Worksheet n°3: Multiple time-scale analysis and amplitude equations

1/ Direct numerical simulation of cylinder flow at Re=100

We solve the unsteady Navier-Stokes equations in perturbative form ($w \coloneqq w_0 + w$) around a cylinder flow at $Re = \nu^{-1} = 100$. The initial condition is the real part of a small amplitude unstable global mode.

$$\mathcal{B}\partial_t w + \mathcal{N}_{w_0} w + \mathcal{L}w = -\frac{1}{2}\mathcal{N}(w,w)$$

$$w(0) = \alpha \mathrm{Re}(\hat{w})$$

with :

$$w = \begin{pmatrix} u \\ v \\ p \end{pmatrix},\ \mathcal{B} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ \mathcal{N}(w_1,w_2) = \begin{pmatrix} u_1 \cdot \nabla u_2 + u_2 \cdot \nabla u_1 \\ 0 \end{pmatrix},\ \mathcal{N}_{w_0} w = \mathcal{N}(w_0, w),$$

$$\mathcal{L} = \begin{pmatrix} -\nu\Delta() & \nabla() \\ -\nabla \cdot () & 0 \end{pmatrix}$$

The base-flow and the global mode are defined by:

$$\frac{1}{2}\mathcal{N}(w_0,w_0) + \mathcal{L}w_0 = 0$$

$$\lambda\mathcal{B}\hat{w} + \left(\mathcal{N}_{w_0} + \mathcal{L}\right)\hat{w} = 0$$

In DNS/Mesh:

FreeFem++ mesh.edp

In DNS/BF:

FreeFem++ init.edp

FreeFem++ newton.edp

In DNS/Eigs:

FreeFem++ eigen.edp

In DNS/DNS:

FreeFem++ init.edp // generate initial condition from small amplitude global mode

FreeFem++ dns.edp // launch DNS simulation

Octave plotlinlog('out_0.txt',1,2,1) // represent energy as a function of time in fig 1

Octave plotlinlin('out_0.txt',1,4,2) // represent v velocity as a function of time

2/ Van der Pol Oscillator: multiple time-scale analysis

The Van der Pol Oscillator corresponds to the following governing equations:

$$w'' + \omega_0^2 w = 2\tilde{\delta} w' - w^2 w'$$

$$w(0) = w_I, w'(0) = 0$$

where the $(\cdot)'$ is the time-derivative, w_I is the initial condition, ω_0 the frequency and $\tilde{\delta}$ the instability strength. Here, we choose: $\omega_0 = 10$, $\tilde{\delta} = 0.3$ and $w_I = 0.01$.

2a/ Numerical time-integration

We integrate in time the above equations. For this,

In VanDerPol:

Octave pkg load all // load external packages for time integration, Fourier analysis, etc.

Octave vdp // integrate in time unforced Van der Pol equations

2b/ One time-scale approach

We try to approximate the solution by considering a small instability strength: $\tilde{\delta} = \delta\epsilon$, with $\epsilon \ll 1$ and $\delta = O(1)$. We look for an approximation of the solution with an expansion of the form:

$$w = \epsilon^{\frac{1}{2}} y \text{ and } y = y_0 + \epsilon y_1 + \cdots.$$

We first try with only one time-scale: $y(t) = y_0(t) + \epsilon y_1(t) + \cdots$

The second-order solution is given by:

$$w = \left(\tilde{A} e^{i\omega_0 t} + \text{c.c.}\right) + \left(\frac{-3\tilde{A}^3 + 12\tilde{\delta}\tilde{A}}{8\omega_0} i e^{i\omega_0 t} + \frac{\mathrm{i}\tilde{A}^3}{8\omega_0} e^{3i\omega_0 t} - \left(2\tilde{\delta}\tilde{A} - \tilde{A}^3\right)\left(\frac{1 + 2i\omega_0 t}{4\omega_0}\right) i e^{i\omega_0 t} + \text{c.c.}\right)$$

$$\tilde{A} = \frac{w_I}{2}$$

To represent this solution, in VanDerPol:

Octave clf // clear all figures

Octave vdp // integrate in time unforced Van der Pol equations

Octave vdp_tlr // show first and second order approximations with one time-scale

2c/ Two time-scales approach

The two time-scale first-order solution is given by:

$$w(t) = \left(\tilde{A} e^{i\omega_0 t} + \text{c.c.}\right)$$

with:

$$\frac{d\tilde{A}}{dt} = \tilde{\delta}\tilde{A} - \frac{1}{2}\tilde{A}^3$$

$$\tilde{A}(0) = \frac{w_I}{2}$$

To represent this solution, in VanDerPol:

