Approximate yet accurate surrogates for large-scale simulation

KAREN E. WILLCOX
Science at Extreme Scales: Where Big Data Meets Large-Scale Computing Tutorials
Institute for Pure and Applied Mathematics
September 17, 2018
1. Motivation
2. General projection framework
3. Computing the basis
4. Approximating nonlinear terms
5. Error analysis and guarantees
6. Adaptive data-driven ROMs
7. Challenges
1. Motivation

Use cases and benefits of ROMs
Outer-loop applications

“Computational applications that form outer loops around a model – where in each iteration an input \( z \) is received and the corresponding model output \( y = f(z) \) is computed, and an overall outer-loop result is obtained at the termination of the outer loop.”

Peherstorfer, W., Gunzburger, SIAM Review, 2018

Examples
- Optimization
  outer-loop result = optimal design
- Uncertainty propagation
  outer-loop result = estimate of statistics of interest
- Inverse problems
- Data assimilation
- Control problems
- Sensitivity analysis

Diagram:
- Input \( z \) to forward model \( f \) which outputs \( y \).
- Feedback loop from \( y \) to outer-loop application.
New Technologies + Data + Computational Power

*a revolution in the world around us*

needing new data-enabled computational science and engineering
Data + Models: real-time adaptive emergency response

Data + Models: real-time adaptive teaching & learning

U.S. Department of Education First in the World Fly-by-Wire project  fbw.mit.edu
Data + Models: self-aware aerospace vehicles

SENSE ➔ INFER ➔ PREDICT ➔ ACT

Singh & W., AIAA J., 2017
Model reduction leverages an offline/online decomposition of tasks

**Offline**
- Generate snapshots/libraries, using high-fidelity models
- Generate reduced models

**Online**
- Select appropriate library records and/or reduced models
- Rapid {prediction, control, optimization, UQ} using multi-fidelity models
Reduced models enable rapid prediction, inversion, design, and uncertainty quantification of large-scale scientific and engineering systems.

1 modeling the data-to-decisions flow 2 exploiting synergies between physics-based models & data 3 principled approximations to reduce computational cost 4 explicit modeling & treatment of uncertainty
2. Projection-based model reduction

extracting the essence of complex problems to make them easier and faster to solve
Start with a physics-based model

large-scale and expensive to solve

Arising, for example, from systems of ODEs or spatial discretization of PDEs describing the system of interest

- which in turn arise from governing physical principles (conservation laws, etc.)

\[
\begin{align*}
\dot{x} &= A(p)x + B(p)u \\
y &= C(p)x
\end{align*}
\]

\[
\begin{align*}
\dot{x} &= f(x, p, u) \\
y &= g(x, p, u)
\end{align*}
\]

\[x \in \mathbb{R}^N: \text{state vector}\]
\[u \in \mathbb{R}^{N_i}: \text{input vector}\]
\[p \in \mathbb{R}^{N_p}: \text{parameter vector}\]
\[y \in \mathbb{R}^{N_o}: \text{output vector}\]
### Example: CFD systems

**modeling the flow over an aircraft wing**

<table>
<thead>
<tr>
<th>$\dot{x} = A(p)x + B(p)u$</th>
<th>$\dot{x} = f(x, p, u)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y = C(p)x$</td>
<td>$y = g(x, p, u)$</td>
</tr>
</tbody>
</table>

- $x(t)$: vector of $N$ flow unknowns  
  e.g., 2D incompressible Navier Stokes  
  $P$ grid points, $N = 3P$  
  $x = [u_1 \, v_1 \, p_1 \, u_2 \, v_2 \, p_2 \, \cdots \, u_P \, v_P \, p_P]^T$

- $p$: input parameters  
  e.g., shape parameters, PDE coefficients

- $u(t)$: forcing inputs  
  e.g., flow disturbances, wing motion

- $y(t)$: outputs  
  e.g., flow characteristic, lift force
Example: modeling combustion instability

\[ \dot{x} = A(p)x + B(p)u \quad \dot{x} = f(x, p, u) \]
\[ y = C(p)x \quad y = g(x, p, u) \]

- \( x(t) \): vector of \( N \) reacting flow unknowns \( p', u', T', Y_{ox}' \) discretized over computational domain
- \( p \): input parameters
e.g., fuel-to-oxidizer ratio, combustion zone length, fuel temperature, oxidizer temperature
- \( u(t) \): forcing inputs
e.g., periodic oscillation of inlet mass flow rate, stagnation temperature, back pressure
- \( y(t) \): output quantities of interest
e.g., pressure oscillation at sensor location
Which states are important?

