



Otto-von-Guericke Universität Magdeburg
Faculty of Mathematics
Summer term 2015

Model Reduction for Dynamical Systems

–Lecture 11–

Peter Benner Lihong Feng

Max Planck Institute for Dynamics of Complex Technical Systems
Computational Methods in Systems and Control Theory
Magdeburg, Germany

benner@mpi-magdeburg.mpg.de feng@mpi-magdeburg.mpg.de
www.mpi-magdeburg.mpg.de/2909616/mor_ss15



Contents



- 1 POD method for steady systems
- 2 POD method for dynamical systems
- 3 RB method for steady systems
- 4 RB method for dynamical systems
- 5 EIM

POD for linear parametric steady systems



Linear Steady systems

$$\begin{aligned} E(\mu)x &= B(\mu), \\ y &= C(\mu)x, \end{aligned} \tag{1}$$

$$\mu = (\mu_1, \dots, \mu_p), \quad x = x(\mu).$$



POD for linear parametric steady systems

POD based PMOR for linear parametric steady systems:

- 1 Specify a group of samples of μ : $\mu^i = (\mu_1^i, \dots, \mu_p^i)$, $i = 1, \dots, m$.
- 2 Solve (1) at each sample μ^i , and get the solution $x(\mu^i)$.
- 3 Form the snapshot matrix $X = (x(\mu^1), \dots, x(\mu^m))$.

- 4 SVD: $X = U \begin{pmatrix} \Sigma & \\ & 0 \end{pmatrix} Q^T$. $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$,
 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d$.

- 5 Specify the order r of the ROM by checking, e.g., $\frac{\sum_{j=r+1}^d \sigma_j}{\sum_{j=1}^d \sigma_j} < \text{tol}$, tol

is a small number, e.g., 10^{-7} .

- 6 Take the first r columns from U : $V = U(:, 1:r)$.

Form the ROM:

$$V^T E(\mu) V z = V^T B(\mu), \quad y = C(\mu) V z.$$



POD for nonlinear parametric steady systems

Nonlinear Steady systems

$$\begin{aligned} E(\mu)x + f(\mu, x) &= B(\mu), \\ y &= C(\mu)x, \end{aligned} \quad (2)$$

$f(\mu, x)$ is a nonlinear function of the state $x = x(\mu)$.

POD based PMOR for nonlinear parametric steady systems:

- Specify a group of samples: $\mu^i = (\mu_1^i, \dots, \mu_p^i)$, $i = 1, \dots, m$.
- Form the snapshot matrix: $X = (x(\mu^1), \dots, x(\mu^m))$, and $F = (f(\mu^1, x(\mu^1)), \dots, f(\mu^m, x(\mu^m)))$.
- SVD: $X = U \begin{pmatrix} \Sigma & \\ & 0 \end{pmatrix} Q^T$. $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$,
 $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d$.
- Specify the order r of the ROM by, e.g., $\frac{\sum_{j=r+1}^d \sigma_j}{\sum_{j=1}^d \sigma_j} < \text{tol}$.
- Form the projection matrix V : $V = U(:, 1:r)$.

POD for nonlinear parametric steady systems



- SVD: $F = U^F \begin{pmatrix} \Sigma^F & \\ & 0 \end{pmatrix} (Q^F)^T$. $\Sigma^F = \text{diag}(\sigma_1^F, \dots, \sigma_d^F)$,
 $\sigma_1^F \geq \sigma_2^F \geq \dots \geq \sigma_d^F$.
- Specify dimension r of the interpolation basis by, e.g.,

$$\frac{\sum_{j=r+1}^d \sigma_j^F}{\sum_{j=1}^d \sigma_j^F} < \text{tol.}$$

- Form the DEIM interpolation basis: $U_r^F = U^F(:, 1:r)$.
- Use U_r^F as the input for the DEIM algorithm and generate the indicis \wp_1, \dots, \wp_l , and the index matrix $P = (e_{\wp_1}, \dots, e_{\wp_l})$.
- Form the ROM:

$$\begin{aligned} V^T E(\mu) Vz &+ V^T U_r^F (P^T U_r^F)^{-1} P^T f(\mu, Vz) \\ &= V^T B(\mu), \\ y &= C(\mu) Vz. \end{aligned}$$



POD for linear parametric dynamical systems

Linear dynamical systems

$$\begin{aligned} E(\mu)dx/dt &= A(\mu)x + B(\mu)u(t), \\ y &= C(\mu)x. \end{aligned} \quad (3)$$

POD based PMOR for linear parametric dynamical systems:

