Weakly nonlinear evolution of stochastically driven nonnormal systems : supplementary materials

Yves-Marie Ducimetière[1] †, Edouard Boujo[1] and François Gallaire[1]

[1]Laboratory of Fluid Mechanics and Instabilities, EPFL, CH1015 Lausanne, Switzerland

This document includes several details of calculations and methods used in the main text.

1. Discretisation of the amplitude equation in the frequency domain

We recall from the main text (Eq.(3.5) for $\boldsymbol{f}_h = \boldsymbol{f}_o$) that we seek the equilibrium solution of the amplitude equation

$$\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{q}}\}\frac{\mathrm{d}\hat{B}}{\mathrm{d}\tau_1} = \hat{g}_2[\hat{B}] + \{\hat{\boldsymbol{a}}, \boldsymbol{f}_o\}(\phi\hat{\xi} - \hat{B}) = 0. \tag{1.1}$$

The first step is to discretize it, in the frequency domain. For this purpose, let ω_s designates the sampling frequency of the corresponding temporal signals, that are discretized using $2(N-1)$ (a power of two) uniformly distributed points between $t = 0$ and $t = T$. In this manner, we have in practice $-\omega_c \leqslant \omega \leqslant \omega_c$ where $\omega_c = \omega_s/2$, and the positive part of this frequency interval is discretized with N uniformly distributed points between $\omega = 0$ and $\omega = \omega_c$. Namely, the discrete set of positive frequencies writes $\omega_n = (n-1)\Delta\omega$ for $n = 1, 2, ...N$ and $\Delta\omega = \omega_c/(N-1)$. The discrete Fourier transforms varying over this interval are real-valued in $\omega_1 = 0$ and $\omega_N = \omega_c$, but generally complex everywhere else; for instance, $\hat{B}(\omega)$ is discretized as $[B_{r,1}, B_{r,2} + iB_{i,2}, ..., B_{r,N-1} + iB_{i,N-1}, B_{r,N}]$ which amounts to $2(N-2) + 2 = 2(N-1)$ independent components in total. Only the variation over the set of positive frequencies is needed, as the Fourier component of a real-valued signal at a negative frequency is the complex conjugate of the one at the opposite frequency : $\hat{B}(-\omega) = \hat{B}(\omega)^*$. With (1.1) discretized, the following simple procedure is implemented in `MATLAB`:

Algorithm :

(i) Choose the values for N and ω_c, which sets the discretisation of the frequencies.

(ii) Over the discrete set of frequencies, pre-compute once for all the *determinitic* fields $\boldsymbol{f}_o$, $\hat{\boldsymbol{q}}$, $\hat{\boldsymbol{a}}$.

(iii) Choose a value for ϕ (which sets the forcing amplitude).

(iv) Draw a white noise ($|\hat{\xi}(\omega)| = 1\ \forall\omega$, but random phases uniformly distributed between 0 and 2π), for instance with the commands `xi = exp(1i*2*pi*rand(1,N))`, then `xi(1)=real(xi(1))/abs(real(xi(1)))` and `xi(N)=real(xi(N))/abs(real(xi(N)))`.

(v) Find the $\hat{B}$ that solves $r(\hat{B}; \phi, \hat{\xi}) = 0$ where $r(\hat{B}; \phi, \hat{\xi}) \doteq \hat{g}_2[\hat{B}] + \{\hat{\boldsymbol{a}}, \boldsymbol{f}_o\}(\phi\hat{\xi} - \hat{B})$, using the nonlinear solver "`fsolve`"; the functional $\hat{g}_2[\hat{B}]$ is evaluated using the commands "`ifft`"

† Email address for correspondence: yves-marie.ducimetiere@epfl.ch

and "fft".

(vi) Update the statistics on $\hat{B}$, for instance its ensemble average, and, if not converged go back to (iv).

Of course the convergence in terms of N and ω_c must be ensured.

