Stabilized Projection-Based Reduced Order Models for Uncertainty Quantification

Irina Kalashnikova¹, Bart G. van Bloemen Waanders², Srinivasan Arunajatesan³, Matthew F. Barone³

¹ Computational Mathematics Department, Sandia National Laboratories, Albuquerque, NM.
² Optimization and UQ Department, Sandia National Laboratories, Albuquerque, NM.
³ Aerosciences Department, Sandia National Laboratories, Albuquerque NM.

SIAM Conference on Uncertainty Quantification (UQ14)
Savannah, Georgia
SAND 2014-2221C
Motivation for Model Reduction in Uncertainty Quantification

• Many real applications require Bayesian inference of high-dimensional random fields.

**Objective:** Given some measured output quantities of interest with noise, estimate the inputs that generated these outputs and their posterior distribution.

• Bayesian inference tools cannot handle high-dimensional parameter spaces → **curse of dimensionality!**

• Every proposed point in MCMC sampling requires a high-fidelity forward solve → **intractable!**

**Reduced Order Models (ROMs) can circumvent these difficulties:**

• Reduce large-dimensional inversion problem to small-dimensional problem by representing unknown input in reduced basis (Karhunen-Loeve/Proper Orthogonal Decomposition).

• Replace high-fidelity forward solve in MCMC algorithm with ROM.

Greenland Ice Sheet Example

• **Measured output:** surface velocity
• **Unknown input:** basal sliding coefficient at bedrock
Proper Orthogonal Decomposition (POD)/Galerkin Method to Model Reduction

Snapshots matrix: $X = (x^1, \ldots, x^K) \in \mathbb{R}^{N \times K}$
SVD: $X = U \Sigma V^T$
Truncation: $\Phi_M = (\phi_1, \ldots, \phi_M) = U(:, 1:M)$

Step 1: Modal Decomposition (POD):
$x(t) \approx \Phi_M x_M(t)$

Step 2: Galerkin Projection of LTI FOM:
$\Phi_M^T \ddot{x}(t) = Ax(t) + Bu(t)$

“Small” ROM LTI System:
$\dot{x}_M(t) = \Phi_M^T A \Phi_M x_M(t) + \Phi_M^T Bu(t)$
$y_M(t) = C \Phi_M x_M(t)$

High-Fidelity Simulations:
- Snapshot 1
- Snapshot 2
- ...,
- Snapshot $K$

$N = \#$ of dofs in high-fidelity simulation
$K = \#$ of snapshots
$M = \#$ of dofs in ROM
($M << N, M << K$)
Stability Issues of POD/Galerkin ROMs

**LTI Full Order Model (FOM)**

\[
\dot{x}(t) = Ax(t) + Bu(t) \\
y(t) = Cx(t)
\]

**LTI Reduced Order Model (ROM)**

\[
\dot{x}_M(t) = A_M x_M(t) + B_M u(t) \\
y_M(t) = C_M x_M(t)
\]

- ROM Linear Time-Invariant (LTI) system matrices given by:

  \[
  A_M = \Phi_M^T A \Phi_M, \quad B_M = \Phi_M^T B, \quad C_M = C \Phi_M
  \]

  **Problem:** \( A \) stable \( \not\Rightarrow \) \( A_M \) stable!

- There is no \textit{a priori} stability guarantee for POD/Galerkin ROMs.

- Stability of a ROM is commonly evaluated \textit{a posteriori} – \textbf{RISKY}!

- Instability of POD/Galerkin ROMs is a \textbf{real} problem in some applications...

  \[\ldots\text{e.g., compressible cavity flows, high-Reynolds number flows,} \ldots\]
Stability Preserving ROM Approaches: Literature Review

Approaches for building stability-preserving POD/Galerkin ROMs found in the literature fall into **two categories:**

1. ROMs which derive *a priori* a stability-preserving model reduction framework (usually specific to an equation set).
   - ROMs based on projection in special ‘energy-based’ (not $L^2$) inner products, e.g., Rowley *et al.* (2004), Barone & Kalashnikova *et al.* (2009), Serre *et al.* (2012).

2. ROMs which stabilize an unstable ROM through an *a posteriori* post-processing stabilization step applied to the algebraic ROM system.
   - Petrov-Galerkin ROMs that solve an optimization problem for the test basis given a trial POD basis, e.g., Amsallem *et al.* (2012), Bond *et al.* (2008).
   - ROMs with increased numerical stability due to inclusion of ‘stabilizing’ terms in the ROM equations, e.g., Wang *et al.* (2012).
**New Approach**: ROM Stabilization via Optimization-Based Eigenvalue Reassignment

- Approach falls in 2\textsuperscript{nd} category of stabilization methods, but ensures stabilized ROM solution deviates minimally from FOM solution.

