Interpolatory tensorial reduced order models for parametric dynamical systems

Alexander V. Mamonov[1] and Maxim A. Olshanskii[1]

[1]University of Houston

Support: ONR N00014-21-1-2370

Motivation and overview

- Projection-based model reduction for parametric dynamical systems
- Classical POD approach: dump all snapshots in one big matrix, project the dynamical system onto its n left singular vectors
- **Problem**: loss of information about dependency of snapshots on parameters
- **Solution**: use snapshot **tensor** instead of a matrix; use **low-rank tensor decompositions** instead of SVD

Parametric dynamical system and POD-ROM

- **Dynamical system** for $\mathbf{u} : [0, T) \to \mathbb{R}^M$ solving

$$\mathbf{u}_t = F(t, \mathbf{u}, \boldsymbol{\alpha}), \quad t \in (0, T), \quad \mathbf{u}|_{t=0} = \mathbf{u}_0 \tag{1}$$

- Parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_D)^T$ from **parameter domain** $\mathcal{A} \subset \mathbb{R}^D$
- **Snapshots** $\phi_k(\boldsymbol{\alpha}) = \mathbf{u}(t_k, \boldsymbol{\alpha}) \in \mathbb{R}^M$, $k = 1, \dots, N$
- For a **fixed** $\boldsymbol{\alpha} \in \mathcal{A}$ assemble **snapshot matrix**

$$\Phi_{\text{pod}}(\boldsymbol{\alpha}) = [\phi_1(\boldsymbol{\alpha}), \dots, \phi_N(\boldsymbol{\alpha})] \in \mathbb{R}^{M \times N}$$

- Choose **ROM dimension** $n \ll M$
- **POD-ROM: project** (1) onto the **first** n **left singular vectors** of $\Phi_{\text{pod}}(\boldsymbol{\alpha})$

POD-ROM for parametric systems

- Sample parameter domain to get the **sampling set**

$$\widehat{\mathcal{A}} = \{\widehat{\alpha}_1, \ldots, \widehat{\alpha}_K\}$$

- Assemble a **huge** matrix of **all** snapshots

$$\Phi_{\widehat{\mathcal{A}}} = [\phi_1(\widehat{\alpha}_1), \ldots, \phi_N(\widehat{\alpha}_1), \ldots, \phi_1(\widehat{\alpha}_K), \ldots, \phi_N(\widehat{\alpha}_K)] \in \mathbb{R}^{M \times KN}$$

- Project dynamical system onto first n left singular vectors of $\Phi_{\widehat{\mathcal{A}}}$
- **Major drawbacks:**
  1. **Very expensive** in both storage and computation
  2. Often **lacks robustness** away from parameter samples $\widehat{\mathcal{A}}$
  3. **Disregards the tensor product structure** of parameter space

Interpolatory tensorial ROM

- Our solution: **interpolatory tensorial ROM (TROM)**
- Combination of two ideas
  1. **Offline stage:** use low-rank **tensor decompositions** to **compress** the snapshot tensor (for all parameters in $\widehat{\mathcal{A}}$) and account for tensor-product structure of parameter space
  2. **Online stage:** for a specific **out-of-sample** $\alpha \in \mathcal{A} \setminus \widehat{\mathcal{A}}$ compute the **reduced basis** using **interpolation**
- Assume first $\mathcal{A}$ being a D-dimensional box sampled on a **Cartesian grid** $\widehat{\mathcal{A}}$ with **nodes**

$$\{\widehat{\alpha}_i^j\}_{i=1,\dots,D,\ j=1,\dots,n_i},$$

so $K = n_1 \times n_2 \times \ldots \times n_D$

- Define **snapshot tensor** $\mathbf{\Phi} \in \mathbb{R}^{M \times n_1 \times \cdots \times n_D \times N}$ with entries

$$(\mathbf{\Phi})_{:,i_1,\dots,i_D,k} = \phi_k(\widehat{\alpha}_1^{i_1}, \dots, \widehat{\alpha}_D^{i_D})$$