2. The particular case of the NSE: discretisation of the amplitude equation in the frequency domain

We recall from the main text that

$$\begin{aligned}\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{q}}\}\frac{\partial \hat{B}}{\partial \tau_1} &= \hat{g}_2[\hat{B}], \quad \text{and} \\ \{\hat{\boldsymbol{a}}, \hat{\boldsymbol{q}}\}\frac{\partial \hat{B}}{\partial \tau_2} &= \{\hat{\boldsymbol{a}}, \boldsymbol{f}_o\}(\phi\hat{\xi} - \hat{B}) + \hat{g}_3[\hat{B}] - \frac{\partial \{\hat{\boldsymbol{a}}, \hat{\boldsymbol{u}}_2[\hat{B}]\}}{\partial \tau_1}.\end{aligned} \tag{2.1}$$

The equilibrium solution(s) of the assembled amplitude equation solves :

$$\begin{aligned}&\frac{\mathrm{d}\hat{B}}{\mathrm{d}\tau_1} = \frac{\partial \hat{B}}{\partial \tau_1} + \sqrt{\epsilon_o}\frac{\partial \hat{B}}{\partial \tau_1} = 0 \Leftrightarrow \\ &\hat{g}_2[\hat{B}] + \sqrt{\epsilon_o}\left[\{\hat{\boldsymbol{a}}, \boldsymbol{f}_o\}(\phi\hat{\xi} - \hat{B}) + \hat{g}_3[\hat{B}] - \frac{\partial \{\hat{\boldsymbol{a}}, \hat{\boldsymbol{u}}_2[\hat{B}]\}}{\partial \tau_1}\right] = 0.\end{aligned} \tag{2.2}$$

In the following, after re-expressing the nonlinear terms $\hat{g}_2[\hat{B}]$, $\hat{g}_3[\hat{B}]$ and $\partial_{\tau_1}\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{u}}_2[\hat{B}]\}$ in (2.2) as convolution integrals, we discretize them in the frequency domain. In this manner, we make their dependency on the discrete set of $\hat{B}_i$ $(i = 1, 2, ..., N)$ as explicit as possible. All the other, linear, terms are simple to discretize.

2.1. *Derivation of the convolution integral*

We first develop:

$$\begin{aligned}&2C(\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_a], \mathcal{F}^{-1}[\hat{\boldsymbol{u}}_b]) = \nabla\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_a]\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_b] + \nabla\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_b]\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_a] \\ &= \frac{T}{4\pi^2}\int_{-\infty}^{\infty}\nabla\hat{\boldsymbol{u}}_a(p)e^{ipt}\mathrm{d}p\int_{-\infty}^{\infty}\hat{\boldsymbol{u}}_b(s)e^{ist}\mathrm{d}s + \frac{T}{4\pi^2}\int_{-\infty}^{\infty}\nabla\hat{\boldsymbol{u}}_b(s)e^{ist}\mathrm{d}s\int_{-\infty}^{\infty}\hat{\boldsymbol{u}}_a(p)e^{ipt}\mathrm{d}p \\ &= \frac{T}{4\pi^2}\iint_{-\infty}^{\infty}2C(\hat{\boldsymbol{u}}_a(p), \hat{\boldsymbol{u}}_b(s))e^{i(p+s)t}\mathrm{d}p\mathrm{d}s \\ &= \frac{T}{4\pi^2}\iint_{-\infty}^{\infty}2C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))e^{i\omega t}\mathrm{d}\omega\mathrm{d}s \\ &= \mathcal{F}^{-1}\left[\frac{\sqrt{T}}{2\pi}\int_{-\infty}^{\infty}2C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s\right],\end{aligned} \tag{2.3}$$

from which it comes immediately that :

$$\mathcal{F}\left[C(\mathcal{F}^{-1}[\hat{\boldsymbol{u}}_a], \mathcal{F}^{-1}[\hat{\boldsymbol{u}}_b])\right] = \frac{\sqrt{T}}{2\pi}\int_{-\infty}^{\infty}C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s. \tag{2.4}$$

2.2. *Discretisation of the convolution integral*

Let ω_s designates the sampling frequency of the temporal signals, such that we have in practice $-\omega_c \leqslant \omega \leqslant \omega_c$ where $\omega_c = \omega_s/2$. Over this set of frequencies, the integrand in (2.4) is defined if and only if we have both $-\omega_c \leqslant s \leqslant \omega_c$ and $\omega - \omega_c \leqslant s \leqslant \omega + \omega_c$. From now on considering only positive frequencies, i.e $0 \leqslant \omega \leqslant \omega_c$, the integrand is then defined if and only if

$$\omega - \omega_c \leqslant s \leqslant \omega_c. \tag{2.5}$$

Thereby

$$\begin{aligned}\int_{-\infty}^{\infty} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s &\approx \int_{\omega-\omega_c}^{\omega_c} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s \\ = \int_{\omega-\omega_c}^{0} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s &+ \int_{0}^{\omega_c} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s.\end{aligned} \tag{2.6}$$