Is there a low-dimensional structure underlying the input-output map?

"Controllable" modes ("Reachable" modes)
- easy to reach, require small control energy
- dominant eigenmodes of a controllability gramian matrix

"Observable" modes
- generate large output energy
- dominant eigenmodes of an observability gramian matrix

\[ \dot{x} = Ax + Bu \]
\[ y = Cx \]
Which states are important?

Is there a low-dimensional structure underlying the input-output map?

• Rigorous theories and scalable algorithms in the linear time-invariant (LTI) case
  – Hankel singular values

• Strong foundations for linear parameter-varying (LPV) systems
  – handling high-dimensional parameters can be a challenge

• Many open questions for the nonlinear case
  – linear methods are founded on the notion of a low-dimensional subspace
  – works well for some nonlinear problems but certainly not all
  – additional challenges related to efficient solution of the ROM
Reduced models

Low-cost but accurate approximations of high-fidelity models via projection onto a low-dimensional subspace.

\[
\begin{align*}
\dot{x} &= A(p)x + B(p)u \\
y &= C(p)x \\
x &\approx Vx_r
\end{align*}
\]

\[
\begin{align*}
r &= V\dot{x}_r - AVx_r - Bu \\
y_r &= CVx_r \\
W^T r &= 0
\end{align*}
\]

\[
\begin{align*}
A_r(p) &= W^T A(p)V \\
B_r(p) &= W^T B(p) \\
C_r(p) &= C(p)V \\
\dot{x}_r &= A_r(p)x_r + B_r(p)u \\
y_r &= C_r(p)x_r
\end{align*}
\]

- \(x \in \mathbb{R}^N\): state vector
- \(p \in \mathbb{R}^{N_p}\): parameter vector
- \(u \in \mathbb{R}^{N_i}\): input vector
- \(y \in \mathbb{R}^{N_o}\): output vector
- \(x_r \in \mathbb{R}^n\): reduced state vector
- \(V \in \mathbb{R}^{N \times n}\): reduced basis
What is the connection between reduced order modeling and machine learning?

**Machine learning**

“Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" with data, without being explicitly programmed.” [Wikipedia]

**Reduced order modeling**

“Model order reduction (MOR) is a technique for reducing the computational complexity of mathematical models in numerical simulations.” [Wikipedia]

The difference in fields is perhaps largely one of history and perspective: model reduction methods have grown from the scientific computing community, with a focus on reducing high-dimensional models that arise from physics-based modeling, whereas machine learning has grown from the computer science community, with a focus on creating low-dimensional models from black-box data streams. Yet recent years have seen an increased blending of the two perspectives and a recognition of the associated opportunities. [Swischuk et al., *Computers & Fluids*, 2018]
3. Computing the basis

Many different methods to identify the low-dimensional subspace
(Some) Large-Scale Reduction Methods

Different mathematical foundations lead to different ways to compute the basis and the reduced model.

  - use data to generate empirical eigenfunctions
  - time- and frequency-domain methods

- **Krylov-subspace methods** (Gallivan, Grimme, & van Dooren, 1994; Feldmann & Freund, 1995; Grimme, 1997, Gugercin et al., 2008)
  - rational interpolation

- **Balanced truncation** (Moore, 1981; Sorensen & Antoulas, 2002; Li & White, 2002)
  - guaranteed stability and error bound for LTI systems
  - close connection between POD and balanced truncation

- **Reduced basis methods** (Noor & Peters, 1980; Patera & Rozza, 2007)
  - strong focus on error estimation for specific PDEs

- **Eigensystem realization algorithm (ERA)** (Juang & Pappa, 1985), **Dynamic mode decomposition (DMD)** (Schmid, 2010), **Loewner model reduction** (Mayo & Antoulas, 2007)
  - data-driven, non-intrusive

Computing the Basis: Proper Orthogonal Decomposition (POD)

(aka Karhunen-Loève expansions, Principal Components Analysis, Empirical Orthogonal Eigenfunctions, ...)

