POD, POD-ROM for Burgers

Laurent CORDIER

Laurent.Cordier@univ-poitiers.fr

PPRIME Institute

Poitiers, France

▷ Solve equation

$$\frac{\partial u}{\partial t} = \frac{1}{\text{Re}} \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} \quad \forall x \in]0;1[\quad \text{and} \quad t \in]0;T[$$

with

$$\begin{aligned} u(x,0) &= \sin(\pi x) & \forall x \in]0;1[\quad (IC) \\ u(0,t) = u(1,t) &= 0 & \forall t \in]0;T] \quad (BC) \end{aligned}$$

▷ Analytical solution

$$u_a(x,t) = \frac{2\pi}{\text{Re}} \frac{\sum_{n=1}^{\infty} a_n \, n \, \sin(n\pi x) \exp\left(-n^2\pi^2 t/\text{Re}\right)}{a_0 + \sum_{n=1}^{\infty} a_n \, \cos(n\pi x) \, \exp\left(-n^2\pi^2 t/\text{Re}\right)}$$

where a_n are Fourier coefficients.

Application to the 1D Burgers equation

- Numerical parameters for the POD analysis
  - $\mathrm{Re} = 10$,
  - $T = 0.1$ and $\Delta t = 10^{-4}$ i.e. $N_t = 1000$ snapshots in the data base,
  - $x \in [0; 1]$ and $\Delta x = \frac{1-0}{N_x - 1}$ with $N_x = 100$.

Matlab

- Application of the Snapshot POD
  Let $\boldsymbol{u}(\boldsymbol{x}, t_k)$, $k = 1, \cdots, N_t$ be N_t "snapshots".
- Algorithm:
  1. We decompose $\boldsymbol{u}(\boldsymbol{x}, t)$ as:

$$\boldsymbol{u}(\boldsymbol{x}, t_k) = \boldsymbol{u_m}(\boldsymbol{x}) + \boldsymbol{v}(\boldsymbol{x}, t_k) \quad k = 1, \cdots, N_t$$

  where :

$$\boldsymbol{u_m}(\boldsymbol{x}) = \frac{1}{N_t} \sum_{k=1}^{N_t} \boldsymbol{u}(\boldsymbol{x}, t_k)$$

  2. We build the time correlation matrix:

$$C_{kl} = \frac{1}{N_t} \left(\boldsymbol{v}(\boldsymbol{x}, t_k), \boldsymbol{v}(\boldsymbol{x}, t_l) \right)_\Omega = \frac{1}{N_t} \int_\Omega \boldsymbol{v}(\boldsymbol{x}, t_k) \cdot \boldsymbol{v}(\boldsymbol{x}, t_l) \, \mathrm{d}\boldsymbol{x}$$

  3. We solve the eigenvalue problem $C\boldsymbol{A} = \lambda \boldsymbol{A} \Longrightarrow \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_t} \geq 0$ and $\boldsymbol{A}_1(t_k), \cdots, \boldsymbol{A}_{N_t}(t_k)$.

$$\boldsymbol{A}_1 = \begin{bmatrix} A_1(t_1) \\ A_1(t_2) \\ \vdots \\ A_1(t_{N_t}) \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} A_2(t_1) \\ A_2(t_2) \\ \vdots \\ A_2(t_{N_t}) \end{bmatrix}, \quad \cdots, \quad \boldsymbol{A}_{N_t} = \begin{bmatrix} A_{N_t}(t_1) \\ A_{N_t}(t_2) \\ \vdots \\ A_{N_t}(t_{N_t}) \end{bmatrix}$$

4. We obtain the POD eigenvectors $\boldsymbol{\Phi}_n(\boldsymbol{x}),\ n = 1, \cdots, N_t$ using:

$$\boldsymbol{\Phi}_n(\boldsymbol{x}) = \frac{1}{N_t \lambda_n} \sum_{k=1}^{N_t} A_n(t_k) \boldsymbol{v}(\boldsymbol{x}, t_k)$$

5. We check the orthogonality/orthonormality relations *i*.e.

$$\frac{1}{N_t} \sum_{k=1}^{N_t} A_n(t_k) A_m(t_k) = \lambda_n \delta_{nm} \quad ; \quad \int_\Omega \boldsymbol{\Phi}_n(\boldsymbol{x}) \cdot \boldsymbol{\Phi}_m(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \delta_{nm}$$

POD Reduced-order model

- We decompose the velocity fields on N_{POD} modes:

$$u(x,t) = u_m(x) + \sum_{k=1}^{N_{\text{POD}}} a_k(t)\Phi_k(x).$$

- Dynamical system with N_{gal} $(\ll N_{\text{POD}})$ modes kept:

$$\frac{d\,a_i(t)}{d\,t} = \mathcal{A}_i + \sum_{j=1}^{N_{\text{gal}}} \mathcal{B}_{ij}\, a_j(t) + \sum_{j=1}^{N_{\text{gal}}} \sum_{k=1}^{N_{\text{gal}}} \mathcal{C}_{ijk}\, a_j(t) a_k(t)$$

$$a_i(0) = (u(x,\,0) - u_m(x),\;\Phi_i(x))_\Omega.$$

$\mathcal{A}_i, \mathcal{B}_{ij}, \mathcal{C}_{ijk}$ depend only on $\boldsymbol{\Phi}$, $\boldsymbol{u_m}$ and Re.

- Dynamics predicted by the POD ROM may be not sufficiently accurate
$\Longrightarrow$ need of identification techniques

POD Reduced-order model

$$\mathcal{A}_i = -\left(\Phi_i, u_m \frac{\partial u_m}{\partial x}\right)_\Omega + \frac{1}{\mathsf{Re}}\left(\Phi_i, \frac{\partial^2 u_m}{\partial x^2}\right)_\Omega$$

$$\mathcal{B}_{ij} = -\left(\Phi_i, u_m \frac{\partial \Phi_j}{\partial x}\right)_\Omega - \left(\Phi_i, \Phi_j \frac{\partial u_m}{\partial x}\right)_\Omega + \frac{1}{\mathsf{Re}}\left(\Phi_i, \frac{\partial^2 \Phi_j}{\partial x^2}\right)_\Omega$$

$$\mathcal{C}_{ijk} = -\left(\Phi_i, \Phi_j \frac{\partial \Phi_k}{\partial x}\right)_\Omega$$