The first of the two terms of the sum in (2.6) is transformed as

$$\begin{aligned}\int_{\omega-\omega_c}^{0} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s &= \int_{0}^{\omega_c-\omega} C(\hat{\boldsymbol{u}}_a(\omega + s), \hat{\boldsymbol{u}}_b(-s))\mathrm{d}s \\ &= \int_{0}^{\omega_c-\omega} C(\hat{\boldsymbol{u}}_a(\omega + s), \hat{\boldsymbol{u}}^*_b(s))\mathrm{d}s,\end{aligned} \tag{2.7}$$

where we used that $\hat{\boldsymbol{u}}_b(-s) = \hat{\boldsymbol{u}}^*_b(s)$ arising from the fact that all temporal signals are real-valued. The second of the two terms of the sum in (2.6) is transformed according to

$$\begin{aligned}&\int_{0}^{\omega_c} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s \\ &= \int_{0}^{\omega} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s + \int_{\omega}^{\omega_c} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s \\ &= \int_{0}^{\omega} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s + \int_{\omega}^{\omega_c} C(\hat{\boldsymbol{u}}^*_a(s - \omega), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s \\ &= \int_{0}^{\omega} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s + \int_{0}^{\omega_c-\omega} C(\hat{\boldsymbol{u}}^*_a(s), \hat{\boldsymbol{u}}_b(s + \omega))\mathrm{d}s.\end{aligned} \tag{2.8}$$

In this manner, only the knowledge of $\hat{\boldsymbol{u}}_a$ and $\hat{\boldsymbol{u}}_b$ over positives frequencies is required. Overall,

$$\begin{aligned}&\int_{\omega-\omega_c}^{\omega_c} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s = \\ &\int_{0}^{\omega} C(\hat{\boldsymbol{u}}_a(\omega - s), \hat{\boldsymbol{u}}_b(s))\mathrm{d}s + \int_{0}^{\omega_c-\omega} C(\hat{\boldsymbol{u}}_a(\omega + s), \hat{\boldsymbol{u}}^*_b(s)) + C(\hat{\boldsymbol{u}}^*_a(s), \hat{\boldsymbol{u}}_b(s + \omega))\mathrm{d}s.\end{aligned} \tag{2.9}$$

Let us now discretize (2.9). As said in the previous section, positive frequencies are discretized using N uniformly distributed points between $\omega = 0$ and $\omega = \omega_c$ (ω_c the cut-off frequency). Namely, the discrete set of positive frequencies writes $\omega_n = (n-1)\Delta\omega$ for $n = 1, 2, \ldots N$ and $\Delta\omega = \omega_c/(N-1)$. Eventually, the discrete version of expression (2.4) reads

$$\begin{aligned}&\mathcal{F}\left[C(\mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_a\right], \mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_b\right])\right] \approx \\ &\frac{\sqrt{T}}{2\pi}\left[\sum_{k=1}^{n} \delta^n_k C(\hat{\boldsymbol{u}}_{a,n-k+1}, \hat{\boldsymbol{u}}_{b,k}) + \sum_{k=1}^{N+1-n} \delta^{N+1-n}_k \left[C(\hat{\boldsymbol{u}}^*_{a,k}, \hat{\boldsymbol{u}}_{b,n+k-1}) + C(\hat{\boldsymbol{u}}_{a,n+k-1}, \hat{\boldsymbol{u}}^*_{b,k})\right]\right],\end{aligned} \tag{2.10}$$

where we used for instance $\hat{\boldsymbol{u}}_a(\omega_n - s_k) = \hat{\boldsymbol{u}}_a(\Delta\omega(n-1-k+1)) = \hat{\boldsymbol{u}}_{a,n-k+1}$. The scalar δ^j_i is a quadrature coefficient where $i \in [1, j]$ is a running index. In our computations, we used the trapezoidal method such that

$$\delta_1^1 = 0, \quad \text{and} \quad \delta_i^j = \begin{cases} \omega_c/(2(N-1)) & \text{if } i = 1 \\ \omega_c/(N-1) & \text{if } 1 < i < j \\ \omega_c/(2(N-1)) & \text{if } i = j \end{cases}, \quad \text{for} \quad j > 1. \tag{2.11}$$

2.3. *Discretisation of the amplitude equation*

Discretisation of $\hat{g}_2[\hat{B}]$:

