Construction of Parametrically-Robust Reduced-Order Models via Nonlinear Programming

Matthew J. Zahr, David Amsallem, Charbel Farhat

Farhat Research Group
Stanford University

CME 334 Presentation
13th November 2012
Outline

1 Introduction
2 Motivation
3 Optimization-Based ROM Construction
4 Digression: Latin Hypercube Sampling
5 Application
6 Conclusion
2.4. Consistency-driven approach for nonlinear model reduction

Model reduction of nonlinear systems is often executed in a somewhat ad hoc manner, where approximations are constructed using intuition and past experience without much reference to properties that a "good" approximation should satisfy. To avoid this pitfall, this work adopts a strategy that enables approximations to be carefully constructed to meet desired conditions. In the proposed approach, if a given model is deemed too computationally expensive for real-time evaluation, an additional approximation is introduced, resulting in another less accurate but more economical model. This results in a hierarchy of models characterized by tradeoffs between accuracy and computational efficiency. The approximations, which are introduced consecutively, are constructed to generate minimal error with respect to the previous model by satisfying optimality and consistency properties that are defined more precisely below.

As shown in Figure 1, the model hierarchy employed in this work consists of three computational models: an original model, and two increasingly "lighter" approximated versions. Each approximated model is generated by acquiring data during the evaluation of the more accurate model for sample inputs, then compressing the data, and finally introducing the approximation that exploits the compressed data.

The high-dimensional model will be referred to as Model I and is taken to be the "truth." When evaluating this model is too computationally intensive for real-time prediction, a projection approximation (Approximation 1) is introduced to reduce the dimensionality of the state equations. This leads to the reduced-order model (ROM), or Model II. If this ROM is still too CPU intensive for online computations, a system approximation (Approximation 2) is introduced to reduce the computational complexity of its processing. The result of the application of this system approximation to Model II can be interpreted as a computational model and therefore will be referred to as Model III in the remainder of this paper.

As previously stated, the approximations should introduce minimal error with respect to the previous model in the hierarchy. To this end, Approximations 1 and 2 will be constructed to be: 1) consistent, and 2) optimal in the sense defined below.

- **Consistent approximation**: An approximation is said here to be consistent if, when implemented without data compression, it introduces no additional error in the solution of the same problem for which data was acquired.

- **Optimal approximation**: An approximation is said here to be optimal if it leads to approximated quantities that minimize some error measure with respect to the previous model in the hierarchy.
We consider the discretization of a *steady* PDE:

\[ c(w, p) = 0, \]

where \( c : \mathbb{R}^{N_w} \times \mathbb{R}^p \rightarrow \mathbb{R}^{N_w}, \ p \in \mathcal{D} \subseteq \mathbb{R}^p \) is a vector of parameters, and \( N_w \) is typically very large. This will be referred to as the High-Dimensional Model (HDM) or Model I.

Let us denote the solution of the Reduced-Order Model (ROM) and hyperreduced ROM as:

- \( c_r(w_r, p) = 0 \) \hspace{1cm} Model II
- \( c_h(w_r, p) = 0 \) \hspace{1cm} Model III

respectively, where \( c_r : \mathbb{R}^{k_w} \times \mathbb{R}^p \rightarrow \mathbb{R}^{k_w} \) and \( c_h : \mathbb{R}^{k_w} \times \mathbb{R}^p \rightarrow \mathbb{R}^{k_w} \).
Model Order Reduction Assumption

**MOR assumption:** the solution of $c(w, p) = 0$ lies in a low-dimensional affine subspace

$$w = \bar{w} + Vw_r$$  \hspace{1cm} (1)

where

- $V \in \mathbb{R}^{N_w \times k_w}$ — right basis
- $w_r \in \mathbb{R}^{k_w}$ — reduced coordinates
- $k_w \ll N_w$.