- Recall that the ROM LTI system is given by:

\[
\begin{align*}
\dot{x}_M(t) &= \tilde{A}_M x_M(t) + B_M u(t) \\
y_M(t) &= C_M x_M(t)
\end{align*}
\]

An exact solution to the ROM LTI system can be derived using the matrix exponential.

- The solution to the ROM LTI system is:

\[
\begin{align*}
x_M(t) &= \exp(tA_M) x_M(0) + \int_0^t \exp((t - \tau)A_M)B_M u(\tau)d\tau \\
\Rightarrow y_M(t) &= C_M \left[ \exp(tA_M) x_M(0) + \int_0^t \exp((t - \tau)A_M)B_M u(\tau)d\tau \right]
\end{align*}
\]

**Goal**: replace unstable $A_M$ with stable $\tilde{A}_M$ so discrepancy b/w ROM output $y_M(t)$ and FOM output $y(t)$ is minimal.

---

ROM Stabilization via Optimization-Based Eigenvalue Reassignment (continued)

ROM Stabilization Optimization Problem
(Constrained Nonlinear Least Squares):

\[
\min \sum_{k=1}^{K} ||y^k - y_M^k||_2^2
\]

s. t. \(Re(\lambda_{i}^u) < 0\) \hfill (1)

- \(\lambda_{i}^u\) = unstable eigenvalues of original ROM matrix \(A_M\).
- \(y_k = y(t_k) = \) snapshot output at \(t_k\).
- \(y_M^k = C_M = \exp(t_kA_M)x_M(0) + \int_0^{t_k} \exp((t_k - \tau)A_M)B_Mu(\tau)d\tau\) = ROM output at \(t_k\).
- ROM stabilization optimization problem is small: < \(O(M)\).
- ROM stabilization optimization problem can be solved by standard optimization algorithms, e.g., interior point method.
  - We use \texttt{fmincon} function in MATLAB’s optimization toolbox.
  - We implement ROM stabilization optimization problem in \textit{characteristic variables} \(z_M(t) = S_M^{-1}x_M(t) \) where \(A_M = S_MD_MS_M^{-1}\).
Algorithm

- Diagonalize the ROM matrix $A_M$: $A_M = S_M D_M S_M^{-1}$.
- Initialize a diagonal $M \times M$ matrix $\tilde{D}_M$. Set $j = 1$.
- for $i = 1$ to $M$
  - if $\text{Re}(D_M(i,i) < 0)$, set $\tilde{D}_M(i,i) = D_M(i,i)$.
  - else, set $\tilde{D}_M(i,i) = \lambda_j^u$.
- Increment $j \leftarrow j + 1$.
- Solve the optimization problem (1) for the eigenvalues $\{\lambda_j^u\}$ using an optimization algorithm (e.g., interior point method).
- Evaluate $\tilde{D}_M$ at the solution of the optimization problem (1).
- Return the stabilized ROM system, given by $A_M \leftarrow \tilde{A}_M = S_M \tilde{D}_M S_M^{-1}$.

- Solution to optimization problem (1) may not be unique.
- Can solve (1) for real or complex-conjugate pair eigenvalues:
  - $\lambda_j^u \in \mathbb{R}$ s.t. constraint $\lambda_j^u < 0$.
  - $\lambda_j^u = \lambda_j^{ur} + i \lambda_j^{uc}$, $\lambda_j^{ur} + 1^{u} = \lambda_j^{ur} - i \lambda_j^{uc} \in \mathbb{C}$ where $\lambda_j^{ur}, \lambda_j^{uc} \in \mathbb{R}$ s.t. constraint $\lambda_j^{ur} < 0$. 


Numerical Results #1: International Space Station (ISS) Benchmark

- FOM: structural model of component 1r of the International Space Station (ISS).
- $A, C$ matrices defining FOM downloaded from NICONET ROM benchmark repository*.
- No inputs (unforced), 1 output; FOM is stable.

*NICONET ROM benchmark repository: [www.icm.tu-bs.de/NICONET/benchmodred.html](http://www.icm.tu-bs.de/NICONET/benchmodred.html).
• $M = 20$ POD/Galerkin ROM constructed from $K = 2000$ snapshots up to time $t = 0.1$.
• $M = 20$ POD/Galerkin ROM has 4 unstable eigenvalues: 2 real, 2 complex
  
• Two options for ROM stabilization optimization problem:
  
  **Option 1:** Solve for $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$ s.t. the constraint $\lambda_1, \lambda_2, \lambda_3, \lambda_4 < 0$.
  
