A Gentle Introduction to Model Order Reduction via Proper Orthogonal Decomposition

Shane McQuarrie

25 October 2019

ODEN INSTITUTE
FOR COMPUTATIONAL ENGINEERING & SCIENCES

The University of Texas at Austin

Outline

Large Dynamical Systems

Many computational science, engineering, and mathematics applications eventually reduce to the problem of solving a (very) large system of ordinary differential equations:

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(t, \mathbf{x}(t)), \qquad\qquad \mathbf{x}(0) = \mathbf{x}_0, \qquad (*)$$

eq:FOM

where

$$\mathbf{x} : \mathbb{R} \to \mathbb{R}^n, \qquad\qquad \mathbf{f} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n, \qquad\qquad \mathbf{x}_0 \in \mathbb{R}^n.$$

We call $(*)$ the *full-order model* (FOM).

Some assumptions:

- We know $\mathbf{f}$ and that it is "nice"
- The dimension n is large ($n \sim 10^8$)
- We can solve $(*)$, but it is computationally expensive to do so

Small Dynamical Systems

The goal of model order reduction (MOR) is to construct a low-dimensional system

$$\frac{d}{dt}\hat{\mathbf{x}}(t) = \hat{\mathbf{f}}(t, \hat{\mathbf{x}}(t)), \qquad \hat{\mathbf{x}}(0) = \hat{\mathbf{x}}_0, \qquad (**)$$

eq:ROM

where

$$\hat{\mathbf{x}} : \mathbb{R} \to \mathbb{R}^r, \qquad \hat{\mathbf{f}} : \mathbb{R} \times \mathbb{R}^r \to \mathbb{R}^r, \qquad \hat{\mathbf{x}}_0 \in \mathbb{R}^r,$$

with $r \ll n$ and $\hat{\mathbf{x}}$ related to $\mathbf{x}$ somehow.

We call $(**)$ the *reduced-order model* (ROM).

The goal is typically $r < 50$ or so.

Strategy: Projection to a Linear Subspace

Idea: approximate $\mathbf{x} \in \mathbb{R}^n$ as a linear combination of r vectors.

That is, find $V \in \mathbb{R}^{n \times r}$ such that

$$\mathbf{x}(t) \approx V\hat{\mathbf{x}}(t),$$

meaning $\text{Range}(V) \approx \{\mathbf{x}(t) \mid t \geq 0\}$.

But how tho?

Strategy: Collect Snapshot Data

To approximate the solution set $\{\mathbf{x}(t) \mid t \geq 0\}$, simulate the FOM $(*)$ and record the solution $\mathbf{x}(t)$ at times $0 = t_0 < t_1 < \cdots < t_{k-1}$. Stack the results column-wise to get the *snapshot matrix*,

$$X = \left[\begin{array}{c|c|c|c} \mathbf{x}(t_0) & \mathbf{x}(t_1) & \cdots & \mathbf{x}(t_{k-1}) \end{array}\right] \in \mathbb{R}^{n \times k}.$$

Now the column space (range) of X is an approximation to $\{\mathbf{x}(t) \mid t \geq 0\}$.

But we want $V \in \mathbb{R}^{n \times r}$ (probably $r < k < n$)...

Strategy:

Use the SVD!

Strategy: Use the SVD!

Theorem

Let $A \in \mathbb{R}^{n\times k}$ *have the singular value decomposition*

$$A = \Phi\Sigma\Psi^{\mathsf{T}}.$$

Then the first r *columns of* Φ *are the best rank-*r *approximation of the column space of* A*. In other words,* $\{\phi_j\}_{j=1}^{r}$ *solves the minimization problem*

$$\underset{\tilde{\phi}_1,\ldots,\tilde{\phi}_r\in\mathbb{R}^n}{argmin} \sum_{j=1}^{k} \left\| \mathbf{x}_j - \sum_{i=1}^{r} \left\langle \mathbf{x}_j, \tilde{\boldsymbol{\phi}}_i \right\rangle_{\mathbb{R}^n} \tilde{\boldsymbol{\phi}}_i \right\|_{\mathbb{R}^n}^2$$

$$such\ that$$

$$\left\langle \tilde{\boldsymbol{\phi}}_i, \tilde{\boldsymbol{\phi}}_j \right\rangle_{\mathbb{R}^n} = \delta_{ij} \quad for \quad 1 \le i, j \le r.$$

Strategy: Use POD to Define the Linear Subspace

Define

$$V_r := \Phi_{:,:r}(X) \in \mathbb{R}^{n \times r},$$

the *POD basis of rank* r corresponding to the snapshot matrix X.

