INSA Toulouse, Applied Mathematics departement
Finte Element Modeling

Model reduction by the POD method and a hybrid POD-DNN method

Programming Practical with FEniCS

Let us consider the same BVP as previously. It is based on the 2D stationary advection-diffusion. It is here parametrized by the diffusivity coefficient $\lambda(\mu)$. The equations read :

$$\begin{cases} -div(\lambda(\mu)\nabla u) + \mathbf{w}\nabla u = f & in \ \ \Omega \\ u = g & in \ \ \Gamma_{in} \\ -\lambda(\mu)\nabla u.n = 0 & in \ \ \Gamma_{wall} \\ -\lambda(\mu)\nabla u.n = 0 & in \ \ \Gamma_{out} \end{cases} \tag{1}$$

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

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

***

Problem (1) is parametrized by the diffusivity parameter $\lambda(\mu)$. The latter may be defined for example as follows :

. $\lambda(\mu) = \mu_0(\mu + 1)$ (affinely parametrized PDE),

. $\lambda(\mu) = \exp(\mu_0(\mu + 1))$ (non-affinely parametrized PDE).

with μ_0 a constant, $\mu_0 > 0$.

μ is supposed to belong to the interval $[\mu_{min}, \mu_{max}]$.

The parameter space $\mathbf{P}_h \in \mathbb{R}^M$ is defined as :

$$\mathbf{P}_h = \{\mu_1, \cdots, \mu_M\} \tag{2}$$

Considering the snapshots (HR FE solutions) obtained from $\mathbf{P}_h$, the goal is to solve the BVP in real-time given a new parameter value μ.

1 POD method

1) Problem property

Given the diffusivity $\lambda(\mu)$, is the resulting BVP
affinely / non-affinely parametrized ? linear / non-linear ?

To start this practical, first upload the Python-Fenics code available on the course Moodle page.

2) Complete the code

a) Plot out a 3D part of the solutions manifold $\mathcal{M}_\mu$ both for the affinely parametrized case and the non-affinely parametrized case.

b) Complete the missing instructions in the code.
(They are indicated in the code by the sentence "To be completed").

c) Print out the CPU-time for :

i) the offline phase ; ii) the two critical sub-steps of the online phase.
(Python command : time.time())
Analyse the results.

Compare the CPU time required for one HR computation (on a fine grid) vs its reduced dimension approximation.

2 Hybrid POD-NN method

In this part, we investigate the POD-NN method studied during the course. The POD-NN method is here applied to the same linear BVP as before : Problem (1).
Recall that this hybrid POD-NN method can be similarly applied to a non-linear BVP.

a) Upload on the course Moodle page the pre-trained NN named *Neural-network.h5*.

b) Perform the NN to predict the solution $U_{rb-hybrid}(x)$ for a parameter value μ of your choice.

c) Compare $U_{rb-hybrid}(x)$ with the POD solution U_{rb} obtained at Stage 1. Discuss the results.

To go further

- Adapt the POD code to solve a parametrized linear BVP with a composite parameter $\mu = (\mu_0, \mu_1) \in \mathbb{R}^2$.
- Adapt the hybrid POD-NN method to a non-linear BVP.