# Projection-based Model Order Reduction techniques

Olivier Zahm[*]

Practical Session M2-SIAM, 2019.

## 1 Description of the model

Consider the problem of finding the function $u : \Omega \to \mathbb{R}$ which satisfies the following Partial Differential Equation (PDE)

$$\begin{aligned} -\nabla \cdot \kappa \nabla u + \alpha \cdot \nabla u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

The domain $\Omega = [0,1]^2$ is a square which is partitioned into four subdomains

$$\Omega = \quad \begin{array}{r|c|c|} \multicolumn{1}{r}{1} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \cline{2-3} & \Omega_1 & \Omega_4 \\ \cline{2-3} 0.5 & \Omega_2 & \Omega_3 \\ \cline{2-3} \multicolumn{1}{r}{0} & \multicolumn{1}{c}{0.5} & \multicolumn{1}{r}{1} \end{array}$$

The diffusion field $\kappa = \kappa(x)$ and the advection field $\alpha = \alpha(x)$ are parametrized by means of a 5-dimensional vector $x = (x_1, \ldots, x_5)$ as follow

$$\kappa(x) = \kappa(s,x) = \sum_{i=1}^{4} x_i 1_{\Omega_i}(s) \qquad \text{and} \qquad \alpha(x) = \alpha(s,x) = x_5 \begin{pmatrix} 1/2 - s_2 \\ s_1 - 1/2 \end{pmatrix}, \tag{2}$$

where $s = (s_1, s_2) \in \Omega$ denotes the spatial variable and $1_{\Omega_i}$ the indicator function[1] associated with $\Omega_i$. The parameter set is defined by

$$\mathcal{P} = \left\{ x \in \mathbb{R}^5 \ \text{ s.t. } \begin{array}{l} 0.05 \leq x_1, \ldots, x_4 \leq 1 \\ -100 \leq x_5 \leq 100 \end{array} \right\}.$$

When it is considered at random, the parameter $X$ is a random vector uniformly distributed over $\mathcal{P}$.

[*]olivier.zahm@inria.fr

[1]The indicator function $1_{\Omega_i}$ is such that $1_{\Omega_i}(s) = 1$ if $s \in \Omega_i$ and $1_{\Omega_1}(s) = 0$ if $s \notin \Omega_i$.

The source term $f = f(s)$ is equal to $f(s) = +1$ if $s \in [0.05, 0.2] \times [0.4, 0.6]$, $f(s) = -1$ if $s \in [0.05, 0.2] \times [0.4, 0.6]$ and $f(s) = 0$ elsewhere. The source term $f$ is independent $x$. Finally, the scalar quantity of interest $Y(x)$ is defined as the mean value of the solution on the subdomain $\Omega_3$, meaning

$$Y(x) = \frac{1}{|\Omega_3|} \int_{\Omega_3} u(x) \mathrm{d}\Omega. \tag{3}$$

## 1.1 Weak solution and finite element approximation

The weak formulation of (1) consists in finding $u = u(x) \in V = H_0^1(\Omega)$ such that

$$\underbrace{\int_\Omega \kappa(x) \nabla u(x) \cdot \nabla v \ \mathrm{d}\Omega + \int_\Omega \alpha(x) \cdot \nabla u(x)\, v \ \mathrm{d}\Omega}_{= a(u(x), v\,; x)} = \underbrace{\int_\Omega f v \ \mathrm{d}\Omega,}_{= \ell(v)} \tag{4}$$

for all $v \in V$. We recall that the weak formulation is obtained by multiplying (1) by a test function $v \in V$, by integrating over $\Omega$ and by doing an integration by part. The Finite Element Method (FEM) consists in replacing $V$ by a finite dimension space $V^h = \mathrm{span}\{\phi_1, \ldots, \phi_N\} \subset V$, $N \gg 1$, which consists of piecewise polynomial functions $\phi_i$ (sometimes called “hat” functions, or “shape” functions). With this discretization, the variational formulation (4) becomes an algebraic equation

$$A(x) u(x) = b,$$

where $A_{i,j}(x) = a(\phi_j, \phi_i\,; x)$, $b_i = \ell(\phi_i)$ and where $u(x)$ is now a vector containing the the coefficients of the approximation on the finite element basis $\{\phi_1, \ldots, \phi_N\}$. For more details on the FEM, see the book [1]. The quantity of interest $Y(x)$ defined by (3) can be expressed as a linear form of the solution vector $u(x)$ as

$$Y(x) = q^T u(x),$$

where $q \in \mathbb{R}^N$ is the vector defined by $q_i = \frac{1}{|\Omega_3|} \int_{\Omega_3} \phi_i \mathrm{d}\Omega$.

