Interpolation of parameterized reduced-order models

David Amsallem
Stanford University
February 06, 2014
Outline

- Parameterized unsteady PDE
- Database approach
- Interpolation of reduced-order bases
- Interpolation of reduced-order matrices
- Applications
Parametrized unsteady PDE

- Linear (linearized) PDE

\[
\frac{dw}{dt}(t) = A(\mu)w(t) + B(\mu)u(t)
\]
Example: aeroelasticity

- Example: aeroelastic analysis of full aircraft configuration

- Hundreds of flight conditions $\mu = (M_\infty, \alpha)$ to clear for flutter
Example: aeroelasticity

- Example

- Non-robustness with respect to the operating condition
Example: aeroelasticity

- Can we afford rebuilding the POD basis?

- Not suitable for real-time analysis
Database approach

- What should we interpolate?
Database approach

- What should we interpolate?
Model interpolation

- Reduced set of equations (after Galerkin projection)

\[
\frac{dq}{dt}(t) = V(\mu)^T A(\mu) V(\mu) q(t) + V(\mu)^T B u(t)
\]

- Full state reconstruction

\[
w(t) \approx V(\mu) q(t)
\]

- Model: \((V(\mu)^T A(\mu) V(\mu), V(\mu)^T B, V(\mu))\)

- Approach #1:
  1. interpolate \(V(\mu)\)
  2. evaluate \((A(\mu), B(\mu))\)
  3. form \((V(\mu)^T A(\mu) V(\mu), V(\mu)^T B)\)
Direct interpolation

- Natural idea: interpolate $V(\mu) \in \mathbb{R}^{N_w \times k}$ entry-by-entry
- Input:
  - target $\mu$
  - precomputed reduced bases $\{V(\mu_l)\}_{l=1}^m$
  - multi-variate interpolation operator $a(\mu) = \mathcal{I}(\mu; \{a(\mu_l), \mu_l\}_{l=1}^m)$
- Algorithm
  1: for $i = 1 : N_w$ do
  2:   for $j = 1 : k$ do
  3:     Compute $v_{ij}(\mu) = \mathcal{I}(\mu; \{v_{ij}(\mu_l), \mu_l\}_{l=1}^m)$
  4:   end for
  5: end for
  6: Form $V(\mu) = [v_{ij}(\mu)]$
Direct interpolation doesn’t work

- Example
  - \( N_w = 3, \; k = 2, \; p = 1 \)
  - for \( \mu_1 = 0, \; V(0) = [v_1, v_2] \)
  - for \( \mu_2 = 1, \; V(1) = [-v_1, v_2] \)
  - target parameter \( \mu = 0.5 \)
  - use linear interpolation
- Interpolation result:
  \[
  V(0.5) = 0.5(V(0) + V(1)) = [0.5(v_1 - v_1), 0.5(v_2 + v_2)] = [0, v_2]
  \]

- What went wrong?
- We haven’t interpolated the correct object
Subspace interpolation

- Projection-based model reduction

\[
\frac{dq}{dt}(t) = V(\mu)^T A(\mu) V(\mu) q(t) + V(\mu)^T B u(t)
\]

- Equivalent full state equation (multiply by \( V(\mu) \))

\[
\frac{dw}{dt}(t) = \Pi_{V(\mu), V(\mu)} A(\mu) w(t) + \Pi_{V(\mu), V(\mu)} B u(t)
\]

- An orthogonal projection is independent of the choice of reduced basis associated to the projection subspace

- Important quantity to interpolate: subspace
The Grassmann manifold

- A subspace $S$ is typically represented by a reduced basis
- The choice of reduced basis is not unique

$$S = \text{range}(V) = \text{range}(VQ), \forall Q \in \text{GL}(k)$$

- Matrix manifolds of interest
  - $G(k, N_w)$ (Grassmann manifold): set of subspaces of dimension $k$ in $\mathbb{R}^{N_w}$
  - $ST(k, N_w)$ (orthogonal Stiefel manifold): set of orthogonal reduced bases of dimension $k$ in $\mathbb{R}^{N_w}$

- Case of model reduction
  - $V(\mu) \in ST(k, N_w)$
  - $\text{range}(V(\mu)) \in G(k, N_w)$