Using (2.10), the functional $\hat{g}_2[\hat{B}] = -\{\hat{\boldsymbol{a}}, \mathcal{F}\left[C(\mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_1\right], \mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_1\right])\right]\}$ with $\hat{\boldsymbol{u}}_1 = \hat{B}\hat{\boldsymbol{q}}$ is discretized as :

$$\hat{g}_{2,n} = \sum_{k=1}^{n} \hat{B}_{n-k+1}\hat{B}_k\Theta_{nk} + \sum_{k=1}^{N+1-n} \hat{B}_{n+k-1}\hat{B}_k^*\Xi_{nk}, \tag{2.12}$$

with

$$\begin{aligned} \Theta_{nk} &= -\frac{\sqrt{T}}{2\pi}\delta_k^n\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n-k+1}, \hat{\boldsymbol{q}}_k)\}, \quad 1 \leqslant k \leqslant n, \\ \Xi_{nk} &= -\frac{\sqrt{T}}{\pi}\delta_k^{N+1-n}\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n+k-1}, \hat{\boldsymbol{q}}_k^*)\}, \quad 1 \leqslant k \leqslant N+1-n. \end{aligned} \tag{2.13}$$

The sums in (2.12) can also be written in matrix form :

$$\hat{g}_2[\hat{B}] = \begin{bmatrix} \Theta_{11}\hat{B}_1 & & \mathbf{0} \\ \vdots & \ddots & \\ \Theta_{1N}\hat{B}_N & \dots & \Theta_{NN}\hat{B}_1 \end{bmatrix} \begin{bmatrix} \hat{B}_1 \\ \vdots \\ \hat{B}_N \end{bmatrix} + \begin{bmatrix} \Xi_{11}\hat{B}_1 & \dots & \Xi_{1N}\hat{B}_N \\ \vdots & \iddots & \\ \Xi_{N1}\hat{B}_N & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{B}_1^* \\ \vdots \\ \hat{B}_N^* \end{bmatrix} \tag{2.14}$$

Discretisation of $\partial_{\tau_1}\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{u}}_2[\hat{B}]\} = \{\hat{\boldsymbol{a}}, \partial_{\tau_1}\hat{\boldsymbol{u}}_2[\hat{B}]\}$:

Since

$$\hat{\boldsymbol{u}}_2[\hat{B}] = R\left(\hat{\boldsymbol{g}}_2[\hat{B}] - \frac{\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{g}}_2[\hat{B}]\}}{\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{q}}\}}\hat{\boldsymbol{q}}\right), \tag{2.15}$$

where we recall that $\{\hat{\boldsymbol{a}}, \hat{\boldsymbol{g}}_2[\hat{B}]\} = \hat{g}_2[\hat{B}]$, expression (2.12) results in the following discretisation for the field $\hat{\boldsymbol{u}}_2[\hat{B}]$:

$$\hat{\boldsymbol{u}}_{2,m} = \sum_{j=1}^{m} \hat{B}_{m-j+1}\hat{B}_j\hat{\boldsymbol{d}}_{mj} + \sum_{j=1}^{N+1-m} \hat{B}_{m+j-1}\hat{B}_j^*\hat{\boldsymbol{h}}_{mj}, \tag{2.16}$$

with

$$\begin{aligned} \hat{\boldsymbol{d}}_{mj} &= R_m\left[-\frac{\sqrt{T}}{2\pi}\delta_j^m C\left(\hat{\boldsymbol{q}}_{m-j+1}, \hat{\boldsymbol{q}}_j\right) - \alpha_{mj}\hat{\boldsymbol{q}}_m\right], \\ \hat{\boldsymbol{h}}_{mj} &= R_m\left[-\frac{\sqrt{T}}{\pi}\delta_j^{N+1-m} C(\hat{\boldsymbol{q}}_{m+j-1}, \hat{\boldsymbol{q}}_j^*) - \beta_{mj}\hat{\boldsymbol{q}}_m\right], \end{aligned} \tag{2.17}$$

and where we defined $\alpha_{mj} = \Theta_{mj}/\{\hat{\boldsymbol{a}}_m, \hat{\boldsymbol{q}}_m\}$ and $\beta_{mj} = \Xi_{mj}/\{\hat{\boldsymbol{a}}_m, \hat{\boldsymbol{q}}_m\}$. The fields $\hat{\boldsymbol{d}}_{mj}$ and $\hat{\boldsymbol{h}}_{mj}$ verify $\{\hat{\boldsymbol{q}}_m, \hat{\boldsymbol{d}}_{mj}\} = \{\hat{\boldsymbol{q}}_m, \hat{\boldsymbol{h}}_{mj}\} = 0$, implying $\{\hat{\boldsymbol{q}}_m, \hat{\boldsymbol{u}}_{2,m}\} = 0$; therefore, each Fourier component generated at second order (i.e $O(\epsilon_o)$) is orthogonal to the Fourier component of the optimal linear solution at the same frequency. The partial derivative of $\hat{\boldsymbol{u}}_{2,m}$ with respect to τ_1 follows directly from (2.16):