- Specify samples of μ and $u(t)$: $w^i = (\mu^i, u^i(t))$, $i = 1, \dots, m$,
 $\mu^i = (\mu_1^i, \dots, \mu_p^i)$.
- Solve (3) at each sample w^i , and get the solution
 $X(w^i) := [x_{t_1}(w^i), \dots, x_{t_N}(w^i)]$.
- SVD of $X(w^i)$: $X(w^i) = U^i \begin{pmatrix} \Sigma^i & \\ & 0 \end{pmatrix} (Q^i)^T$.
 $\Sigma^i = \text{diag}(\sigma_1^i, \dots, \sigma_d^i)$, $i = 1, \dots, m$, $\sigma_1^i \geq \sigma_2^i \geq \dots \geq \sigma_d^i$.
- Determine the order r_i for each i by e.g.

$$\left(\sum_{j=r_i+1}^d \sigma_j^i \right) / \left(\sum_{j=1}^d \sigma_j^i \right) < \text{tol.}$$

- Let: $V_i = U^i(:, 1 : r_i)$, $i = 1, \dots, m$.

POD for linear parametric dynamical systems



- Form $X = (V_1, \dots, V_m) \in \mathbb{R}^{n \times \tilde{N}}$, $\tilde{N} = \sum_{i=1}^m r_i$.
- SVD: $X = U \begin{pmatrix} \Sigma & \\ & 0 \end{pmatrix} Q^T$, $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$.
- Specify the order r of the ROM by e.g. $\frac{\sum_{j=r+1}^d \sigma_j}{\sum_{j=1}^d \sigma_j} < \text{tol}$.
- Form the projection matrix: $V = U(:, 1:r)$.
- Form the ROM:

$$\begin{aligned} V^T E(\mu) V dz/dt &= V^T A(\mu) V z + V^T B(\mu) u(t), \\ \hat{y} &= C(\mu) V z. \end{aligned}$$

POD for nonlinear parametric dynamical systems



Nonlinear parametric dynamical systems

$$\begin{aligned} E(\mu)dx/dt &= A(\mu) + f(\mu, x) + B(\mu)u(t), \\ y &= C(\mu)x. \end{aligned} \quad (4)$$

POD based PMOR for nonlinear parametric dynamical systems:

- 1. Specify samples of μ and $u(t)$: $w^i = (\mu^i, u^i(t))$, $i = 1, \dots, m$,
 $\mu^i = (\mu_1^i, \dots, \mu_p^i)$.
- 2. Solve (4) at each sample w^i , and get the solution
 $X(w^i) := [x_{t_1}(w^i), \dots, x_{t_N}(w^i)]$.
- 3. Form the snapshots of f :

$$F = [f(X(w^1)), \dots, f(X(w^m))] \in \mathbb{R}^{n \times N \cdot m},$$

where $f(X(w^i)) := [f(x_{t_1}(w^i)), \dots, f(x_{t_N}(w^i))]$.

- 4. SVD of $X(w^i)$: $X(w^i) = U^i \begin{pmatrix} \Sigma^i & \\ & 0 \end{pmatrix} (Q^i)^T$.
 $\Sigma^i = \text{diag}(\sigma_1^i, \dots, \sigma_d^i)$, $i = 1, \dots, m$.

POD for nonlinear parametric dynamical systems



- 5. Determine the order r_i for each i by, e.g., $\frac{\sum_{j=r_i+1}^d \sigma_j^s}{\sum_{j=1}^d \sigma_j^i} < tol$, let:

$$V_i = U^i(:, 1 : r_i), i = 1, \dots, m.$$
- 6. Form $X = (V_1, \dots, V_m) \in \mathbb{R}^{n \times \tilde{N}}$, $\tilde{N} = \sum_{i=1}^m r_i$.
- 7. SVD: $X = U \begin{pmatrix} \Sigma & \\ & 0 \end{pmatrix} Q^T$. $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_d)$.
- 8. Specify the order r of the ROM by e.g. $\frac{\sum_{j=r+1}^d \sigma_j}{\sum_{j=1}^d \sigma_j} < tol$.
- 9. Form the projection matrix: $V = U(:, 1 : r)$.