- Consider $K$ snapshots $x_1, x_2, \ldots, x_K \in \mathbb{R}^N$ [Sirovich, 1991] (solutions at selected times or parameter values)

- Form the snapshot matrix $X = [x_1 \ x_2 \ \ldots \ x_K]

- Choose the $n$ basis vectors $V = [V_1 \ V_2 \ \cdots \ V_n]$ to be left singular vectors of the snapshot matrix, with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq \sigma_{n+1} \geq \cdots \geq \sigma_K$

- This is the optimal projection in a least squares sense:

$$\min_V \sum_{i=1}^{K} ||x_i - VV^T x_i||^2_2 = \sum_{i=n+1}^{K} \sigma_i^2$$
4. Nonlinear model reduction

General projection framework applies, but leads to complications
Projection-based nonlinear reduced models

approximation of high-fidelity models via projection onto a low-dimensional subspace

Projection-based nonlinear reduced models

\[ \dot{x} = f(x, p, u) \]
\[ y = g(x, p, u) \]

\[ x \approx V x_r \]

\[ r = V \dot{x}_r - f(V x_r, p, u) \]
\[ y_r = g(V x_r, p, u) \]

\[ W^T r = 0 \]

\[ \dot{x}_r = W^T f(V x_r, u) \]
\[ y_r = g(V x_r) \]

FOM

\[ x \in \mathbb{R}^N: \text{state vector} \]
\[ p \in \mathbb{R}^{N_p}: \text{parameter vector} \]
\[ u \in \mathbb{R}^{N_i}: \text{input vector} \]
\[ y \in \mathbb{R}^{N_o}: \text{output vector} \]

ROM

\[ x_r \in \mathbb{R}^n: \text{reduced state vector} \]
\[ V \in \mathbb{R}^{N \times n}: \text{reduced basis} \]

dimension is reduced, but evaluating nonlinear term still scales with large dimension \( N \)
Nonlinear POD ROMs

For nonlinear systems, standard POD projection approach leads to a model that is low order but still expensive to solve.

\[
\begin{aligned}
\dot{x} &= f(x, u) \\
y &= g(x)
\end{aligned}
\]

\[
\begin{aligned}
x &= Vx_r \\
\dot{x}_r &= V^T f(Vx_r, u) \\
y_r &= g(Vx_r)
\end{aligned}
\]

- The cost of evaluating the nonlinear term
  \[
  f_r(x_r, u) = V^T f(Vx_r, u)
  \]
  still depends on \( N \), the dimension of the large-scale system.

- Can achieve efficient nonlinear reduced models via interpolation, e.g., (Discrete) Empirical Interpolation Method [Barrault et al., 2004; Chaturantabut & Sorensen, 2010], Missing Point Estimation [Astrid et al., 2008], GNAT [Carlberg et al., 2013]

  \[
  \dot{x}_r = A_r x_r + E_r f_r(D_r x_r, u)
  \]
Discrete Empirical Interpolation Method (DEIM)

Additional layer of approximation to make the reduced-order nonlinear term fast to evaluate

Chaturantabut & Sorensen, *SISC*, 2010

\[ \begin{align*}
\dot{x} &= f(x, u) \\
y &= g(x)
\end{align*} \quad \Rightarrow \quad \begin{align*}
\dot{x}_r &= V^T f(V x_r, u) \\
y_r &= g(V x_r)
\end{align*} \]

• Collect snapshots of \( f(x, u) \); compute DEIM basis \( U \) for the nonlinear term (use POD to identify a linear subspace)

• Select \( m \) interpolation points in \( P \in \mathbb{R}^{m \times N} \) at which to sample \( f \)

• Approximate \( f_r(x_r, u) \):

\[
V^T f(V x_r, u) \approx \underbrace{V^T U (P^T U)^{-1}}_{n \times m \text{ (precompute)}} P^T f(V x_r, u)
\]

