INSA Toulouse, Applied Mathematics departement
Finte Element Modeling

Model reduction by POD and POD-NN methods
Offline-online strategy
Project

Let us consider the following μ-parametrized stationary advection-diffusion model in 2D :

$$\begin{cases} -div\,(\lambda(\mu_1)\,\nabla u) + \mathbf{w}\cdot\nabla u = f(\mu_2) & \text{in } \Omega \\ u = g & \text{on } \Gamma_{in} \\ -\lambda(\mu_1)\,\nabla u\cdot n = 0 & \text{on } \Gamma_{wall} \\ -\lambda(\mu_1)\,\nabla u\cdot n = 0 & \text{on } \Gamma_{out} \end{cases} \tag{1}$$

Where : u is the unknown scalar field ; $\mathbf{w}$ is a given velocity field.

The boundary of Ω is decomposed as follows : $\partial\Omega = \Gamma_{in} \cup \Gamma_{wall} \cup \Gamma_{out}$.

***

Problem (1) is parametrized by the diffusivity parameter $\lambda(\mu_1)$ and the source terme $f(\mu_2)$ such that :

— $\lambda(\mu_1) = \exp(\mu_1 - 11)$ (non-affinely parametrized PDE).

— $f(\mu_2) = A\cos(\mu_2 L x)$, ($x$ is the space variable).

with $\mu_1 \in [\mu_{\min}, \mu_{\max}]$, $\mu_{\min} = 1$ and $\mu_{\max} = 10$. $A = 10$ being a scalar amplitude, L the length of the domain Ω_h (here $L = 2$), and $\mu_2 \in \left[0, \frac{\pi}{L}\right]$.

The goal is to solve this BVP in real-time, therefore in reduced basis, for input parameter values $\mu = (\mu_1, \mu_2) \in \mathbb{R}^2$.

To start this project, use the Python-FEniCS code provided as part of the practical session (available on the course Moodle page).

1 The POD reduced basis method

1.0.1 Prolem property

a) Based on the expression of the diffusivity $\lambda(\mu)$ provided above, determine the nature of the resulting PDE :

Is the problem *affinely or non-affinely parametrized*?
Is it a *linear or non-linear* problem?

1.0.2 Offline phase

1. Plot the parameter space in 2D by computing a chosen number for μ_1 and μ_2. Analyze the results.
2. Plot a 3D subset of the solution manifold $\mathcal{M}_\mu$. Analyze the results.
3. Compute The corresponding snapshots and store them in a matrix $\mathbb{S} \in \mathbb{R}^{N_h \times N_s}$, with N_h the FE space dimension and N_s the snapshots number.
4. Compute the correlation matrix $\mathbb{C}$ and its eigenmodes(eigenvalues and eigen vectors).
5. plot the normalized eigenvalues in decreasing order and analyse the results.
6. Compute the reduced dimension N_{rb} for a tolerance ε_{POD} and truncate the reduced matrix $\mathbb{B}_{rb}$.
7. Plot the reduced dimension N_{rb} in function of tolerance ε_{POD} (for different tolerances ε_{POD}).
8. Show numerically that the error estimation satisfied by the POD holds.

1.0.3 Online phase

1. Chose a value fo the parameters μ_1 and μ_2.
2. Assembly the stiffness matrix and the RHS and solve the FE system $\mathbb{A}_h^\mu \mathbf{u}_h(\mu) = \mathbf{F}_h^\mu$.
3. Construct and solve the reduced system.
4. Compute the CPU time for the FE system and the reduced system. Analyze the results.
5. Plot the FE solution, the reduced basis solution and the error map.
6. Compare the result obtained by the L^2, H^1 norm in a table.

2 The hybrid POD-NN method

2.1 Offline phase

1. Load the data from numpy files, compute the reduced outputs normalize the data, create Neural network and train it.

2. plot the mean square error (mse) of the loss function.

2.2 Online phase

1. Predict a solution for the same parameter μ_{new} as for the POD method in Section 1.
2. Compare the obtained solution with FE and POD ones.
3. Compute the L^2 error and analyse the results.