AME60714: Advanced Numerical Methods
Homework 4: Due Wednesday, November 11, 2020

Instructions: Complete three problems of your choice.

Problem 1: (30 points) Consider the parametrized nonlinear system of ODEs (perhaps the result of semi-discretization of a PDE)

$$\boldsymbol{M}\boldsymbol{q}_{,t} + \boldsymbol{f}(\boldsymbol{q};\boldsymbol{\mu}) = \boldsymbol{0}, \qquad \boldsymbol{q}(0;\boldsymbol{\mu}) = \overset{\circ}{\boldsymbol{q}}$$

where $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is a fixed mass matrix, $\boldsymbol{q}(t;\boldsymbol{\mu}) \in \mathbb{R}^N$ is the ODE state, $\boldsymbol{\mu} \in \mathbb{R}^M$ is the parameter vector, $\boldsymbol{f}(\boldsymbol{q};\boldsymbol{\mu})$ is the nonlinear velocity function, and $\overset{\circ}{\boldsymbol{q}}$ is the initial condition. Finally, assume $M \ll N$.

a) Approximate the solution

$$\boldsymbol{q}(t;\boldsymbol{\mu}) \approx \boldsymbol{q}_r(t;\boldsymbol{\mu}) := \bar{\boldsymbol{q}} + \boldsymbol{V}\boldsymbol{y}(t;\boldsymbol{\mu}),$$

where $\boldsymbol{V} \in \mathbb{R}^{N\times k}$ is the reduced basis ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$) and $\bar{\boldsymbol{q}}$ is an affine offset, $\boldsymbol{y}(t;\boldsymbol{\mu}) \in \mathbb{R}^k$ are the reduced coordinates, and $\boldsymbol{q}_r(t;\boldsymbol{\mu}) \in \mathbb{R}^N$ is the approximation to $\boldsymbol{q}(t;\boldsymbol{\mu})$ in the reduced subspace. Based on this solution approximation, a Galerkin reduced-order model takes the form of a system of k ODEs governing the reduced coordinates $\boldsymbol{y}(t;\boldsymbol{\mu})$

$$\boldsymbol{M}_r\boldsymbol{y}_{,t} + \boldsymbol{h}(\boldsymbol{y};\boldsymbol{\mu}) = \boldsymbol{0}, \qquad \boldsymbol{y}(0;\boldsymbol{\mu}) = \overset{\circ}{\boldsymbol{y}}.$$

Derive the reduced velocity function $\boldsymbol{h}(\boldsymbol{y};\boldsymbol{\mu})$, mass matrix $\boldsymbol{M}_r$, and initial condition $\overset{\circ}{\boldsymbol{y}}$. Discuss the advantages of taking $\bar{\boldsymbol{q}} = \overset{\circ}{\boldsymbol{q}}$.

b) Suppose we have collected snapshots $\{\boldsymbol{q}(t_i,\boldsymbol{\mu}_j)\}$ for $i = 1,\ldots,N_t$ and $j = 1,\ldots,N_{\boldsymbol{\mu}}$. How can these be used to construct a basis $\boldsymbol{V}$ via POD? (Hint: your answer should involve $\bar{\boldsymbol{q}}$)

c) What is the computational complexity of evaluating the nonlinear terms $\boldsymbol{h}(\boldsymbol{y};\boldsymbol{\mu})$ and $\dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{y}}(\boldsymbol{y};\boldsymbol{\mu})$? Assume $\boldsymbol{f}(\boldsymbol{q};\boldsymbol{\mu})$ and $\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}(\boldsymbol{q};\boldsymbol{\mu})$ are $\mathcal{O}(NM)$ complexity (usually the Jacobian is sparse).

d) What is the computational complexity of a single first-order explicit (forward Euler) and implicit (backward Euler) time step for the unreduced and reduced system? How do the complexity of the reduced and unreduced systems compare for $k \ll N$?

e) Repeat the (a)-(c) for the cases below where the velocity function $\boldsymbol{f}$ has special structure and discuss the implications.