Since Φ has orthonormal columns, so does V_r. In particular,

- $V_r^\mathsf{T} V_r = I \in \mathbb{R}^{r \times r}$
- $V_r V_r^\mathsf{T}$ is the orthogonal projector onto

$$\text{Range}(V_r) \approx \text{Range}(X) \approx \{\mathbf{x}(t) \mid t \geq 0\}$$

Strategy: Substitute the Approximation

Since V_r is orthonormal,

$$\mathbf{x}(t) \approx V_r\hat{\mathbf{x}}(t) \qquad \Longrightarrow \qquad V_r^{\mathsf{T}}\mathbf{x}(t) \approx \cancel{V_r^{\mathsf{T}}V_r}\hat{\mathbf{x}}(t) = \hat{\mathbf{x}}(t).$$

Substituting this into the FOM $(*)$,

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(t, \mathbf{x}(t)), \qquad \mathbf{x}(0) = \mathbf{x}_0,$$

we obtain

$$\frac{d}{dt}V_r\hat{\mathbf{x}}(t) = \mathbf{f}(t, V_r\hat{\mathbf{x}}(t)), \qquad V_r\hat{\mathbf{x}}(0) = \mathbf{x}_0,$$

$$\cancel{V_r^{\mathsf{T}}V_r}\frac{d}{dt}\hat{\mathbf{x}}(t) = V_r^{\mathsf{T}}\mathbf{f}(t, V_r\hat{\mathbf{x}}(t)), \qquad \cancel{V_r^{\mathsf{T}}V_r}\hat{\mathbf{x}}(0) = V_r^{\mathsf{T}}\mathbf{x}_0,$$

$$\boxed{\frac{d}{dt}\hat{\mathbf{x}}(t) = V_r^{\mathsf{T}}\mathbf{f}(t, V_r\hat{\mathbf{x}}(t))}, \qquad \boxed{\hat{\mathbf{x}}(0) = V_r^{\mathsf{T}}\mathbf{x}_0 =: \hat{\mathbf{x}}_0}.$$

Strategy: Precompute When Possible

Suppose the FOM operator $\mathbf{f}$ is of the form

$$\mathbf{f}(t, \mathbf{x}(t)) = A\mathbf{x}(t) + \mathbf{c},$$

where

$$A \in \mathbb{R}^{n \times n}, \qquad \mathbf{c} \in \mathbb{R}^{n}.$$

Then the corresponding ROM is

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{x}}(t) = V_r^{\mathsf{T}}\mathbf{f}(t, V_r\hat{\mathbf{x}}(t)) &= V_r^{\mathsf{T}} A V_r \hat{\mathbf{x}}(t) + V_r^{\mathsf{T}}\mathbf{c} \\ &= \hat{A}\hat{\mathbf{x}}(t) + \hat{\mathbf{c}}, \end{aligned}$$

where

$$\hat{A} := V_r^{\mathsf{T}} A V_r \in \mathbb{R}^{r \times r}, \qquad \hat{\mathbf{c}} := V_r^{\mathsf{T}}\mathbf{c} \in \mathbb{R}^{r}.$$

Summary

Offline Phase (construct the ROM)

1. Simulate the n-dimensional FOM to collect snapshots, stacked column-wise into the matrix X.
2. Compute the rank-r POD basis matrix V_r corresponding to X (the first r left singular vectors).
3. Construct $\hat{A}$, $\hat{\mathbf{c}}$ from A, $\mathbf{c}$, and V_r. This defines an r-dimensional ROM.

Online Phase (simulate the ROM)

Given an initial value $\mathbf{x}_0$,

1. Project the initial value as $\hat{\mathbf{x}}_0 = V_r^{\mathsf{T}}\mathbf{x}_0$.
2. Solve the ROM with initial value $\hat{\mathbf{x}}_0$, obtaining $\hat{\mathbf{x}}(t)$
3. Reconstruct the solution in the initial space as $\mathbf{x}_{\texttt{ROM}}(t) = V_r\hat{\mathbf{x}}(t)$.

Example: Heat Equation

Consider the following one-dimensional parabolic PDE.

$$
\begin{aligned}
x_t(t,\omega) &= \frac{1}{\pi^2} x_{\omega\omega}(t,\omega), && (t,\omega) \in (0,T] \times \Omega, \\
x_\omega(t,0) &= x_\omega(t,1) = 0, && t > 0, \\
x(0,\omega) &= \cos(\pi\omega) + \varepsilon \cos(4\pi\omega) + 2, && \omega \in \Omega.
\end{aligned}
$$

Discretizing with a simple finite difference scheme yields a linear ODE,

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x}(t).$$

We can reduce this system!

`https://tinyurl.com/y28qhsfl`