- Interpolation on the Grassmann manifold (quotient manifold) using quantities belonging to the Stiefel manifold

$$G(k, N_w) = ST(k, N_w)/O(k)$$
The Grassmann manifold

- Matrix manifolds of interest
  - $G(k, N_w)$ (Grassmann manifold): set of subspaces of dimension $k$ in $\mathbb{R}^{N_w}$
  - $ST(k, N_w)$ (orthogonal Stiefel manifold): set of orthogonal reduced bases of dimension $k$ in $\mathbb{R}^{N_w}$
Interpolation on matrix manifolds

- First example: the circle

- Standard interpolation fails
- Idea: interpolate on a linear space $\Rightarrow$ a tangent space to the manifold
Interpolation on matrix manifolds

- **Input:**
  - precomputed matrices \( \{A(\mu_l)\}_{l=1}^m \)
  - multi-variate interpolation operator \( a(\mu) = \mathcal{I}(\mu; \{a(\mu_l), \mu_l\}_{l=1}^m) \)
  - map \( m_A \) from the manifold \( \mathcal{M} \) to the tangent space of \( \mathcal{M} \) at \( m_A \)
  - inverse map \( m_A^{-1} \) from the tangent space to \( \mathcal{M} \) at \( m_A \) to the manifold \( \mathcal{M} \)

1: **for** \( l = 1 : m \) **do**
2: \hspace{1em} Compute \( \Gamma(\mu_l) = m_A(A(\mu_l)) \)
3: **end for**
4: **for** \( i = 1 : N_w \) **do**
5: \hspace{1em} **for** \( j = 1 : k \) **do**
6: \hspace{2em} Compute \( \Gamma_{ij}(\mu_l) = \mathcal{I}(\mu; \{\Gamma_{ij}(\mu_l), \mu_l\}_{l=1}^m) \)
7: \hspace{1em} **end for**
8: **end for**
9: Form \( \Gamma(\mu) = [\Gamma_{ij}(\mu)] \) and compute \( A(\mu) = m_A^{-1}(\Gamma(\mu)) \)

- **Requirement:** the interpolation operator \( \mathcal{I} \) preserves the tangent space.

  **for instance:** \( a(\mu) = \mathcal{I}(\mu; \{a(\mu_l), \mu_l\}_{l=1}^m) = \sum_{l=1}^m \theta_l(\mu)a(\mu_l) \)
Interpolation on matrix manifolds

- How do we find $m_A$ and its inverse $m_A^{-1}$
- Idea: use concepts from differential geometry
- Geodesics
  - generalize straight lines on manifolds
  - uniquely defined given a point $x$ of the manifold and a tangent vector $\xi$ at this point

$$\gamma(t; x, \xi) : [0, 1] \rightarrow \mathcal{M}$$

$$\gamma(0; x, \xi) = x, \quad \dot{\gamma}(0, x, \xi) = \xi$$

- Exponential map

$$\text{Exp}_x : T_x \mathcal{M} \rightarrow \mathcal{M} \quad \xi \longmapsto \gamma(1; x, \xi)$$

- Logarithm map (defined in a neighborhood $\mathcal{U}_x$ of $x$)

$$\text{Log}_x : \mathcal{U}_x \subset \mathcal{M} \rightarrow T_x \mathcal{M} \quad y \longmapsto \text{Exp}_x^{-1}(y)$$
Interpolation on matrix manifolds

- Application to interpolation of points on a circle
Interpolation on matrix manifolds

Case of the Grassmann manifold

- Logarithmic map
  1. Compute the thin SVD
     \[(I - V_0 V_0^T) V_1 (V_0^T V_1)^{-1} = U \Sigma Z^T \]
  2. Compute
     \[\Gamma = U \tan^{-1}(\Sigma) Z^T\]
  3. \(\text{Log}_{S_0}(S_1) \leftrightarrow \Gamma\)

- Exponential map of \(\xi \in T_{S_0} G \leftrightarrow \Gamma\)
  1. Compute the thin SVD
     \[\Gamma = U \Sigma Z^T\]
  2. Compute
     \[V = V_0 Z \cos \Sigma + U \sin \Sigma\]
  3. \(\text{Exp}_{S_0}(\xi) = \text{range}(V)\)

- Note: the trigonometric operators only apply to the diagonal elements of the matrices
Interpolation on matrix manifolds