$$\begin{aligned}\partial_{\tau_1}\hat{\boldsymbol{u}}_{2,n} &= \sum_{k=1}^{n}(\partial_{\tau_1}\hat{B}_{n-k+1})\hat{B}_k\hat{\boldsymbol{d}}_{nk} + \sum_{k=1}^{n}\hat{B}_{n-k+1}(\partial_{\tau_1}\hat{B}_k)\hat{\boldsymbol{d}}_{nk} \\ &+ \sum_{k=1}^{N+1-n}(\partial_{\tau_1}\hat{B}_{n+k-1})\hat{B}_k^*\hat{\boldsymbol{h}}_{nk} + \sum_{k=1}^{N+1-n}\hat{B}_{n+k-1}(\partial_{\tau_1}\hat{B}_k^*)\hat{\boldsymbol{h}}_{nk}.\end{aligned} \tag{2.18}$$

Since $\partial_{\tau_1}\hat{B} = \hat{g}_2[\hat{B}]/\{\hat{\boldsymbol{a}},\hat{\boldsymbol{q}}\}$, we can again use (2.12) and

$$\partial_{\tau_1}\hat{B}_m = \sum_{j=1}^{m}\hat{B}_{m-j+1}\hat{B}_j\alpha_{mj} + \sum_{j=1}^{N+1-m}\hat{B}_{m+j-1}\hat{B}_j^*\beta_{mj}. \tag{2.19}$$

Evaluating (2.19) in $m = n-k+1$, $m = n+k-1$ and $m = k$ yields, respectively:

$$\partial_{\tau_1}\hat{B}_{n-k+1} = \sum_{j=1}^{n-k+1}\hat{B}_{n-k-j+2}\hat{B}_j\alpha_{n-k+1,j} + \sum_{j=1}^{N-n+k}\hat{B}_{n-k+j}\hat{B}_j^*\beta_{n-k+1,j}, \tag{2.20}$$

$$\partial_{\tau_1}\hat{B}_{n+k-1} = \sum_{j=1}^{n+k-1}\hat{B}_{n+k-j}\hat{B}_j\alpha_{n+k-1,j} + \sum_{j=1}^{N-n-k+2}\hat{B}_{n+k+j-2}\hat{B}_j^*\beta_{n+k-1,j}, \tag{2.21}$$

and

$$\partial_{\tau_1}\hat{B}_k^* = \sum_{j=1}^{k}\hat{B}_{k-j+1}^*\hat{B}_j^*\alpha_{kj}^* + \sum_{j=1}^{N+1-k}\hat{B}_{k+j-1}^*\hat{B}_j\beta_{kj}^*. \tag{2.22}$$

After injecting (2.20),(2.21) and (2.22) in (2.18), and projecting on the adjoint, we end up on

$$\begin{aligned}\{\hat{\boldsymbol{a}}_n,\partial_{\tau_1}\hat{\boldsymbol{u}}_{2,n}\} &= \sum_{k=1}^{n}\hat{B}_k\left(\sum_{j=1}^{n-k+1}\hat{B}_{n-k-j+2}\hat{B}_j\mathcal{G}_{nkj} + \sum_{j=1}^{N-n+k}\hat{B}_{n-k+j}\hat{B}_j^*\mathcal{H}_{nkj}\right) \\ &+ \sum_{k=1}^{n}\hat{B}_{n-k+1}\left(\sum_{j=1}^{k}\hat{B}_{k-j+1}\hat{B}_j\mathcal{I}_{nkj} + \sum_{j=1}^{N+1-k}\hat{B}_{k+j-1}\hat{B}_j^*\mathcal{J}_{nkj}\right) \\ &+ \sum_{k=1}^{N+1-n}\hat{B}_k^*\left(\sum_{j=1}^{n+k-1}\hat{B}_{n+k-j}\hat{B}_j\mathcal{K}_{nkj} + \sum_{j=1}^{N-n-k+2}\hat{B}_{n+k+j-2}\hat{B}_j^*\mathcal{L}_{nkj}\right) \\ &+ \sum_{k=1}^{N+1-n}\hat{B}_{n+k-1}\left(\sum_{j=1}^{k}\hat{B}_{k-j+1}^*\hat{B}_j^*\mathcal{M}_{nkj} + \sum_{j=1}^{N+1-k}\hat{B}_{k+j-1}^*\hat{B}_j\mathcal{N}_{nkj}\right),\end{aligned} \tag{2.23}$$