• Considerable success on a range of problems

• But some open challenges
  – for strongly nonlinear systems, require **so many DEIM points** that ROM is inefficient (e.g., Huang et al., *AIAA* 2018)
  – introduces **additional approximation**; difficult to **analyze** error convergence, stability, etc.
Linear Model

FOM: \( \dot{E}x = Ax + Bu \)

ROM: \( \hat{E}\dot{x} = \hat{A}\hat{x} + \hat{B}u \)

Precompute the ROM matrices:

\( \hat{A} = V^TAV, \hat{B} = V^TB, \hat{E} = V^TEV \)

Quadratic Model

FOM: \( \dot{E}x = Ax + Bu + H(x \otimes x) \)

ROM: \( \hat{E}\dot{x} = \hat{A}\hat{x} + \hat{B}u + \hat{H}(\hat{x} \otimes \hat{x}) \)

Precompute the ROM matrices and tensor:

\( \hat{H} = V^TH(V \otimes V) \)
Quadratic-bilinear (QB) systems

Advantages:
- efficient offline/online decomposition
- amenable to analysis (errors, stability, etc.)

\[
\begin{align*}
\text{FOM:} & \quad \dot{E}\dot{x} = A\dot{x} + Bu + H(x \otimes x) + \sum_{k=1}^{m} N_k \dot{x} u_k \\
& \quad \text{linear} \quad \text{quadratic} \quad \text{bilinear}
\end{align*}
\]

- Quadratic tensor \( H \in \mathbb{R}^{n \times n^2} \)
- Bilinear interaction: \( N_k \in \mathbb{R}^{n \times n}, \ k = 1, \ldots, m \)

\[
\begin{align*}
\text{ROM:} & \quad \dot{\hat{E}}\dot{\hat{x}} = \hat{A}\dot{\hat{x}} + \hat{B}u + \hat{H}(\hat{x} \otimes \hat{x}) + \sum_{k=1}^{m} \hat{N}_k \dot{\hat{x}} u_k \\
& \quad \hat{A} = V^T AV \quad \hat{N}_k = V^T N_k V \\
& \quad \hat{B} = V^T B \quad \hat{H} = V^T H(V \otimes V) \\
& \quad \hat{E} = V^T EV
\end{align*}
\]
Polynomial systems

Could keep going to higher order

Model becomes more complex but retains efficient offline/online decomposition

\[
\dot{x} = A x + B u + \sum_{i=1}^{m} \mathbf{G}^{(i)} (x \otimes x) u_k
\]

FOM:

\[
\dot{x} = A \hat{x} + \hat{B} u + \hat{H} (\hat{x} \otimes \hat{x})
\]

ROM:

Possibility to pre-compute reduced tensors is major advantage

\[
\hat{G}^{(4)} = V^T G^{(4)} \left( V \otimes V \otimes V \otimes V \right)
\]

\[
\hat{G}^{(3)} = V^T G^{(3)} \left( V \otimes V \otimes V \right)
\]
5. Error analysis and guarantees
(or lack thereof)
Error analysis and guarantees

What rigorous statements can we make about the quality of the reduced-order models?

• Strong theoretical foundations in the LTI case (error bounds, error estimators)
• Solid theoretical foundations for some classes of linear parametrized PDEs (error estimators)
• Error indicators may be available (e.g., residual)
• Few/no guarantees available otherwise
• Nonlinear systems are a particular challenge
• Many important open research questions
Error analysis and guarantees

What rigorous statements can we make about the quality of the reduced-order models?