POD for nonlinear parametric dynamical systems



- 10. SVD: $F = U^F \begin{pmatrix} \Sigma^F & \\ & 0 \end{pmatrix} (Q^F)^T$. $\Sigma^F = \text{diag}(\sigma_1^F, \dots, \sigma_d^F)$.
- 11. Specify the dimension r of the basis for interpolation by, e.g.,

$$\frac{\sum_{j=r+1}^d \sigma_j^F}{\sum_{j=1}^d \sigma_j^F} < \text{tol.}$$

- 12. Form the DEIM interpolation basis: $U_f = U^F(:, 1:r)$.
- 13. Use U_f as the input for DEIM algorithm and generate the indices \wp_1, \dots, \wp_l , and the index matrix $P = (e_{\wp_1}, \dots, e_{\wp_l})$.
- 14. Form the ROM:

$$\begin{aligned} V^T E(\mu) V dz/dt &= V^T A(\mu) Vz \\ &\quad + V^T U_f (P^T U_f)^{-1} P^T f(\mu, Vz) \\ &\quad + V^T B(\mu) u(t), \\ \hat{y} &= C(\mu) Vz. \end{aligned}$$

Issues under active research

- Adaptive (optimal) sampling of the parameters μ_1, \dots, μ_l , and the inputs $u(t)$.
- When and how to update the DEIM interpolation?
- Error estimation/bound.

Reduced Basis (RB) method for steady systems



RB method for linear parametric steady systems

- 1 Specify a training set of samples: $\mathcal{P}_{train} := \{\mu^1, \dots, \mu^m\}$.
- 2 Specify an error estimator/indicator/bound $\eta(\mu)$.
- 3 Use Algorithm 1 to compute the projection matrix V .
- 4 Form the ROM:

$$\begin{aligned} V^T E(\mu) Vz &= V^T B(\mu), \\ y &= C(\mu) Vz. \end{aligned}$$

RB method for steady systems



Compute V for steady systems

Algorithm 1 Greedy Algorithm for steady systems

Input: \mathcal{P}_{train} , $tol_{RB} (< 1)$

Output: Projection matrix V representing the reduced basis.

- 1: Initialization: $V = []$, $\mu^* = \mu^1$, $\eta(\mu^*) = 1$
 - 2: **while** the error $\eta(\mu^*) > tol_{RB}$ **do**
 - 3: Solve (1) or (2) to get the solution $x(\mu^*)$.
 - 4: Update: $V := \text{orth} [V, x(\mu^*)]$.
 - 5: Find $\mu^* := \arg \max_{\mu \in \mathcal{P}_{train}} \eta(\mu)$.
 - 6: **end while**
-



RB method for steady systems

RB method for nonlinear parametric steady systems

- 1 Specify $\mathcal{P}_{train} := \{\mu^1, \dots, \mu^m\}$.
- 2 Specify an error estimator/indicator/bound $\eta(\mu)$.
- 3 Use Algorithm 1 to compute the projection matrix V .
- 4 Compute the EIM basis U_f using the empirical interpolation method (EIM).
- 5 Form the ROM:

$$\begin{aligned} V^T E(\mu) Vz &+ V^T U_f \beta(\mu) \\ &= V^T B(\mu), \\ y &= C(\mu) Vz. \end{aligned}$$

Key points:

- EIM.
- $\beta(\mu)$ must be updated for each sample of μ when solving the ROM!

RB method for dynamical systems



RB method for linear parametric dynamical systems

- 1 Specify $\mathcal{P}_{train} := \{w^1, \dots, w^m\}$, $w^i = (\mu^i, u^i(t))$.
- 2 Specify an error estimator/indicator/bound $\eta(w)$.
- 3 Use Algorithm 2 to compute the projection matrix V .
- 4 Form the ROM:

$$\begin{aligned} V^T E(\mu) V dz/dt &= V^T A(\mu) Vz + V^T B(\mu) u(t), \\ y &= C(\mu) Vz. \end{aligned}$$



RB method for dynamical systems

Compute V for dynamical systems

Algorithm 2 POD-Greedy algorithm for dynamical systems

Input: $\mathcal{P}_{train}, tol_{RB} (< 1)$

Output: Projection matrix: $V = [V_1, \dots, V_m]$

- 1: Initialization: $s = 0, V = [], w^* = (\mu^1, u^1(t)), \eta_s(w^*) = 1.$
- 2: **while** the error $\eta(w^*) > tol_{RB}$ **do**
- 3: Compute the trajectory $X = [x(w^*, t_1), \dots, x(w^*, t_N)].$
- 4: $\mathcal{W} = \text{colspan}\{V\}.$
- 5: IF $V \neq []$ **do**
 For each t_k , compute $x(w^*, t_k) := x(w^*, t_k) - \text{Proj}_{\mathcal{W}}[x(w^*, t_k)].$
 ENDIF
- 6: SVD: $X = U \Sigma V^T, V_{s+1} := U(:, 1).$
- 7: Update: $V = [V, V_{s+1}],$
- 8: $s = s + 1.$
- 9: Find $w^* := \arg \max_{w \in \mathcal{P}_{train}} \eta(w).$
- 10: **end while**