where we defined the following third-order tensors

$$
\begin{aligned}
\mathcal{G}_{nkj} &= \alpha_{n-k+1,j}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{d}}_{n,k}\}, \quad 1 \leqslant j \leqslant n-k+1, \quad 1 \leqslant k \leqslant n \\
\mathcal{H}_{nkj} &= \beta_{n-k+1,j}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{d}}_{n,k}\}, \quad 1 \leqslant j \leqslant N-n+k, \quad 1 \leqslant k \leqslant n \\
\mathcal{I}_{nkj} &= \alpha_{kj}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{d}}_{n,k}\}, \quad 1 \leqslant j \leqslant k, \quad 1 \leqslant k \leqslant n \\
\mathcal{J}_{nkj} &= \beta_{kj}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{d}}_{n,k}\}, \quad 1 \leqslant j \leqslant N+1-k, \quad 1 \leqslant k \leqslant n \\
\mathcal{K}_{nkj} &= \alpha_{n+k-1,j}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{h}}_{n,k}\}, \quad 1 \leqslant j \leqslant n+k-1, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{L}_{nkj} &= \beta_{n+k-1,j}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{h}}_{n,k}\}, \quad 1 \leqslant j \leqslant N-n-k+2, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{M}_{nkj} &= \alpha^*_{kj}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{h}}_{n,k}\}, \quad 1 \leqslant j \leqslant k, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{N}_{nkj} &= \beta^*_{kj}\{\hat{\boldsymbol{a}}_n, \hat{\boldsymbol{h}}_{n,k}\}, \quad 1 \leqslant j \leqslant N+1-k, \quad 1 \leqslant k \leqslant N+1-n.
\end{aligned}
\tag{2.24}
$$

Discretisation of $\hat{g}_3[\hat{B}]$:

We recall that

$$\hat{g}_3[B] = -\{\hat{\boldsymbol{a}}, \mathcal{F}\left[2C(\mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_1\right], \mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_2\right])\right]\}. \tag{2.25}$$

Using again (2.10) leads to the following discretisation

$$
\begin{aligned}
&(\mathcal{F}\left[2C(\mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_1\right], \mathcal{F}^{-1}\left[\hat{\boldsymbol{u}}_2\right])\right])_n = \\
&\frac{\sqrt{T}}{\pi}\left[\sum_{k=1}^{n} \delta^n_k C(\hat{\boldsymbol{u}}_{1,n-k+1}, \hat{\boldsymbol{u}}_{2,k}) + \sum_{k=1}^{N+1-n} \delta^{N+1-n}_k \left[C(\hat{\boldsymbol{u}}^*_{1,k}, \hat{\boldsymbol{u}}_{2,n+k-1}) + C(\hat{\boldsymbol{u}}_{1,n+k-1}, \hat{\boldsymbol{u}}^*_{2,k})\right]\right].
\end{aligned}
\tag{2.26}
$$

In addition, using (2.16) and $\hat{\boldsymbol{u}}_{1,i} = \hat{B}_i \hat{\boldsymbol{q}}_i$, we can further express

$$
\begin{aligned}
C(\hat{\boldsymbol{u}}_{1,n-k+1}, \hat{\boldsymbol{u}}_{2,k}) &= C\left(\hat{B}_{n-k+1}\hat{\boldsymbol{q}}_{n-k+1}, \sum_{j=1}^{k} \hat{B}_{k-j+1}\hat{B}_j \hat{\boldsymbol{d}}_{kj} + \sum_{j=1}^{N+1-k} \hat{B}_{k+j-1}\hat{B}^*_j \hat{\boldsymbol{h}}_{kj}\right) \\
&= \hat{B}_{n-k+1}\left[\sum_{j=1}^{k} \hat{B}_{k-j+1}\hat{B}_j C\left(\hat{\boldsymbol{q}}_{n-k+1}, \hat{\boldsymbol{d}}_{kj}\right) + \sum_{j=1}^{N+1-k} \hat{B}_{k+j-1}\hat{B}^*_j C\left(\hat{\boldsymbol{q}}_{n-k+1}, \hat{\boldsymbol{h}}_{kj}\right)\right],
\end{aligned}
\tag{2.27}
$$