Substituting this assumption into the HDM yields the overdetermined nonlinear system of equations

$$c(\bar{w} + Vw_r, p) = 0$$  \hspace{1cm} (2)
Now, we close the previous equation by requiring that the residual be orthogonal to the subspace spanned by the columns of some matrix $W \in \mathbb{R}^{N_w \times k_w}$

$$c_r(w_r, p) = W^T c(\bar{w} + Vw_r, p) = 0.$$  \hspace{0.5cm} (3)

Two standard choices for $W$:

- $W = V$ \hspace{1cm} \text{Galerkin ROM}
  - “Optimal” for problems with SPD Jacobians
- $W = \frac{\partial c}{\partial w} V$ \hspace{1cm} \text{Least-Squares Petrov-Galerkin ROM}
  - “Optimal” for problems with non-SPD Jacobians
ROM Construction Philosophy

- ROM Construction usually considered an “offline” cost
  - Computation time allowed to be very large
    - Real-time ROM applications
- In many-query analyses with ROMs, the only important time is the total time from the beginning of ROM training to the end of the ROM optimization
  - ROM construction is time-critical
    - Optimization
    - Uncertainty Quantification
The following is a standard algorithm for constructing a Reduced Order Model.

**Algorithm 1** Generic ROM Construction

Input: $k_w$, size of the Reduced Order Basis  
Output: $V$, the Reduced Order Basis  
Select $k$ training points: $\{p_1, p_2, \ldots, p_k\} \subset \mathcal{D}$  
for $j = 1, 2, \ldots, k$ do  
    Solve $c(w, p_j) = 0 \rightarrow \hat{w}_j$  
end for  
$X = [\hat{w}_1 \  \hat{w}_2 \ \cdots \ \hat{w}_k]$  
Compute SVD of $X$: $X = U\Sigma V^T$  
$V = U(:, 1 : k_w)$ (Compression)
Suppose we construct a ROM from sampling \( k \) training points \( \{p_1, \ldots, p_k\} \).

- ROM is likely to perform well for test points, \( \bar{p} \), “close” to one of the training points, i.e. for \( \bar{p} \in \{p \in D \mid \exists j \in \{1, \ldots, k\} \text{ such that } ||p - p_j|| < \epsilon_j\} \)

- It is well-documented that, for test points not satisfying this criterion, ROMs tend to perform poorly.

Thus, ROMs suffer from non-robustness from training points.
The overall goal of this project is the use of ROMs as a surrogate for the HDM in an optimization algorithm.

Optimization algorithms all reduce to a search in the parameter space for a local optimum.

For ROMs to be useful for optimization, they must have parametric-robustness and their construction cannot be prohibitively expensive.
Due to the large cost required to train a ROM and the general lack of parametric robustness, we state the design of experiments goal:

*Construct a parametrically-robust ROM using as few sample points as possible*

We use a greedy, optimization-based algorithm for the training of a ROM.
Outline

1 Introduction
2 Motivation
3 Optimization-Based ROM Construction
4 Digression: Latin Hypercube Sampling
5 Application
6 Conclusion

Zahr, Amsallem, Farhat
Parametrically-Robust ROMs
Suppose $V$ is constructed from $\{p_1, p_2, \ldots, p_k\}$, a very coarse sampling of the parameter space.

**Algorithm 2** Progressive Sampling ROM Construction

**Input:** original ROM, $V$, $\{p_1, p_2, \ldots, p_k\}$

**Output:** new ROM, $V$

1. while Not Converged do
2.  Determine $p \in D$, where ROM error is largest $\rightarrow p^*$
3.  Solve $c(w, p^*) = 0 \rightarrow \hat{w}$ (Snapshots)
4.  Use $\hat{w}$ to update $V$
5. end while
Suppose $V$ is constructed from $\{p_1, p_2, \ldots, p_k\}$, a very coarse sampling of the parameter space.

**Algorithm 3 Progressive Sampling ROM Construction**

**Input:** original ROM, $V$, $\{p_1, p_2, \ldots, p_k\}$

**Output:** new ROM, $V$

1. while Not Converged do
2. Determine $p \in D$, where ROM error is largest $\rightarrow p^*$
3. Solve $c(w, p^*) = 0 \rightarrow \hat{w}$ (Snapshots)
4. Use $\hat{w}$ to update $V$
5. end while
HDM-Constrained Sampling Approach

Given $V$ generated from a very coarse sampling of parameters $\{p_1, p_2, \ldots, p_k\}$, solve

$$\underset{p \in \mathcal{D}, \ w \in \mathbb{R}^{Nw}, \ w_r \in \mathbb{R}^{kw}}{\text{maximize}} \quad \frac{1}{2} \|w - (\bar{w} + Vw_r)\|_2^2$$

subject to

$$c(w, p) = 0$$
$$c_r(w_r, p) = 0.$$  \hspace{1cm} (4)

Output: A parameter $p^*$ where the error is largest.