Offline stage: tensor compression

- We need a **low-rank tensor** approximation $\widetilde{\mathbf{\Phi}}$ to snapshot tensor

$$\left\| \widetilde{\mathbf{\Phi}} - \mathbf{\Phi} \right\|_F \leq \widetilde{\varepsilon} \left\| \mathbf{\Phi} \right\|_F$$

- **Three possibilities:**
  1. **Canonical polyadic (CP)** decomposition

$$\mathbf{\Phi} \approx \widetilde{\mathbf{\Phi}} = \sum_{r=1}^{R} \mathbf{u}^r \circ \boldsymbol{\sigma}_1^r \circ \cdots \circ \boldsymbol{\sigma}_D^r \circ \mathbf{v}^r$$

  2. **High order SVD (HOSVD)** Tucker form

$$\mathbf{\Phi} \approx \widetilde{\mathbf{\Phi}} = \sum_{j=1}^{\widetilde{M}} \sum_{q_1=1}^{\widetilde{n}_1} \cdots \sum_{q_D=1}^{\widetilde{n}_D} \sum_{k=1}^{\widetilde{N}} (\mathbf{C})_{j,q_1,\ldots,q_D,k} \mathbf{u}^j \circ \boldsymbol{\sigma}_1^{q_1} \circ \cdots \circ \boldsymbol{\sigma}_D^{q_D} \circ \mathbf{v}^k$$

  3. **Tensor train (TT)** decomposition

$$\mathbf{\Phi} \approx \widetilde{\mathbf{\Phi}} = \sum_{j_1=1}^{\widetilde{r}_1} \cdots \sum_{j_{D+1}=1}^{\widetilde{r}_{D+1}} \mathbf{u}^{j_1} \circ \boldsymbol{\sigma}_1^{j_1,j_2} \circ \cdots \circ \boldsymbol{\sigma}_D^{j_D,j_{D+1}} \circ \mathbf{v}^{j_{D+1}}$$

Online stage: interpolation

- Define **interpolation process** via vectors $\mathbf{e}^i(\boldsymbol{\alpha}) \in \mathbb{R}^{n_i}$ such that for smooth f we have

$$f(\alpha_i) \approx \sum_{j=1}^{n_i} [\mathbf{e}^i(\boldsymbol{\alpha})]_j f(\widehat{\alpha}_i^j), \quad i = 1, \dots, D,$$

  e.g., use **Lagrange interpolation** on p nearest grid nodes

- Using k-mode tensor vector product $\times_k$, define **extraction**

$$\Phi_e(\boldsymbol{\alpha}) = \boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \cdots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha}) \in \mathbb{R}^{M \times N},$$

  which extracts from whole snapshot tensor $\boldsymbol{\Phi}$ a **matrix** of **interpolated** snapshots most relevant to $\boldsymbol{\alpha}$

- **Remark:** if $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$ then $\Phi_e(\widehat{\boldsymbol{\alpha}}) = \Phi_{\text{pod}}(\widehat{\boldsymbol{\alpha}})$

Online stage: interpolation and reduced basis

- **Online stage** is performed for a **specific** $\alpha \in \mathcal{A}$
- Once we have $\widetilde{\boldsymbol{\Phi}} \approx \boldsymbol{\Phi}$ from **offline stage**, use extraction

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \widetilde{\boldsymbol{\Phi}} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \cdots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha}) \in \mathbb{R}^{M \times N}$$

- Compute **thin** SVD of **low-rank**

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \mathrm{Z}\widetilde{\Sigma}\mathrm{Y}^T$$

- If $\mathrm{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots]$, let the orthonormal **reduced basis** be

$$\mathcal{Z}_n(\boldsymbol{\alpha}) = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$$

- **Interpolatory TROM** is obtained by **projecting** the dynamical system onto

$$\text{span } \mathcal{Z}_n(\boldsymbol{\alpha})$$

Computational efficiency

- In practice, there is no need to compute SVD of $\widetilde{\Phi}_e(\alpha)$ or even to assemble $\widetilde{\Phi}_e(\alpha)$!
- Instead, all necessary calculations can be performed for a **small core matrix**
- Also, no need to form $\mathcal{Z}_n(\alpha) = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ explicitly
- Can write $\mathbf{z}_j = \mathrm{U}\boldsymbol{\beta}_j(\alpha)$, where U is **computed and stored offline** and **parameter-specific coefficients** $\boldsymbol{\beta}_j(\alpha)$ are computed **online**
- Dynamical system can be pre-projected onto U at the **offline** stage