as well as

$$
\begin{aligned}
C(\hat{\boldsymbol{u}}^*_{1,k}, \hat{\boldsymbol{u}}_{2,n+k-1}) &= C\left(\hat{B}^*_k \hat{\boldsymbol{q}}^*_k, \sum_{j=1}^{n+k-1} \hat{B}_{n+k-j}\hat{B}_j \hat{\boldsymbol{d}}_{n+k-1,j} + \sum_{j=1}^{N-n-k+2} \hat{B}_{n+k+j-2}\hat{B}^*_j \hat{\boldsymbol{h}}_{n+k-1,j}\right) \\
&= \hat{B}^*_k\left[\sum_{j=1}^{n+k-1} \hat{B}_{n+k-j}\hat{B}_j C\left(\hat{\boldsymbol{q}}^*_k, \hat{\boldsymbol{d}}_{n+k-1,j}\right) + \sum_{j=1}^{N-n-k+2} \hat{B}_{n+k+j-2}\hat{B}^*_j C\left(\hat{\boldsymbol{q}}^*_k, \hat{\boldsymbol{h}}_{n+k-1,j}\right)\right],
\end{aligned}
\tag{2.28}
$$

and eventually

$$
\begin{aligned}
C(\hat{\boldsymbol{u}}_{1,n+k-1}, \hat{\boldsymbol{u}}^*_{2,k}) &= C\left(\hat{B}_{n+k-1}\hat{\boldsymbol{q}}_{n+k-1}, \sum_{j=1}^{k} \hat{B}^*_{k-j+1}\hat{B}^*_j\hat{\boldsymbol{d}}^*_{kj} + \sum_{j=1}^{N+1-k} \hat{B}^*_{k+j-1}\hat{B}_j\hat{\boldsymbol{h}}^*_{kj}\right) \\
&= \hat{B}_{n+k-1}\left[\sum_{j=1}^{k} \hat{B}^*_{k-j+1}\hat{B}^*_j C\left(\hat{\boldsymbol{q}}_{n+k-1}, \hat{\boldsymbol{d}}^*_{kj}\right) + \sum_{j=1}^{N+1-k} \hat{B}^*_{k+j-1}\hat{B}_j C\left(\hat{\boldsymbol{q}}_{n+k-1}, \hat{\boldsymbol{h}}^*_{kj}\right)\right].
\end{aligned}
\tag{2.29}
$$

This results in the following discretisation for $\hat{g}_3[B]$:

$$
\begin{aligned}
\hat{g}_{3,n} = &\sum_{k=1}^{n} \hat{B}_{n-k+1}\left[\sum_{j=1}^{k} \hat{B}_{k-j+1}\hat{B}_j\mathcal{A}_{nkj} + \sum_{j=1}^{N+1-k} \hat{B}_{k+j-1}\hat{B}^*_j\mathcal{B}_{nkj}\right] \\
&+ \sum_{k=1}^{N+1-n} \hat{B}^*_k\left[\sum_{j=1}^{n+k-1} \hat{B}_{n+k-j}\hat{B}_j\mathcal{C}_{nkj} + \sum_{j=1}^{N-n-k+2} \hat{B}_{n+k+j-2}\hat{B}^*_j\mathcal{D}_{nkj}\right] \\
&+ \sum_{k=1}^{N+1-n} \hat{B}_{n+k-1}\left[\sum_{j=1}^{k} \hat{B}^*_{k-j+1}\hat{B}^*_j\mathcal{E}_{nkj} + \sum_{j=1}^{N+1-k} \hat{B}^*_{k+j-1}\hat{B}_j\mathcal{F}_{nkj}\right],
\end{aligned}
\tag{2.30}
$$

where we defined the following third-order tensors

$$
\begin{aligned}
\mathcal{A}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^n_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n-k+1}, \hat{\boldsymbol{d}}_{kj})\}, \quad 1 \leqslant j \leqslant k, \quad 1 \leqslant k \leqslant n \\
\mathcal{B}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^n_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n-k+1}, \hat{\boldsymbol{h}}_{kj})\}, \quad 1 \leqslant j \leqslant N+1-k, \quad 1 \leqslant k \leqslant n \\
\mathcal{C}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^{N+1-n}_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}^*_k, \hat{\boldsymbol{d}}_{n+k-1,j})\}, \quad 1 \leqslant j \leqslant n+k-1, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{D}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^{N+1-n}_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}^*_k, \hat{\boldsymbol{h}}_{n+k-1,j})\}, \quad 1 \leqslant j \leqslant N-n-k+2, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{E}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^{N+1-n}_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n+k-1}, \hat{\boldsymbol{d}}^*_{kj})\}, \quad 1 \leqslant j \leqslant k, \quad 1 \leqslant k \leqslant N+1-n \\
\mathcal{F}_{nkj} &= -\frac{\sqrt{T}}{\pi}\delta^{N+1-n}_k\{\hat{\boldsymbol{a}}_n, C(\hat{\boldsymbol{q}}_{n+k-1}, \hat{\boldsymbol{h}}^*_{kj})\}, \quad 1 \leqslant j \leqslant N+1-k, \quad 1 \leqslant k \leqslant N+1-n.
\end{aligned}
\tag{2.31}
$$