- Linear in $\boldsymbol{q}$, nonlinear dependence on $\boldsymbol{\mu}$

$$\boldsymbol{f}(\boldsymbol{q};\boldsymbol{\mu}) = \boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{q},$$

where the computational complexity of evaluating $\boldsymbol{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$ is $\mathcal{O}(NM)$

- Linear in $\boldsymbol{q}$, affine dependence on $\boldsymbol{\mu}$

$$\boldsymbol{f}(\boldsymbol{q};\boldsymbol{\mu}) = \boldsymbol{A}(\boldsymbol{\mu})\boldsymbol{q}, \qquad \boldsymbol{A}(\boldsymbol{\mu}) = \sum_{i=1}^{s} \boldsymbol{A}_i g_i(\boldsymbol{\mu}),$$

where the computational complexity of evaluating $g_i(\boldsymbol{\mu}) \in \mathbb{R}$ is $\mathcal{O}(M)$ and $s \ll N$

- Quadratic in $\boldsymbol{q}$, affine dependence on $\boldsymbol{\mu}$

$$\boldsymbol{f}_i(\boldsymbol{q};\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{H}_{ijk}\boldsymbol{q}_j\boldsymbol{q}_k + \boldsymbol{A}_{ij}\boldsymbol{q}_j + \boldsymbol{b}_i g(\boldsymbol{\mu}),$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times N\times N}$, $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, $\boldsymbol{b} \in \mathbb{R}^N$, and the computational complexity of evaluating $g(\boldsymbol{\mu}) \in \mathbb{R}$ is $\mathcal{O}(M)$

Problem 2: (20 points) Consider the following parametrized nonlinear system of equations (perhaps the result of discretization of PDE)

$$\boldsymbol{r}(\boldsymbol{q};\boldsymbol{\mu}) = \boldsymbol{0},$$

where $\boldsymbol{q}(\boldsymbol{\mu}) \in \mathbb{R}^N$ is the state and $\boldsymbol{\mu} \in \mathbb{R}^M$ is a vector of parameters. Assume that the equation $\boldsymbol{r}(\cdot;\boldsymbol{\mu}) = \boldsymbol{0}$ has a unique solution for each $\boldsymbol{\mu} \in \mathbb{R}^M$. We will approximate the state as

$$\boldsymbol{q}(\boldsymbol{\mu}) \approx \boldsymbol{q}_r := \boldsymbol{V}\boldsymbol{y}(\boldsymbol{\mu}),$$

where $\boldsymbol{V} \in \mathbb{R}^{N\times k}$ is a reduced basis ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$, $k \ll N$), and we defined the reduced coordinates $\boldsymbol{y}(\boldsymbol{\mu})$ as

$$\boldsymbol{y}(\boldsymbol{\mu}) = \arg\min_{\boldsymbol{w}} \|\boldsymbol{r}(\boldsymbol{V}\boldsymbol{w};\boldsymbol{\mu})\|^2_{\boldsymbol{\Theta}}, \tag{1}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{N\times N}$ is a symmetric, positive definite matrix defining the norm, i.e., $\|\boldsymbol{x}\|_{\boldsymbol{\Theta}} = \sqrt{\boldsymbol{x}^T\boldsymbol{\Theta}\boldsymbol{x}}$.

a) Derive the nonlinear system of equations governing the reduced coordinates $\boldsymbol{y}$ by writing the first-order optimality system of (1).

b) Prove the minimum-residual reduced-order model is exact provided the exact solution lies in the reduced subspace. That is, suppose $\boldsymbol{q}(\boldsymbol{\mu}) \in \text{Ran}(\boldsymbol{V})$ ($\text{Ran}(\boldsymbol{V})$ is the span of the columns of $\boldsymbol{V}$) and prove that $\boldsymbol{V}\boldsymbol{y}(\boldsymbol{\mu}) = \boldsymbol{q}(\boldsymbol{\mu})$.