  **Option 2:** Solve for $\lambda_1 + \lambda_2 i, \lambda_1 - \lambda_2 i \in \mathbb{C}, \lambda_3, \lambda_4 \in \mathbb{R}$ s.t. the constraint $\lambda_1, \lambda_3, \lambda_4 < 0$.
  
• Initial guess for fmincon interior point method: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = -1$.

| ROM                                | $\sqrt{\sum_{k=1}^{K} ||y^k - y^k_M||_2^2}$ |
|------------------------------------|---------------------------------------------|
| Unstabilized POD                   | 1737.8                                      |
| Optimization Stabilized POD (Real Poles) | 0.0259                                      |
| Optimization Stabilized POD (Complex-Conjugate Poles) | 0.0252                                      |
Numerical Results #2: Electrostatically Actuated Beam Benchmark

- FOM = 1D model of electrostatically actuated beam.
- Application of model: microelectromechanical systems (MEMS) devices such as electromechanical radio frequency (RF) filters.
- 1 input corresponding to periodic on/off switching, 1 output, initial condition $x(0) = 0_N$.
- Second order linear semi-discrete system of the form:
  \[
  M\ddot{x}(t) + E\dot{x}(t) + Kx(t) = Bu(t) \\
  y(t) = Cx(t)
  \]
- Matrices $M, E, K, B, C$ specifying the problem downloaded from the Oberwolfach ROM repository*.
- 2nd order linear system re-written as 1st order LTI system for purpose of analysis/model reduction.
- FOM is stable.

Numerical Results #2: Electrostatically Actuated Beam Benchmark (continued)

- $M = 17$ POD/Galerkin ROM constructed from $K = 1000$ snapshots up to time $t = 0.05$.
- $M = 17$ POD/Galerkin ROM has 4 unstable eigenvalues (all real).
  - Two options for ROM stabilization optimization problem:
    - **Option 1**: Solve for $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathbb{R}$ s.t. the constraint $\lambda_1, \lambda_2, \lambda_3, \lambda_4 < 0$.
    - **Option 2**: Solve for $\lambda_1 + \lambda_2 i, \lambda_1 - \lambda_2 i, \lambda_3 + \lambda_4 i, \lambda_3 - \lambda_4 i \in \mathbb{C}$ s.t. the constraint $\lambda_1, \lambda_3 < 0$.
- Initial guess for `fmincon` interior point method: $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = -1$.

| ROM | $\sqrt{\sum_{k=1}^{K} ||y_k - y_{M,k}||_2^2} / \sqrt{\sum_{k=1}^{K} ||y_k||_2^2}$ |
|-----|--------------------------------------------------|
| Unstabilized POD | $NaN$ |
| Optimization Stabilized POD (Real Poles) | 0.0194 |
| Optimization Stabilized POD (Complex-Conjugate Poles) | 0.0205 |
| Balanced Truncation | $1.370e-6$ |
Summary & Future Work

- A new ROM stabilization approach that modifies \textit{a posteriori} an unstable ROM LTI system by changing the system’s unstable eigenvalues is proposed.
- In the proposed stabilization algorithm, a constrained nonlinear least squares optimization problem for the ROM eigenvalues is formulated to minimize error in ROM output.
- Excellent performance of the proposed algorithm is evaluated on two benchmarks.
- Paper on the proposed new method was just published in CMAME!


Ongoing/Future work

- Applications to \textit{Uncertainty Quantification (UQ)}.
- Studies of predictive capabilities of stabilized ROMs (robustness w.r.t. parameter changes).
- Extensions to nonlinear problems.
Acknowledgements

• This work was funded by Laboratories’ Directed Research and Development (LDRD) Program at Sandia National Laboratories.

• Special thanks to

  • Prof. Lou Cattafesta (Florida State University)
  • Prof. Karen Willcox (MIT)

for useful discussions that led to some of the ideas presented here.

Thank You! Questions?
ikalash@sandia.gov
http://www.sandia.gov/~ikalash

References


References (continued)


• NICONET ROM benchmark repository: [www.icm.tu-bs.de/NICONET/benchmodred.html](http://www.icm.tu-bs.de/NICONET/benchmodred.html).