Octave clf // clear all figures

Octave vdp // integrate in time unforced Van Der Pol equations

Octave vdp_tlr // show first and second order approximations with one time-scale

Octave vdp_mts // show first and second order approximations with two time-scales

3/ Van der Pol Oscillator with harmonic forcing

We consider the forced Van der Pol oscillator:

$$w'' + \omega_0^2 w = 2\tilde{\delta}w' - w^2 w' + \tilde{E}\cos\omega_f t,$$

where ω_f and $\tilde{E}$ are respectively the forcing frequency and the forcing amplitude. Here, we choose: $\omega_f = 25$ and $\tilde{E} = 600$. The first-order two time-scale solution is given by:

$$w(t) = 2\tilde{A}\cos(\omega_0 t + \phi) + \frac{\tilde{E}}{\omega_0^2 - \omega_f^2}\cos\omega_f t$$

with:

$$\frac{d\tilde{A}}{dt} = \left[\tilde{\delta} - \frac{1}{4}\left(\frac{\tilde{E}}{\omega_0^2 - \omega_f^2}\right)^2\right]\tilde{A} - \frac{1}{2}\tilde{A}^3$$

To represent this solution, in VanDerPol:

Octave clf // clear all figures

Octave vdpf // integrate in time unforced Van Der Pol equations

Vary the forcing amplitude $\tilde{E}$ from 0 to 600 and observe in each case the resulting frequency spectrum.

4/ Forced Navier-Stokes equations

We consider the Navier-Stokes equation in perturbative form ($w := w_0 + w$) with a forcing term acting on the momentum equations:

$$\mathcal{B}\partial_t w + \mathcal{N}_{w_0} w + \mathcal{L}w = \tilde{\delta}\mathcal{M}(w_0 + w) - \frac{1}{2}\mathcal{N}(w, w) + \left(\tilde{E}e^{i\omega_f t} f + \mathrm{c.c}\right).$$

Here:

$$w = \begin{pmatrix} u \\ v \\ p \end{pmatrix},\ \mathcal{L} = \begin{pmatrix} -\nu_c \Delta() & \nabla() \\ -\nabla \cdot () & 0 \end{pmatrix},\ \mathcal{M} = \begin{pmatrix} -\Delta & 0 \\ 0 & 0 \end{pmatrix}.$$

The viscosity ν has been replaced by $\nu = \nu_c - \tilde{\delta}$, where ν_c is the critical viscosity which achieves marginal stability of the linear dynamics $Re_c = \nu_c^{-1} = 46.6$.

The base-flow is given by:

$$\frac{1}{2}\mathcal{N}(w_0, w_0) + \mathcal{L}w_0 = 0,$$

while $\tilde{E}$ and ω_f correspond respectively to the forcing amplitude and forcing frequency. The forcing structure f (acting solely on the momentum equations, so that $\mathcal{B}f = f$) is also given.

In the following, we consider a slightly supercritical regime (the Reynolds number is slightly above the critical Reynolds number):

$$\tilde{\delta} = \epsilon\delta,\ \epsilon \ll 1, \delta = O(1),$$

and a small- amplitude forcing, which scales as:

$$\tilde{E} = \epsilon^{\frac{1}{2}} E, E = O(1).$$

We look for an approximation of the solution under the form:

$$w = \epsilon^{\frac{1}{2}}\left(y_0(t, \tau = \epsilon t) + \epsilon^{\frac{1}{2}} y_{\frac{1}{2}}(t, \tau = \epsilon t) + \epsilon^1 y_1(t, \tau = \epsilon t) + \cdots\right)$$

The second-order solution is given by:

$$w = \left(\tilde{A}e^{i\omega_c t} y_A + \mathrm{c.c}\right) + \left(\tilde{E}e^{i\omega_f t} y_E + \mathrm{c.c}\right) + \tilde{\delta} w_\delta + \left(\tilde{A}^2 e^{2i\omega_c t} y_{AA} + \mathrm{c.c.}\right) + \left|\tilde{A}\right|^2 y_{A\bar{A}} + \left|\tilde{E}\right|^2 y_{E\bar{E}} + \left(\tilde{A}\tilde{E}e^{i(\omega_c+\omega_f)t} y_{AE} + \mathrm{c.c.}\right) + \left(\tilde{A}\bar{\tilde{E}}e^{i(\omega_c-\omega_f)t} y_{A\bar{E}} + \mathrm{c.c.}\right) + \cdots$$