Problem 3: (40 points) In this problem, you will apply projection-based model reduction to accelerate the convection-diffusion problem from Homework 3 (no optimization). The evolution of the concentration of the contaminant $q(x,t)$ is modeled by the convection-diffusion equation

$$q_{,t} + \nabla\cdot(\beta q) - \nu\Delta q = 0 \quad \text{in} \quad \Omega\times(0,T],$$

where $\Omega = [-1,1]\times[-1,1]$ is the spatial domain, $\beta(x) = (\sin(\pi x_1)\cos(\pi x_2), -\cos(\pi x_1)\sin(\pi x_2))$ is the velocity field (Taylor-Green vortex), $\nu = 10^{-3}$ is the diffusion coefficient, and $T = 4$ is the final time. The boundary ($\partial\Omega$) is split into Neumann (Γ_N) and Dirichlet (Γ_D) (Figure 1): $\partial\Omega = \overline{\Gamma_N \cup \Gamma_D}$ with boundary conditions

$$\nabla q\cdot n = 0 \quad \text{on} \quad \Gamma_N, \qquad q = 0 \quad \text{on} \quad \Gamma_D,$$

where $n : \partial\Omega \to \mathbb{R}^2$ is the outward unit normal to $\partial\Omega$. The initial condition is

$$q(x,0) = \mathring{q}(x) = 5\exp\left(-\frac{x_1^2 + x_2^2}{0.1}\right).$$

Semi-discretization of the above PDE leads to a system of ODEs (setup for you in `FEdu`; see starter code)

$$\boldsymbol{M}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = 0, \quad \boldsymbol{q}(0) = \mathring{\boldsymbol{q}}, \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is the mass matrix, $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is the discretization of the convection and diffusion terms, $\boldsymbol{q}(t) \in \mathbb{R}^N$ is the state of the discretized PDE, and $\mathring{\boldsymbol{q}} \in \mathbb{R}^N$ is the discretization of the initial condition $\mathring{q}$. This system of ODEs is called the high-dimensional model (HDM).

a) Use backward Euler to integrate the HDM in time; use $N_t = 400$ time steps and $\Delta t = 10^{-2}$. Save the solution at each timestep (call these snapshots), compress using POD to form a basis $\boldsymbol{V} \in \mathbb{R}^{N\times k}$ ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$), and plot the decay of the singular values (use `semilogy` plot) and visualize the first three POD modes using the provided visualization routines.

b) Approximate the solution $\boldsymbol{q}(t) \approx \boldsymbol{q}_r(t) := \boldsymbol{V}\boldsymbol{y}(t)$, where $\boldsymbol{y}(t) \in \mathbb{R}^k$ are the reduced coordinates. Derive a system of ODEs that governs the reduced coordinates $\boldsymbol{y}$ using a Galerkin projection and the initial condition $\boldsymbol{y}(0) = \mathring{\boldsymbol{y}}$. Identify all terms that can be precomputed, i.e., computed once-and-for-all before timestepping initiated. Assuming of these terms are precomputed, what is the complexity of a single time step of the ROM compared to the HDM. Comment on the expected cost of the ROM relative to the HDM for $k \ll N$.

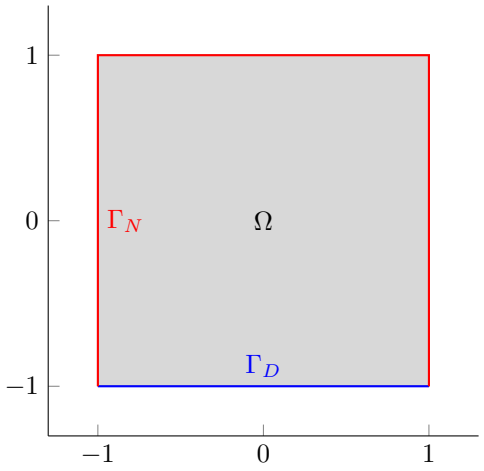

Figure 1: Convection-diffusion configuration

c) Run the ROM simulation for a reduced basis of dimension $k \in \{1, \ldots, 10, 15, 20, 25, 30\}$. For each k, record the CPU time required to complete the timestepping (not the precomputations) as well as the maximum error over all time steps