- Case of the Grassmann manifold
Application to aeroelasticity

- Aeroelastic behavior of the F-16

![Graph showing lift vs. time for different ROB methods](image)

R0M (90) built using the

- Directly computed ROB
- Precomputed ROB #1
- Precomputed ROB #2
- Precomputed ROB #3
- Precomputed ROB #4
- Interpolated ROB (4-point Grassmann manifold method)
Application to aeroelasticity

- Aeroelastic behavior of a commercial aircraft

![Airbus AMP model](image)

**Unsteady pressure distribution**
- Upper surface
- Lower surface

**Interpolated ROB (Grassmann)**

![Pressure and Damping Graph](image)

- Frequency vs Pressure
- Damping vs Pressure

David Amsallem

Interpolation of parameterized reduced-order models

ROM (46) Vetrano et al., ASD Journal 2011
Application to aeroelasticity

- Aeroelastic behavior of the F-16

- Dominant cost: computation of $A(\mu)$ and $B(\mu)$ for a new value of $\mu$
- Approach #2: interpolate $(V(\mu)^T A(\mu) V(\mu), V(\mu)^T B(\mu))$
Example

- Simple example: mass-spring system with two degrees of freedom
- $\mu = k_1 - 0.1$

$$k_1 \quad \tilde{k} \quad k_2$$

$${m_1} \quad \quad \quad \quad \quad {m_2}$$

$$x_1 \quad \quad \quad \quad \quad x_2$$

- $V(\mu)^T A(\mu) V(\mu) = \Lambda(\mu)$
Interpolation on a matrix manifold

- $\Lambda(\mu)$ belongs to the manifold of symmetric positive definite matrices (diagonal)
- Interpolate on the manifold using $(\Lambda(0), \Lambda(2.9))$
Example: Mode veering and mode crossing

- The issue is the mode veering: the coordinates of the reduced matrices are not consistent.

![Mode veering](image)

- There would be an issue also with mode crossing (the eigenfrequencies are ordered increasingly in $\Lambda$).
Consistent interpolation on matrix manifolds

- **Solution**: pre-process the reduced matrices (Step A)
- Consistency enforced by the solution of an orthogonal Procrustes problem

\[
\min_{Q_i} \quad \|V_i Q_i - V_{i0}\|_F, \quad \forall i = 1, \ldots, m
\]

- **Analytical solution using the SVD**
  1. Compute \( P_{i,i0} = V_i^T V_{i0} \)
  2. Compute the SVD \( P_{i,i0} = U_{i,i0} \Sigma_{i,i0} Z_{i,i0}^T \)
  3. Let \( Q_i = U_{i,i0} Z_{i,i0}^T \)

- Can be processed online or offline
- **Step B**: interpolation on a matrix manifold
Application 1

![Graph](image_url)

- **Pre-computed Points**
- **Exact Eigenvalues**
- **Eigenvalues Obtained by Step B Only**
- **Eigenvalues Obtained by Steps A and B**
Application 2

- More challenging example: wing with tank and sloshing effect
- The hydro-elastic effects affect the eigen-frequencies and eigen-modes of the structure
- The parameter $\mu$ defines the level of fuel in the tank $0 \leq \mu \leq 100\%$
Application 2

David Amsallem

Interpolation of parameterized reduced-order models
Application 2

Interpolation of parameterized reduced-order models
The MAC between two eigenmodes $\phi$ and $\psi$ is

$$\text{MAC}(\phi, \psi) = \frac{|\phi^T \psi|^2}{(\phi^T \phi)(\psi^T \psi)}$$

- When $\phi$ and $\psi$ are normalized $\text{MAC}(\phi, \psi) = |\phi^T \psi|^2$
- $P_{i,i_0}$ is the matrix of square roots of the MACs between the modes contained in $V(\mu_i)$ and those contained in $V(\mu_{i_0})$.
- This is the Modal Assurance Criterion Square Root (MACSR)
Application 3

- Aeroelastic study of the wing-tank system
- 2 parameters: fill level and free-stream Mach number $M_\infty$
- Database approach
Application 3

Interpolation of parameterized reduced-order models
Application 3

- Effect of Step A

![Graph showing the effect of Step A](image)

David Amsallem

Interpolation of parameterized reduced-order models
Application 3

- Bifurcation detection

David Amsallem

Interpolation of parameterized reduced-order models
Mobile computing using a database of ROMs
References