Cost: Every iteration requires HDM and ROM solution.
Given \( V \) generated from a very coarse sampling of parameters \( \{p_1, p_2, \ldots, p_k\} \), solve

\[
\begin{align*}
\text{maximize} \quad & \frac{1}{2} \left\| c(\bar{w} + Vw_r, p) \right\|^2_2 \\
\text{subject to} \quad & c_r(w_r, p) = 0
\end{align*}
\]

\( (5) \)

**Output**: A parameter \( p^* \) where the HDM residual evaluated at ROM solution is largest.

**Cost**: Every iteration requires ROM solution.
Difficulty of ROM-Constrained Sampling Objective

The objective function in (5) is, in general, non-concave and not even defined throughout the entire domain because the ROM may fail away from training points.

\[ \frac{1}{2} \| [c(w_0 + Vw_r, p)] \|^2 \]

---

Objective Function of ROM-Constrained Problem
Outline

1. Introduction
2. Motivation
3. Optimization-Based ROM Construction
4. Digression: Latin Hypercube Sampling
5. Application
6. Conclusion
LHS Demonstration
LHS Demonstration
LHS Demonstration

![Latin Hypercube Sampling Diagram](image)
LHS Demonstration
LHS Demonstration
LHS Demonstration

![LHS Demonstration Diagram](image-url)
LHS Demonstration
LHS Demonstration
LHS Demonstration
LHS Demonstration
LHS Demonstration
LHS Demonstration
### LHS Demonstration

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Zahr, Amsallem, Farhat**

**Parametrically-Robust ROMs**
Outline

1. Introduction
2. Motivation
3. Optimization-Based ROM Construction
4. Digression: Latin Hypercube Sampling
5. Application
6. Conclusion
Potential Nozzle Flow - Cubic Spline Parametrization

\[
\frac{d}{dx} (A(x)\rho(x)u(x)) = 0
\]  

(6)

Nozzle Configuration and Bounds
Here, we construct two of ROMs: $\mathcal{R}$ and $\tilde{\mathcal{R}}$.

$\mathcal{R}$ is constructed with $n_s$ sample points using the ROM-Constrained sampling and $\tilde{\mathcal{R}}$ is constructed with $n_s$ sample points using LHS sampling.

Generate set $\mathcal{Z} \subset \mathcal{D}$, where $|\mathcal{Z}| = 500$. This is the test set.

$\mathcal{R}_i$ and $\tilde{\mathcal{R}}_i$ are tested on the set of parameters $\mathcal{Z}$.
Sampled Nozzle Shapes

**Figure:** Optimization-Based Sampling

**Figure:** Latin Hypercube Sampling
Mach Distribution of Sampled Nozzle Shapes

**Figure:** Optimization-Based Sampling

**Figure:** Latin Hypercube Sampling

Zahr, Amsallem, Farhat

Parametrically-Robust ROMs
Introduction

Motivation

Optimization-Based ROM Construction

Digression: Latin Hypercube Sampling

Application

Conclusion

ROMOpt Sample Experiment Results

% of Samples

L2-Error between HDM and ROM (%)

OptSample ROM
LHS ROM

Failure Rate of OptSample ROM = 1%
Failure Rate of LHS ROM = 17%

Zahr, Amsallem, Farhat
Parametrically-Robust ROMs
Introduction

Motivation

Optimization-Based ROM Construction

Digression: Latin Hypercube Sampling

Application

Conclusion

Zahr, Amsallem, Farhat

Parametrically-Robust ROMs
Implementation of ROM optimization-based sampling builds Reduced Order Models that tend to cover the parameter space much better than a simple randomized sampling algorithm.