$$E = \max_{n \in \{1,\ldots,N_t\}} \|\boldsymbol{q}_n - \boldsymbol{V}\boldsymbol{y}_n\|,$$

where $\boldsymbol{q}_n \in \mathbb{R}^N$ is the HDM solution at timestep n and $\boldsymbol{y}_n \in \mathbb{R}^k$ are the ROM reduced coordinates at timestep n. Plot the error (E) as a function of the ROM size (k) and the ROM speedup relative to the HDM (CPU time of HDM divided by the CPU time of the ROM) as a function of the ROM size (k). Comment on the errors and speedups you observe; keep in mind that $N = 900$ (if you did not change the defaults in the starter code).

d) What is misleading about the results obtained thus far? What happens when you use the same reduced basis $\boldsymbol{V}$ but change the initial condition $\mathring{\boldsymbol{q}}$?

Problem 4: (40 points) Consider the following parametrized steady-state advection-diffusion equation

$$\nabla \cdot (\beta q) - \nu \Delta q = f \quad \text{in} \quad \Omega, \qquad q = 0 \quad \text{on} \quad \partial\Omega,$$

where $\Omega = \{x \in \mathbb{R}^2 \mid x_1^2 + x_2^2 \leqslant 2\}$ is the spatial domain, $\beta(\mu) = (\mu_2 \cos\mu_1, \mu_2 \sin\mu_1)$ is the constant (parametrized) velocity field, $\nu = 1$ is the diffusion coefficient, $f = 10$ is the source term, and $\mu \in \mathcal{D} := [0, \pi] \times [0, 10]$ are the parameters. Discretization of the above PDE using the finite element method leads to a (affine) parametrized system of linear equations (setup for you in `FEdu`; see starter code)

$$\boldsymbol{K}(\mu)\boldsymbol{q} = \boldsymbol{f}, \qquad \boldsymbol{K}(\mu) := \boldsymbol{K}_0 + g_1(\mu)\boldsymbol{K}_1 + g_2(\mu)\boldsymbol{K}_2, \tag{3}$$

where $\boldsymbol{q}(\mu) \in \mathbb{R}^N$ is the solution of the discretized PDE, $\boldsymbol{K}_0, \boldsymbol{K}_1, \boldsymbol{K}_2 \in \mathbb{R}^{N \times N}$ are sparse stiffness-like matrices, $\boldsymbol{f} \in \mathbb{R}^N$ is the discretization of the source term f, $g_1(\mu) = \mu_2 \cos\mu_1$, and $g_2(\mu) = \mu_2 \sin\mu_1$. This system of equation is the HDM.

a) Suppose you are given a reduced basis $\boldsymbol{V} \in \mathbb{R}^{N \times k}$ ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$) and approximate the solution as $\boldsymbol{q}(\mu) \approx \boldsymbol{q}_r(\mu) := \boldsymbol{V}\boldsymbol{y}(\mu)$, where $\boldsymbol{y}(\mu) \in \mathbb{R}^k$ are the reduced coordinates. Derive the Galerkin ROM corresponding to the HDM. Identify all terms that can be precomputed, i.e., independent of μ. Assuming of these terms are precomputed, what is the complexity of solving the ROM compared to the HDM. Comment on the expected cost of the ROM relative to the HDM for $k \ll N$.

b) Define a finite collection of the parameter space $\mathcal{D}_r^{\text{unif}} \subset \mathcal{D}$ by uniformly sampling $\mathcal{D}$ in each dimension with r samples ($|\mathcal{D}_r| = r^2$). Construct a reduced basis by solving the HDM at each $\mu \in \mathcal{D}_r^{\text{unif}}$ (be sure

