Computation of temporal and spatial stability for parallel flows

Jérôme Hœpffner

In this note, I discuss how to compute in Matlab the temporal and spatial stability of a parallel flow. The code is available on my home page at www.fukagata.mech.keio.ac.jp/∼jerome/web/codes.php, treating the stability of the mixing layer with an hyperbolic tangent base profile. The presentation is made such that it is easy to see the similarities between the two types of stability analyses.

The Navier-Stokes equations linearized about the base flow profile U in two-dimensions is

$$\begin{aligned} u_t + Uu_x + U'v &= -p_x + \Delta u/Re, \\ v_t + Uv_x &= -p_y + \Delta v/Re, \\ u_x + v_y &= 0 \end{aligned}$$

where subscripts x, y, t denote partial differentiation, U' is the y derivative of the base flow profile and Δ is the Laplacian. Rewriting this equation in operator form, we have

$$0 = \left[- \begin{pmatrix} \partial_t & 0 & 0 \\ 0 & \partial_t & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} -U\partial_x + \Delta/Re & U' & -\partial_x \\ 0 & -U\partial_x + \Delta/Re & -\partial_y \\ \partial_x & \partial_y & 0 \end{pmatrix} \right] \begin{pmatrix} u \\ v \\ p \end{pmatrix}$$

where $\partial_{\{x,y,t\}}$ denote the partial derivative operators.

We now consider *normal modes*, that is we assume an oscillating behaviour in x and time of the flow solution

$$u = \hat{u}(y) \mathrm{e}^{ikx - i\omega t} \tag{1}$$

where $\hat{u}$ is the amplitude function of u in y, k is the spatial wavenumber (in x) and ω is the temporal pulsation. The exponential structure in (1) allows the solution to oscillate and grow/decay in space and time, depending on the real and imaginary parts of k and ω. In the *temporal* analysis, the solution grows/decay and oscillates in time but only oscillate in space: $k \in \mathbb{R}$ is given, and one obtains ω from the dynamic equations. In the *spatial* analysis, one assume that the solution only oscillates in time at a given spatial position, but is allowed to grow/decay and oscillate in space: $\omega \in \mathbb{R}$ is given, and k is obtained from the dynamics equations.

Injecting (1) in the dynamic equations, we can replace

$$\begin{aligned} \partial_t &\to -i\omega, \\ \partial_x &\to ik, \\ \partial_{xx} &\to -k^2. \end{aligned}$$

We obtain

$$0 = \Bigg[-\omega \underbrace{\begin{pmatrix} -i & 0 & 0 \\ 0 & -i & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{E} + \overbrace{\underbrace{\begin{pmatrix} \partial_{yy}/Re & U' & 0 \\ 0 & \partial_{yy}/Re & -D \\ 0 & D & 0 \end{pmatrix}}_{A_{00}} + k \underbrace{\begin{pmatrix} -iU & 0 & -i \\ 0 & -iU & 0 \\ i & 0 & 0 \end{pmatrix}}_{A_1} + k^2 \underbrace{\begin{pmatrix} -I/Re & 0 & 0 \\ 0 & -I/Re & 0 \\ 0 & 0 & 0 \end{pmatrix}}_{A_2}}^{A} \Bigg] \underbrace{\begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{pmatrix}}_{\hat{q}}$$

where we have explicitely decomposed into operators that multiply ω and the different powers of k. This equation is the *dispersion relation*. If k is given, ω is the eigenvalue of a generalized eigenvalue problem

$$\omega E \hat{q} = A \hat{q},$$

whereas if ω is given, then k is the eigenvalue of a polynomial eigenvalue problem

$$0 = (A_0 + kA_1 + k^2 A_2)\hat{q}, \tag{2}$$

with $A_0 = -\omega E + A_{00}$. In Matlab, this last equation can be solved as such using the function `polyeig`, with the code `k=polyeig(A0,A1,A2)`. In general, the direct solution of polynomial eigenvalue problems can be heavy. For

the present case, we can transform the polynomial eigenvalue problem into a generalized eigenvalue problem, to be quickly solved with `eig` or `eigs`. To do this, we augment the system with the variable $\hat{h} = k\hat{q}$

$$0 = \underbrace{\begin{pmatrix} A_0 & 0 \\ 0 & I \end{pmatrix}}_{\mathcal{A}_0} \begin{pmatrix} \hat{q} \\ \hat{h} \end{pmatrix} + k \underbrace{\begin{pmatrix} A_1 & A_2 \\ -I & 0 \end{pmatrix}}_{\mathcal{A}_1} \begin{pmatrix} \hat{q} \\ \hat{h} \end{pmatrix}$$

where the first row is (2) and the second row enforces the definition of $\hat{h}$. In Matlab the solution of this equation is written `k=eig(AA0,-AA1)`.

We can simplify a little further this expression since k^2 does not multiply the pressure. We only need to augment the system with $k\hat{u}$ and $k\hat{v}$, to obtain the final system

$$0 = \left[\underbrace{\left(\begin{array}{cc|c} A_0 & & \\ & & \\ \hline & I & \\ & & I \end{array}\right)}_{\mathcal{A}_0} + k \underbrace{\left(\begin{array}{cc|c} & & -I/Re \\ A_1 & & -I/Re \\ \hline -I & & \\ & -I & \end{array}\right)}_{\mathcal{A}_1} \right] \cdot \left(\begin{array}{c} \hat{u} \\ \hat{v} \\ \hline \hat{p} \\ \hline k\hat{u} \\ k\hat{v} \end{array}\right)$$

which we solve using `eig` for the eigenvalues k and the associated eigenvectors.