@sorbonne-universite.fr` by Sunday, April 9th at midnight.