• POD

• Reduced basis method has a strong focus on error estimates that exploit underlying structure of the PDE

Elliptic PDES:


Parabolic PDES:
6. Adaptive and Data-driven ROMs

Towards effective, efficient ROMs for a broader class of complex systems
Model reduction leverages an **offline/online** decomposition of tasks

**Offline**
- Generate snapshots/libraries, using high-fidelity models
- Generate reduced models

**Online**
- Select appropriate library records and/or reduced models
- Rapid {prediction, control, optimization, UQ} using multi-fidelity models
Classically

- Reduced models are built and used in a **static** way:
  - offline phase: sample a high-fidelity model, build a low-dimensional basis, project to build the reduced model
  - online phase: use the reduced model

Data-driven reduced models

- Recognize that conditions may change and/or initial reduced model may be inadequate
  - offline phase: build an initial reduced model
  - online phase: **learn** and **adapt** using dynamic data
A data-driven **offline/online** approach

**Offline**
- Generate snapshots/libraries, using high-fidelity models
- Generate reduced models

**Online**
- Dynamically collect data from sensors/simulations
- Classify system behavior
- Select appropriate library records and/or reduced models
- Rapid \{prediction, control, optimization, UQ\} using multi-fidelity models
- Adapt reduced models
- Adapt sensing strategies
Adaptation and learning are data-driven
• sensor data collected online (e.g., structural sensors on board an aircraft)
• simulation data collected online (e.g., over the path to an optimal solution)
but the physics-based model remains as an underpinning.

Achieve adaptation in a variety of ways:
• adapt the basis (Cui, Marzouk, W., 2014)
• adapt the way in which nonlinear terms are approximated (ADEIM: Peherstorfer, W., 2015)
• adapt the reduced model itself (Peherstorfer, W., 2015)
• construct localized reduced models; adapt model choice (LDEIM: Peherstorfer, Butnaru, W., Bungartz, 2014)
Consider a system with observable and latent parameters.
Classical approaches build the new reduced model from scratch.
A dynamic reduced model adapts in response to the data, without recourse to the full model.
Data-driven reduced models

- *adapt* directly from sensor data
- *avoid* (expensive) inference of latent parameter
- *avoid* recourse to full model

- incremental SVD methods (exploit structure of a rank-one snapshot update)
- operator inference methods (non-intrusive)
- convergence guarantees in idealized noise-free case
Example: locally damaged plate

High-fidelity: finite element model

Reduced model: proper orthogonal decomposition
Data-driven adaptation: locally damaged plate

Adapting the ROM after damage

Speedup of $10^4$ cf. rebuilding ROM
Localized and adaptive reduced models

- Automatic model management based on machine learning
  - **Cluster** set of snapshots $S = \{x_1, \ldots, x_M\} \subseteq \mathbb{R}^N$
    - into $S_1 \cup \ldots \cup S_k$
      - (using e.g. k-means)
  - Create a separate **local reduced model** for each cluster
  - Derive a basis $Q \in \mathbb{R}^{N \times m}, m \ll N$
    - to obtain low-dimensional **indicator** $z_i = Q^T x_i$ that describes state $x_i$
  - Learn a **classifier** $g: \mathcal{Z} \rightarrow \{1, \ldots, k\}$ to
    - map from low-dimensional indicator $z$ to model index
      - (using e.g. nearest neighbors)
  - Classify current state/indicator online and select model

→ Localized DEIM (**LDEIM**): Reduced models are tailored to local system behavior
Localized and adaptive reduced models

- Example: Reacting flow with one-step reaction
  \[ 2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O} \]

- Governed by convection-diffusion-reaction equation
  \[ \kappa \Delta y - \nu \nabla y + F(y, \mu) = 0 \quad \text{in } \Omega \]

- Exponential nonlinearity (Arrhenius-type source term)

POD-LDEIM: Combining 4 local models with machine-learning-based model management achieves accuracy improvement by up to two orders of magnitude compared to a single, global model
7. Conclusions and Challenges
Conclusions

• Many engineered systems of the future will have abundant sensor data
• Many systems of the future will leverage edge computing
  → an important role for reduced models, adaptive modeling, multifidelity modeling, uncertainty quantification
  → important to leverage the relative strengths of models and data
Challenges

Where do existing theories and methods fall short?

• Nonlinear parameter-varying systems
  → moving beyond linear subspaces
  → effective & efficient approximation of nonlinear terms
  → adaptive, data-driven methods

• Multiscale problems
  → effects of unresolved scales (closure)
  → ROMs across multiple scales

• Lack of rigorous error guarantees
  → especially for nonlinear problems

• Model inadequacy

• Intrusiveness of most existing model reduction methods has limited their impact


Useful References


Useful References


Useful References


Useful References


