Model Order Reduction Techniques
Problem Set 2 — RB for Linear Affine Elliptic Problems

Christophe Prud'homme

1 Problem Statement — Design of a Thermal Fin

We consider the problem of designing a thermal fin described in Problem Set 1. In PS1 we looked at some thoeretical issues (weak formulation and optimization formulation, convergence of the reduced basis approximation) and derived the necessary reduced basis quantities, i.e., expressions for $A_N(\mu)$, F_N , and L_N . This problem set is devoted to implementing the reduced basis approximation and solving a simple design problem.

1.1 Part 1 – Reduced Basis Approximation

The point of departure for the reduced basis approximation is a high – dimensional finite element "truth" discretization. In the offline stage we require the finite element solution to build the reduced basis and we thus also need the FE matrices. In this problem set we skip the FE assembly step and provide all of the necessary data for use in Python (see Appendix 1).

We saw in class that the reduced basis solution $u_N(\mu) \in \mathbb{R}^N$ satisfies the set of $N \times N$ linear equations,

$$A_N(\mu)u_N(\mu) = F_N; \tag{1}$$

and that the outputis given by

$$T_{root\,N}(\mu) = L_N^T u_N(\mu). \tag{2}$$

We derived expressions for $A_N(\mu) \in \mathbb{R}^{N\times N}$ in terms of $A_N(\mu)$ and Z, $F_N \in \mathbb{R}^N$ in terms of F_N and Z, and $L_N \in \mathbb{R}^N$ in terms of L_N and Z; here Z is an $\mathcal{N} \times N$ matrix, the jth column of which is $u_N(\mu j)$ (the nodal values of $u_N(\mu j)$). Finally, it follows from affine parameter dependence that $A_N(\mu)$ can be expressed as

$$A_N(\mu) = \sum_{q=1}^{Q} \Theta^q(\mu) A_N^q. \tag{3}$$

The goal is to implement an offline/ online version of the reduced – basis method following the computational decomposition indicated below.

- Offline
  1. Choose N.
  2. Choose the sample S_N .
  3. Construct Z.
  4. Construct $A_N^q, q = 1, \ldots, Q; F_N;$ and L_N.
- Online
  1. Form $A_N(\mu)$ from (3).
  2. Solve $A_N(\mu)u_N(\mu) = F_N$.
  3. Evaluate the output $T_{root\,N}(\mu)$ from (2).

1 The idea is that the offline stage is done only once, generating a small datafile with the $A_N^q, q = 1, \ldots, Q$, F_N, and L_N; the on-line stage then accesses this datafile to provide real-time response to new μ queries. For the required off-line finite element calculations in this and the following questions, you should first use the coarse triangulation $\mathcal{T}_{h,\text{coarse}}$.
a) Show that the operation count for the on-line stage of your code is independent of $\mathcal{N}$. In particular show that the operation count (number of floating-point operations) for the on-line stage, for each new μ of interest, can be expressed as

$$c_1 N^{\gamma_1} + c_2 N^{\gamma_2} + c_3 N^{\gamma_3}, \tag{4}$$

for $c_1, c_2, c_3, \gamma_1, \gamma_2$, and γ_3 independent of N. Give values for the constants $c_1, c_2, c_3, \gamma_1, \gamma_2$, and γ_3.
b) We first consider a one parameter ($P = 1$) problem. To this end, we keep the Biot number fixed at $Bi = 0.1$ and assume that the conductivities of all fins are equivalent, i.e., $k_1 = k_2 = k_3 = k_4$, but are allowed to vary between 0.1 and 10 – we thus have $\mu \in D = [0.1, 10]$. The sample set S_N for $N_{max} = 8$ is given in the datafile `RB_sample.sample1`.

1. Generate the reduced basis "matrix" Z and all necessary reduced basis quantities. You have two options: you can use the solution "snapshots" directly in Z or perform a Gram-Schmidt orthonormalization to construct Z (Note that you require the X – inner product to perform Gram-Schmidt; here, we use $(\cdot, \cdot)_X = a(\cdot, \cdot; \mu)$, where $\mu = 1$ – all conductivities are 1 and the Biot number is 0.1). Calculate the condition number of $A_N(\mu)$ for $N = 8$ and for $\mu = 1$ and $\mu = 10$ with and without Gram – Schmidt orthonormalization. What do you observe? Solve the reduced basis approximation (where you use the snapshots directly in Z) for $\mu_1 = 0.1$ and $N = 8$. What is $u_N(\mu_1)$? How do you expect $u_N(\mu_2)$ to look like for $\mu_2 = 10.0$? What about $\mu_3 = 1.0975$? Solve the Gram – Schmidt orthonormalized reduced basis approximation for $\mu_1 = 0.1$ and $\mu 2 = 10$ for $N = 8$. What do you observe? Can you justify the result? For the remaining questions you should use the Gram – Schmidt orthonormalized reduced basis approximation.