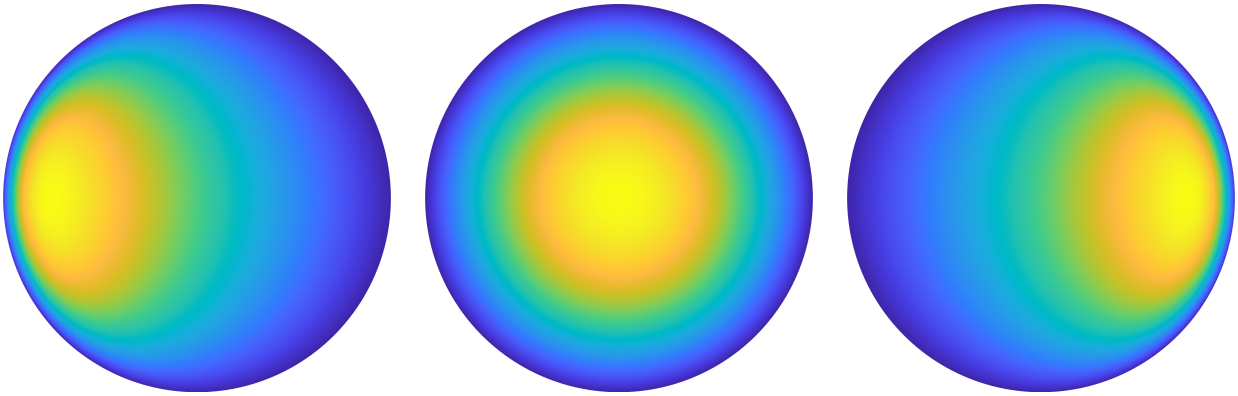

Figure 2: Solution at 3 parameter configurations: $\mu = (0, 10)$ (*left*), $\mu = (0, 0)$ (*center*), $\mu = (\pi/2, 10)$ (*right*).

to orthogonalize the snapshots; do not compress them). Solve the corresponding ROM for each point in $\mathcal{D}_{\text{test}} := \mathcal{D}_{10}^{\text{unif}}$. For $r = 1, \ldots, 6$, record the CPU time required to solve all parameters in the test set $\mathcal{D}_{\text{test}}$ and the maximum error

$$E = \max_{\mu \in \mathcal{D}_{\text{test}}} \|\boldsymbol{q}(\mu) - \boldsymbol{V}\boldsymbol{y}(\mu)\| .$$

Plot the error (E) as a function of the ROM size (r^2). Comment on the errors and speedups you observe; keep in mind that $N = 1161$ (if you did not change the defaults in the starter code).

c) Repeat (b) with greedy training. Define the collection of candidate points $\mathcal{D}_{\text{cand}} := \mathcal{D}_{8}^{\text{unif}}$ and use the same test set from (b): $\mathcal{D}_{\text{test}}$. Initialize the training set with $\mu^{(1)} = (0, 0)$. At iteration k of the greedy algorithm, the kth training parameter will be determined as the parameter in $\mathcal{D}_{\text{cand}}$ where the ROM (build from the reduced basis at iteration k) maximizes the HDM residual

$$\mu^{(k)} = \arg\max_{\mu \in \mathcal{D}_{\text{cand}}} \|\boldsymbol{K}(\mu)\boldsymbol{V}_{k-1}\boldsymbol{y}_{k-1}(\mu) - \boldsymbol{f}\|,$$

where $\boldsymbol{V}_{k-1} \in \mathbb{R}^{N \times (k-1)}$ is the reduced basis at iteration k and $\boldsymbol{y}_{k-1}(\mu)$ are the corresponding reduced coordinates. Once the new parameter is selected $\mu^{(k)}$, the HDM is sampled $\boldsymbol{q}(\mu^{(k)})$ and the basis updated to $\boldsymbol{V}_k$. The iteration usually terminates when a maximum number of samples are collected or the HDM residual at all candidate points is below a given tolerance. For this problem, terminate the greedy method when r^2 samples have been collected. For a given r, once the basis is constructed, solve the corresponding ROM for each point in $\mathcal{D}_{\text{test}}$. For $r = 1, \ldots, 6$, record the CPU time required to solve all parameters in the test set $\mathcal{D}_{\text{test}}$ and the maximum error (E). Plot the error (E) as a function of the ROM size (r^2). Comment on this sampling strategy relative to the uniform sampling strategy in (c).