Appendix: ISS Benchmark (fmincon performance)

<table>
<thead>
<tr>
<th></th>
<th>Real Poles</th>
<th>Complex-Conjugate Poles</th>
</tr>
</thead>
<tbody>
<tr>
<td># upper bound</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td># iterations</td>
<td>29</td>
<td>27</td>
</tr>
<tr>
<td># function</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>evaluations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\nabla L</td>
<td>$ at convergence</td>
</tr>
</tbody>
</table>

(1st order optimality)

Current Function Value

- Real Poles: $0.00683859$
- Complex-Conjugate Poles: $0.00640948$

First-Order Optimality

- Real Poles: $4.00842e-07$
- Complex-Conjugate Poles: $5.50885e-07$
### Appendix: ISS Benchmark (CPU Times)

<table>
<thead>
<tr>
<th>Model</th>
<th>Operations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOM</td>
<td>Time-Integration</td>
<td>1.71e2</td>
</tr>
<tr>
<td>ROM – offline stage</td>
<td>Snapshot collection (FOM time-integration)</td>
<td>1.71e2</td>
</tr>
<tr>
<td></td>
<td>Loading of matrices/snapshots</td>
<td>6.99e-2</td>
</tr>
<tr>
<td></td>
<td>POD</td>
<td>6.20</td>
</tr>
<tr>
<td></td>
<td>Projection</td>
<td>8.18e-3</td>
</tr>
<tr>
<td></td>
<td>Optimization</td>
<td>2.28e1</td>
</tr>
<tr>
<td>ROM – online stage</td>
<td>Time-integration</td>
<td>3.77</td>
</tr>
</tbody>
</table>

- To offset total pre-process time of ROM (time required to run FOM to collect snapshots, calculate the POD basis, perform the Galerkin projection, and solve the optimization problem (1)), the ROM would need to be run 53 times.

- Solution of optimization problem is very fast: takes < 1 minute to complete.
Appendix: Electrostatically Actuated Beam Benchmark \textit{(fmincon performance)}

<table>
<thead>
<tr>
<th></th>
<th>Real Poles</th>
<th>Complex-Conjugate Poles</th>
</tr>
</thead>
<tbody>
<tr>
<td># upper bound constraints</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td># iterations</td>
<td>60</td>
<td>31</td>
</tr>
<tr>
<td># function evaluations</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>$</td>
<td>\nabla L</td>
<td>$ at convergence (1st order optimality)</td>
</tr>
</tbody>
</table>

Current Function Value

First-Order Optimality

Current Function Value

First-Order Optimality

Current Function Value

First-Order Optimality

Current Function Value

First-Order Optimality

Current Function Value

First-Order Optimality

Current Function Value

First-Order Optimality
## Appendix: Electrostatically Actuated Beam Benchmark (CPU Times)

<table>
<thead>
<tr>
<th>Model</th>
<th>Operations</th>
<th>CPU time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOM</td>
<td>Time-Integration</td>
<td>7.10e4</td>
</tr>
<tr>
<td>ROM – offline stage</td>
<td>Snapshot collection (FOM time-integration)</td>
<td>7.10e4</td>
</tr>
<tr>
<td></td>
<td>Loading of matrices/snapshots</td>
<td>5.17</td>
</tr>
<tr>
<td></td>
<td>POD</td>
<td>1.09e1</td>
</tr>
<tr>
<td></td>
<td>Projection</td>
<td>2.55e1</td>
</tr>
<tr>
<td></td>
<td>Optimization</td>
<td>8.79e1</td>
</tr>
<tr>
<td>ROM – online stage</td>
<td>Time-integration</td>
<td>6.78</td>
</tr>
</tbody>
</table>

- To offset total pre-process time of ROM (time required to run FOM to collect snapshots, calculate the POD basis, perform the Galerkin projection, and solve the optimization problem (1)), the ROM would need to be run 1e4 times (due to large CPU time of FOM).
- Solution of optimization problem is very fast: takes ~1.5 minute to complete.
Appendix: Electrostatically Actuated Beam Benchmark (Eigenvalues)

Unstable Eigenvalues

<table>
<thead>
<tr>
<th>( \lambda_6 )</th>
<th>( \lambda_{12} )</th>
<th>( \lambda_{14} )</th>
<th>( \lambda_{17} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.053</td>
<td>48.985</td>
<td>12.650</td>
<td>0.05202</td>
</tr>
</tbody>
</table>

Stabilized Eigenvalues (Real)

<table>
<thead>
<tr>
<th>( \lambda_6 )</th>
<th>( \lambda_{12} )</th>
<th>( \lambda_{14} )</th>
<th>( \lambda_{17} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7,043,505</td>
<td>-35.364</td>
<td>-153,033</td>
<td>-99,175</td>
</tr>
</tbody>
</table>

Stabilized Eigenvalues (Complex Conjugates)

<table>
<thead>
<tr>
<th>( \lambda_6 )</th>
<th>( \lambda_{12} )</th>
<th>( \lambda_{14} )</th>
<th>( \lambda_{17} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-106,976 + 551.77i</td>
<td>-106,976 - 551.77i</td>
<td>-2954.1 - 1244.7i</td>
<td>-2954.1 + 1244.7i</td>
</tr>
</tbody>
</table>