2. Verify that, for $\mu = 1.5$ (recall that Biot is still fixed at 0.1) and $N = 8$, the value of the output is $T_{root\,N}(\mu) = 1.53107$.

3. We next introduce a regular test sample, $\Xi_{test} \subset D$, of size $ntest = 100$ (in Python you can simply use `linspace(0.1, 10, 100)` to generate Ξ_{test}). Plot the convergence of the maximum relative error in the energy norm $\max_{\mu \in \Xi_{test}} |||u(\mu) - u_N(\mu)|||_\mu / |||u(\mu)|||_\mu$ and the maximum relative output error $\max \mu \in \Xi_{test} |T_{root}(\mu) - T_{root}N(\mu)|/T_{root}(\mu)$ as a function of N (use the Python command `semilogy` for plotting).

4. Compare the average CPU time over the test sample required to solve the reduced basis online stage with direct solution of the FE approximation as a function of N.

5. What value of N do you require to achieve a relative accuracy in the output of 1%. What savings in terms of CPU time does this % correspond to?

6. Solve problems b) 3. to 5. using the medium and fine FE triangulation. Is the dependence on $\mathcal{N}$ as you would anticipate?

c) We now consider another one parameter ($P = 1$) problem. This time, we assume that the conductivities are fixed at $\{k_1, k_2, k_3, k_4\} = \{0.4, 0.6, 0.8, 1.2\}$, and that only the Biot number, Bi, is allowed to vary from 0.01 to 1. The sample set S_N for $N_{max} = 11$ is given in the datafile `RB_sample.sample2`. Generate an orthonormal Z from the sample set using the medium triangulation.

- Verify that, for $\mu_0 = 0.4, 0.6, 0.8, 1.2, 0.15$, i.e. $Bi = 0.15$, the value of the output is $T_{root\,N}(\mu 0) = 1.51561$.

- We next introduce a regular test sample, $\Xi_{test} \subset D$, of size $ntest = 100$ (in Python you can simply use `linspace(0.01, 1, 100)` to generate Ξ_{test}). Plot the convergence of the maximum relative error in the energy norm $\max_{\mu \in \Xi_{test}} |||u(\mu) - u_N(\mu)|||_\mu / |||u(\mu)|||_\mu$ and the maximum relative output error $\max_{\mu \in \Xi_{test}} |T_{root}(\mu) - T_{root\,N}(\mu)|/T_{root}(\mu)$ as a function of N (use the Python command `semilogy` for plotting).

- The Biot number is directly related to the cooling method; higher cooling rates (higher Bi) imply lower (better) T_{root} but also higher (worse) initial and operational costs. We can thus define (say) a total cost function as

$$C(Bi) = Bi + T_{root}(Bi), \tag{5}$$

  minimization of which yields an optimal solution. Apply your (online) reduced – basis approx – imation for $T_{root\,N}$ (that is, replace $T_{root}(Bi)$ in (5) with $T_{root\,N}(Bi)$) to find the optimal Bi. Any (simple) optimization procedure suffices for the minimization.

d) We consider now a two parameter ($P = 2$) problem where the conductivities are assumed to be equivalent, i.e., $k_1 = k_2 = k_3 = k_4$, but are allowed to vary between 0.1 and 10; and the Biot number, Bi, is allowed to vary from 0.01 to 1. The sample set S_N for $N_{max} = 46$ is given in the datafile `RB_sample.sample3`. Generate an orthonormal Z from the sample set using the coarse triangulation.

1. 1. We next introduce a regular grid, $\Xi_{test} \subset D$, of size $ntest = 400$ (a regular 20×20 grid). Plot the convergence of the maximum relative error in the energy norm $\max_{\mu\in\Xi_{test}} |||u(\mu) - u_N(\mu)|||_\mu / |||u(\mu)|||_\mu$ and the maximum relative output error $max_{\mu\in\Xi_{test}} |T_{root}(\mu) - T_{root\,N}(\mu)|/T_{root}(\mu)$ as a function of N.

e) We now consider the POD method and we wish to compare it with the Greedy approximation. To this end, we sample log randomly the parameter space ($P = 2$) and take $n_{\text{train}} = 100$ samples. Build the POD approximation using these samples as training set and compare the results with the Greedy approximation. Compute the RIC and the dimension of the POD space (N) such that the RIC is 99% of the total energy. Plot the POD and Greedy convergence of the maximum relative error in the energy norm $\max_{\mu\in\Xi_{test}} |||u(\mu) - u_N(\mu)|||_\mu / |||u(\mu)|||_\mu$ and the maximum relative output error $max_{\mu\in\Xi_{test}} |T_{root}(\mu) - T_{root\,N}(\mu)|/T_{root}(\mu)$ as a function of N.

2 Appendix 1 – Finite Element Method Implementation

We use Feel++ to implement the finite element matrices.