RB method for dynamical systems



RB method for nonlinear parametric dynamical systems

- 1 Specify $\mathcal{P}_{train} := \{w^1, \dots, w^m\}$, $w^i = (\mu^i, u^i(t))$.
- 2 Specify an error estimator/indicator/bound $\eta(w)$.
- 3 Use Algorithm 2 to compute the projection matrix V .
- 4 Compute the EIM basis U_f using the empirical interpolation method (EIM).
- 5 Form the ROM:

$$\begin{aligned} V^T E(\mu) V dz/dt &= V^T A(\mu) Vz + V^T U_f \beta(w, t) + V^T B(\mu) u(t,) \\ y &= C(\mu) Vz. \end{aligned}$$

Key points:

- EIM.
- $\beta(w, t)$ must be updated for **each value of μ , and each $u(t)$, at each time step t_i** , when solving the ROM!
- Efficient error estimation.

Empirical Interpolation Method (EIM)



Algorithm 3: EIM for steady systems

Input: Snapshots of $f(\mu, x(\mu))$:

$$F = \{f(\mu^1, x(\mu^1)), \dots, f(\mu^m, x(\mu^m))\},$$

$f^k := f(\mu^k, x(\mu^k))$, $P = []$, $U = []$.

Output: EI basis: $U = [u_1, \dots, u_M]$ and

Indices: $\{\varphi_1, \dots, \varphi_M\}$, index matrix: $P = [e_{\varphi_1}, \dots, e_{\varphi_M}]$.

1. $s = 1$, $\xi_1 := \arg \max_{f^k \in F} \|f^k\|_2$, $\varphi_1 := \arg \max_{j \in \{1, \dots, n\}} |\xi_{1,j}|$, $u_1 := \xi_1 / \xi_{1,\varphi_1}$.

$P = [P, e_{\varphi_1}]$, $U = [U, u_1]$.

Empirical Interpolation Method (EIM)



(Continued) Algorithm 3: EIM for steady systems

while $\|\xi_s\|_2 > \epsilon$ **do**

2: $s := s + 1$. For each f^k , $k = 1, \dots, m$: $f^k \approx \sum_{i=1}^{s-1} \beta_{ki} u_i$, i.e.

$$f^k \approx U \beta_k, \beta_k = (\beta_{k1}, \dots, \beta_{ks-1})^T$$

β_k is determined by interpolation:

$$P^T f^k = P^T U \beta_k,$$

so that $\beta_k = (P^T U)^{-1} P^T f^k$.

Let $f^{k^*} := \arg \max_{f^k \in F} \|f^k - U \beta_k\|_2$, $\xi_s = f^{k^*} - U \beta_{k^*}$

4: **if** $\|\xi_s\|_2 \leq \epsilon$ **then**

Stop and set $M = s - 1$.

6: **else**

Define $\varphi_s := \arg \max_{j \in \{1, \dots, n\}} |\xi_{s,j}|$, $u_s := \xi_s / \xi_{s, \varphi_s}$.

8: $P = [P, e_{\varphi_s}]$, $U = [U, u_s]$.

end if

10: **end while**

Application of EI to ROM simulation



Simulating the ROM:

$$\begin{aligned} V^T E(\mu) Vz + V^T U_f \beta(\mu) &= V^T B(\mu), \\ y &= C(\mu) Vz. \end{aligned}$$

- $U_f = U$, U is from Algorithm 3 (EIM). $f(\mu, Vz(\mu)) \approx U\beta(\mu)$, $\forall \mu$.
- Using the index matrix P from Algorithm 1,

$$P^T f(\mu, Vz(\mu)) = P^T U\beta(\mu), \forall \mu. \text{ (Interpolation)}$$

$$\beta(\mu) = (P^T U)^{-1} P^T f(\mu, Vz(\mu)), \forall \mu.$$

For each value of $\mu = (p_1, \dots, p_l)$, using Newton's method to solve

$$\hat{g}(\mu, Vz(\mu)) = \hat{B}(\mu),$$

where $\hat{g}(\mu, Vz(\mu)) = V^T E(\mu) Vz + \underbrace{V^T U (P^T U)^{-1} P^T f(\mu, Vz(\mu))}_{\text{precomputed}}$,

$$\hat{B}(\mu) = V^T B(\mu).$$

Empirical Interpolation Method (EIM)