As a summary, upon the choice of ϕ and $\hat{\boldsymbol{\xi}}$ we are led to solve

$$
\begin{aligned}
&r(\hat{B}; \phi, \hat{\xi}) = 0, \quad \text{with} \\
&r(\hat{B}; \phi, \hat{\xi}) \doteq \hat{g}_2[\hat{B}] + \sqrt{\epsilon_o}\left[\gamma(\phi\hat{\xi} - \hat{B}) + \hat{g}_3[\hat{B}] - \{\hat{\boldsymbol{a}}, \partial_{\tau_1}\hat{\boldsymbol{u}}_2\}\right],
\end{aligned}
\tag{2.32}
$$

where we defined $\gamma = \{\hat{\boldsymbol{a}}, \boldsymbol{f}_o\}$. At the discrete level, this amounts to solving for a system of N nonlinearly coupled equations for the N unknowns $\hat{B}_n$ ($n = 1, 2, ..., N$):

$$
r_n = \hat{g}_{2,n} + \sqrt{\epsilon_o}\left[\gamma_n(\phi\hat{\xi}_n - \hat{B}_n) + \hat{g}_{3,n} - \{\hat{\boldsymbol{a}}_n, \partial_{\tau_1}\hat{\boldsymbol{u}}_{2,n}\}\right] = 0, \quad n = 1, 2, ..., N, \tag{2.33}
$$

where $\hat{g}_{2,n}$, $\hat{g}_{3,n}$ and $\{\hat{\boldsymbol{a}}_n, \partial_{\tau_1}\hat{\boldsymbol{u}}_{2,n}\}$ are evaluated using respectively the simple sum (2.12) and the double sums (2.30) and (2.23). This suggests the following procedure:

Algorithm :

(i) Choose the values for N and ω_c, which sets the discretisation of the frequencies.

(ii) Over the discrete set of frequencies, pre-compute once for all the *determinitic* fields $\boldsymbol{f}_o$, $\hat{\boldsymbol{q}}$, $\hat{\boldsymbol{a}}$, the scalar γ, and the tensors Θ, Ξ, $\mathcal{A}$,$\mathcal{B}$,$\mathcal{C}$,$\mathcal{D}$,$\mathcal{E}$,$\mathcal{F}$,$\mathcal{G}$,$\mathcal{H}$,$\mathcal{I}$,$\mathcal{J}$,$\mathcal{K}$,$\mathcal{L}$,$\mathcal{M}$,$\mathcal{N}$.

(iii) Choose a value for ϕ (which sets the forcing amplitude).

(iv) Draw randomly a white noise ($|\hat{\xi}(\omega)| = 1\ \forall\omega$, but random phases uniformly distributed between 0 and 2π).

(v) Solve the system (2.33) by means of a nonlinear solver, for instance `fsolve` on `MATLAB`.

(vi) Update the statistics on $\hat{B}$, for instance its ensemble average, and, if not converged go back to (iv).

Of course the convergence in terms of N and ω_c must be ensured.

3. The particular case of the NSE: numerical implementation

The linear and nonlinear NSE are solved for (u_x,u_y,p) by means of the Finite Element Method with Taylor-Hood (P2, P2, P1) elements, respectively, after implementation of their weak form in the software FreeFem++. The steady solution of the nonlinear NSE is found using the iterative Newton–Raphson method, and the linear operators are built thanks to a sparse solver available in FreeFem++. The optimal forcing structure $\boldsymbol{f}_o$ is found on the software `MATLAB` after discretizing the integral expression for B^∞ and performing the lower-upper (LU) decomposition of the resolvent operators to speed-up their application. Finally, DNS are performed in FreeFem++ by applying a time scheme based on the characteristic–Galerkin method. We refer to Mantic-Lugo & Gallaire (2016) for the validation of the codes with existing literature when possible and for the mesh convergence analysis, since the same codes have been used.

SUPPLEMENTARY MATERIALS REFERENCES

Mantic-Lugo, V. & Gallaire, F. 2016 Self-consistent model for the saturation mechanism of the response to harmonic forcing in the backward-facing step flow. *J. Fluid Mech.* **793**, 777–97.