## 1.2 Affine decomposability

Using the parametrization (2) of $\kappa(x)$ and $\alpha(x)$, the bilinear form $a(\cdot, \cdot\,; x)$ can be written as

$$a(u(x), v\,; x) = \sum_{k=1}^{4} x_k \underbrace{\int_{\Omega_i} \nabla u(x) \cdot \nabla v \ \mathrm{d}\Omega}_{= a_k(u(x), v)} \ + \ x_5 \underbrace{\int_\Omega \begin{pmatrix} 1/2 - s_2 \\ s_1 - 1/2 \end{pmatrix} \cdot \nabla u(x)\, v \ \mathrm{d}\Omega,}_{a_5(u(x), v)}$$

which, after FEM discretization, yields the decomposition

$$A(x) = \sum_{k=1}^{5} x_k \, A_k, \tag{5}$$

where $(A_k)_{i,j} = a_k(\phi_j, \phi_i)$. Thus, $A(x)$ admits an affine decomposition with respect to the parameter

## 2 Tasks

The above described benchmark is implemented in the file `parametrizedPDE.m`. It is an object oriented code which uses Matlab© classes. The file `ProjectionBasedMOR.m` provides a tutorial to learn how to use this class. Read and run the `ProjectionBasedMOR.m` one section after the other to learn how to:

- instantiate an object: `model = parametrizedPDE();`
- draw a random parameter: `X = model.randX();`
- compute the a solution: `u = model.u(X);`
- compute the quantity of interest: `Y = model.q'*u;`
- plot the solution: `model.plotSol(u)`
- get the FEM (full) system: `model.A(X)` and `model.b`
- get the matrices of the affine decomposition (5):

$$A_1 = \texttt{model.K1}, \dots, \quad A_4 = \texttt{model.K4}, \quad A_5 = \texttt{model.Ad},$$

### Task#1: Offline-online efficient galerkin projection

Using the affine decomposition of the operator $A(x)$, implement the offline-online efficient algorithm presented in the Lecture Note, see Algorithm 2. Tasks:

- Observe the speedup of the online phase and the slowdown of the offline phase.
- Estimate the computation time of the offline phases and of the online phases (per sample evaluation) of Algorithm 1 and of Algorithm 2. After how many sample evaluations does Algorithm 2 becomes more interesting to use compared to Algorithm 1?
- How this evolves with respect to $r$? (repeat the previous analysis for, say, $r \in \{5, 10, 20, 50, 100\}$)

### Task#2: POD

In the `script.m` file, the reduced space $V_r$ (see the matrix `Vr`) is built in a naive way using random snapshots of the solution. A better way to build $V_r$ is to use the Proper Orthogonal Decomposition (POD) or the Reduced Basis (RB) method, see Sections 2 and 3 of the Lecture notes. Implement the POD (Algorithm 3). Hint: according to Remark 2.2, the function `[U,D,V]=svd(A)` can be useful...

- Compare the quality of the POD basis and the one of the naive algorithm. For instance, one can plot the $L^2$-error as a function of $r \in \{5, 10, 20, 50, 100\}$.
- Compare the $L^2$-error of the POD with the sum of the eigenvalues $\sqrt{\sum_{i \geq r+1} \hat{\sigma}_i^2}$, see Equation (7) in the Lecture notes.

### Task#3: Reduced basis

Implement and test the RB method (Algorithm 4). How does this compare to the POD in terms of number of solution evaluations? of reduced space dimension? of $L^2$ versus $L^\infty$ error?

## References

[1] T. J. Hughes, *The finite element method: linear static and dynamic finite element analysis*, Courier Corporation, 2012.