Algorithm 4: EIM for dynamical systems

Input: Snapshots of $f(w, x(t, w))$ (note that f depends not only on μ , but also on the input $u(t)$):

$$F = [f(w^1, x(t_1, w^1)), \dots, f(w^1, x(t_N, w^1)) \dots, f(w^m, x(t_N, w^m))],$$

$$w^i = (\mu^i, u^i(t)), \quad f^k := F(:, k), \quad P = [], \quad U = [].$$

Output: EI basis: $U = [u_1, \dots, u_M]$ and

Indices: $\{\varphi_1, \dots, \varphi_M\}$, index matrix P : $P = [e_{\varphi_1}, \dots, e_{\varphi_M}]$.

$$1. \quad s = 1, \quad \xi_1 := \arg \max_{f^k, 1 \leq k \leq N \cdot m} \|f^k\|_2,$$

$$\varphi_1 := \arg \max_{j \in \{1, \dots, n\}} |\xi_{1,j}|, \quad u_1 := \xi_1 / \xi_{1, \varphi_1}. \quad P = [P, e_{\varphi_1}], \quad U = [U, u_1].$$

Empirical Interpolation Method (EIM)



(Continued) Algorithm 4: EIM for dynamical systems

while $\|\xi_m\|_2 > \epsilon$ **do**

2: $s := s + 1$. For each $f^k, k = 1, \dots, N \cdot m : f^k \approx \sum_{i=1}^{s-1} c_{ki} u_i$, i.e.

$$f^k \approx U c_k, c_k = (c_{k1}, \dots, c_{ks-1})^T$$

c_k is determined by interpolation:

$$P^T f^k = P^T U c_k,$$

so that $c_k = (P^T U)^{-1} P^T f^k$.

Let $f^{k^*} := \arg \max_{f^k, 1 \leq k \leq N \cdot K} \|f^k - U c_k\|$. $\xi_s = f^{k^*} - U c_{k^*}$

4: **if** $\|\xi_s\| \leq \epsilon$ **then**

Stop and set $M = s - 1$.

6: **else**

Define $\varphi_s := \arg \max_{j \in \{1, \dots, n\}} |\xi_{s,j}|$, $u_s := \xi_s / \xi_{s, \varphi_s}$.

8: **end if**

end while

Application of EI to ROM simulation



Simulating the ROM:

$$\begin{aligned} V^T E(\mu) Vz/dt &= V^T A(\mu) Vz + V^T U_f \beta(w, t) \\ &+ V^T B(\mu), \\ y &= C(\mu) Vz. \end{aligned}$$

- $U_f = U$, U is from Algorithm 2. $f(\mu, Vz(\mu, t)) \approx U\beta(w, t)$, $\forall \mu$.
- Using the index matrix P from Algorithm 2,

$$P^T f(\mu, Vz(\mu, t)) = P^T U\beta(w, t), \forall \mu, \forall t. \text{(Interpolation)}$$

$$\beta(w, t) = (P^T U)^{-1} P^T f(\mu, Vz(\mu, t)), \forall \mu.$$

For each value of μ , each value of input $u(t)$, and at each time step t_i , compute:

$$\underbrace{V^T U (P^T U)^{-1} P^T}_{\text{precomputed}} f(\mu, Vz(\mu, t_i)),$$







to get $V^T U_f \beta(w, t_i)$ in the ROM.

Issues under active research

- Optimal/adaptive defining/updating the training set: \mathcal{P}_{train} .
- Adaptive adjustment of the EIM basis and the reduced basis.
- More efficient error estimation/bound.

Reference



-  M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations*, C. R. Acad. Sci. Paris, Ser. I, 339 (2004), pp. 667–672.
-  M. DROHMANN, B. HAASDONK, AND M. OHLBERGER, *Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation*, SIAM J. Sci. Comput., 34 (2012), pp. 937–969.
-  B. HAASDONK AND M. OHLBERGER, *Reduced basis method for finite volume approximations of parametrized linear evolution equations*, ESAIM: M2AN, 42 (2008), pp. 277–302.
-  A. T. PATERA AND G. ROZZA, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*, Version 1.0, MIT 2006.
-  Y. ZHANG, L. FENG, S. LI, AND P. BENNER, *Accelerating PDE constrained optimization by the reduced basis method: application to batch chromatography*, to appear in International Journal for Numerical Methods in Engineering, 2015.
-  Y. ZHANG, L. FENG, S. LI, AND P. BENNER, *An efficient output error bound for the reduced basis method applied to parametrized evolution equations*, submitted to SIAM Journal on Scientific Computing.

And many more