With :

$$i\omega_c \mathcal{B} y_A + \mathcal{N}_{w_0} y_A + \mathcal{L} y_A = 0$$

$$i\omega_f \mathcal{B} y_E + \mathcal{N}_{w_0} y_E + \mathcal{L} y_E = f$$

$$\mathcal{N}_{w_0} y_\delta + \mathcal{L} y_\delta = \mathcal{M} y_0$$

$$2i\omega_c \mathcal{B} y_{AA} + \mathcal{N}_{w_0} y_{AA} + \mathcal{L} y_{AA} = -\frac{1}{2}\mathcal{N}(y_A, y_A)$$

$$\mathcal{N}_{w_0} y_{A\bar{A}} + \mathcal{L} y_{A\bar{A}} = -\mathcal{N}(y_A, \bar{y}_A)$$

$$\mathcal{N}_{w_0} y_{E\bar{E}} + \mathcal{L} y_{E\bar{E}} = -\mathcal{N}(y_E, \bar{y}_E)$$

$$2i(\omega_c + \omega_f)\mathcal{B} y_{AE} + \mathcal{N}_{w_0} y_{AE} + \mathcal{L} y_{AE} = -\mathcal{N}(y_A, y_E)$$

$$2i(\omega_c - \omega_f)\mathcal{B} y_{A\bar{E}} + \mathcal{N}_{w_0} y_{A\bar{E}} + \mathcal{L} y_{A\bar{E}} = -\mathcal{N}(y_A, \bar{y}_E)$$

And:

$$\frac{d\tilde{A}}{dt} = \lambda\delta\tilde{A} - \mu\tilde{A}|\tilde{A}|^2 - \pi\tilde{A}|\tilde{E}|^2$$

where:

$$\lambda =< \tilde{y}_A,\ \mathcal{M}y_A > -< \tilde{y}_A, \mathcal{N}(y_A, y_\delta)$$

$$\mu =< \tilde{y}_A, \mathcal{N}(y_A, y_{A\bar{A}}) + \mathcal{N}(\bar{y}_A, y_{AA}) >$$

$$\pi =< \tilde{y}_A, \mathcal{N}(y_A, y_{E\bar{E}}) + \mathcal{N}(y_{\bar{E}}, y_{AE}\ ) + \mathcal{N}(y_E,\ y_{A\bar{E}}) >$$

$$-i\omega_c\mathcal{B}\tilde{y}_A + \widetilde{\mathcal{N}}_{w_0}\tilde{y}_A + \tilde{\mathcal{L}}\tilde{y}_A = 0$$

$$< \tilde{y}_A, \mathcal{B}y_A > \ = 1$$

4a/ In AmplEq/Mesh:

FreeFem++ mesh.edp // generate mesh

In AmplEq/BF:

FreeFem++ init.edp // generate initial guess for Newton iterations

FreeFem++ newton.edp // Newton iteration

In AmplEq/Eigs:

FreeFem++ eigen.edp // compute global mode

FreeFem++ eigenadj.edp // compute adjoint global mode

FreeFem++ norm.edp // generate scaled adjoint global mode

In AmplEq/WNL:

FreeFem++ udelta.edp // generate modification of base-flow due to increase in Reynolds number

FreeFem++ uAA.edp // generate second harmonic due to interaction of global mode with himself

FreeFem++ uAAb.edp // generate zero-harmonic due to interaction of global mode with adjoint of himself

FreeFem++ lambda.edp // compute λ coefficient of Stuart-Landau equation

FreeFem++ mu.edp // compute μ coefficient of Suart-Landau equation

FreeFem++ forcing.edp // define external forcing (spatial structure anf frequency)

FreeFem++ uE.edp // coumpute response due to external forcing

FreeFem++ uAE.edp // compute AE-harmonic due to interaction of response to external forcing with global mode

4b/ Complete program uAEb.edp to compute the $A\bar{E}$ harmonic due to the interaction of the global mode with the adjoint of the response due to external forcing.

4c/ Complete program uEEb.edp to compute the zero-harmonic due to the interaction of the external forcing response with the conjugate of himself.

4d/ Complete program pi.edp to compute the π coefficient.

5/ Forced Direct numerical simulation

We integrate in time the forced Navier-Stokes equations at $Re = \nu^{-1} = 100$:

$$\mathcal{B}\partial_t w + \mathcal{N}_{w_0} w + \mathcal{L} w = -\frac{1}{2}\mathcal{N}(w,w) + \left(\tilde{E}e^{i\omega_f t} f + \mathrm{c.c}\right)$$

where:

$$w = \begin{pmatrix} u \\ v \\ p \end{pmatrix},\ \mathcal{L} = \begin{pmatrix} -\nu\,\Delta() & \nabla() \\ -\nabla\cdot() & 0 \end{pmatrix}$$

In DNS/DNS:

FreeFem++ dnsf.edp // launch forced DNS simulation

Octave plotlinlog('out_4000.txt',1,2,1) // represent energy as a function of time in fig 1

Octave plotlinlin('out_4000.txt',1,4,2) // represent v velocity as a function of time in fig 2

Octave spectrum // compare spectrum with and without control