Particular case: HOSVD-TROM

- **Offline stage:** compute tensor approximation

$$\boldsymbol{\Phi} \approx \widetilde{\boldsymbol{\Phi}} = \sum_{j=1}^{\widetilde{M}} \sum_{q_1=1}^{\widetilde{n}_1} \cdots \sum_{q_D=1}^{\widetilde{n}_D} \sum_{k=1}^{\widetilde{N}} (\mathbf{C})_{j,q_1,\dots,q_D,k} \mathbf{u}^j \circ \boldsymbol{\sigma}_1^{q_1} \circ \cdots \circ \boldsymbol{\sigma}_D^{q_D} \circ \mathbf{v}^k,$$

assemble matrices

$$\mathrm{U} = [\mathbf{u}^1, \dots, \mathbf{u}^{\widetilde{M}}] \in \mathbb{R}^{M \times \widetilde{M}}, \ \mathrm{S}_i = [\boldsymbol{\sigma}_i^1, \dots, \boldsymbol{\sigma}_i^{\widetilde{n}_i}] \in \mathbb{R}^{n_i \times \widetilde{n}_i}, \ i = 1, \dots, D$$

- **Online stage:** form **core matrix**

$$\mathrm{C}_e(\alpha) = \mathbf{C} \times_2 \left(\mathrm{S}_1 \mathbf{e}^1(\alpha) \right) \times_3 \left(\mathrm{S}_2 \mathbf{e}^2(\alpha) \right) \cdots \times_{D+1} \left(\mathrm{S}_D \mathbf{e}^D(\alpha) \right) \in \mathbb{R}^{\widetilde{M} \times \widetilde{N}}$$

Reduced basis coefficients $\beta_j(\alpha)$ are the first n left singular vectors of $\mathrm{C}_e(\alpha)$

Prediction analysis

- We measure **prediction power** of reduced basis $\mathcal{Z}_n(\alpha) = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ by how well $\mathbf{u}(t_k, \alpha)$ can be represented in it
- The following bound holds

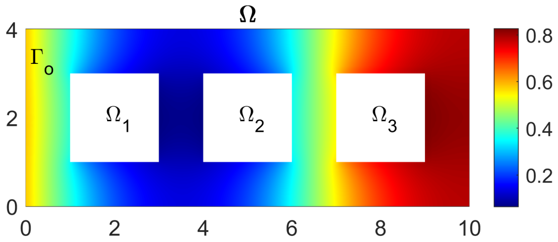
$$
\frac{1}{3NM} \sum_{k=1}^{N} \left\| \mathbf{u}(t_k, \alpha) - \sum_{j=1}^{n} \langle \mathbf{u}(t_k, \alpha), \mathbf{z}_j \rangle \mathbf{z}_j \right\|_{\ell^2}^2
$$

$$
\leq \frac{1}{NM} \left((C_e)^{2D} \widetilde{\varepsilon}^2 \, \|\boldsymbol{\Phi}\|_F + \sum_{i=n+1}^{N} \widetilde{\sigma}_i^2 \right) + C_a C_{\mathbf{u}} \max \left\{ (C_e)^{2(D-1)}, 1 \right\} \delta^{2p},
$$

where

- C_a, C_e depend on interpolation process
- $C_{\mathbf{u}} = \|\mathbf{u}\|_{C(0,T;C^p(\mathcal{A}))}$
- $\widetilde{\sigma}_i$ are the singular values of $\widetilde{\Phi}_e(\alpha)$
- δ is the maximum grid step of $\widehat{\mathcal{A}}$

Numerical experiments: heat equation, $D = 4$



- Heat equation

$$w_t = \Delta w, \quad t \in (0, T]$$

  for $w(t, \mathbf{x}, \alpha)$ in a domain Ω with three holes

- Parameters ($D = 4$) enter the **boundary conditions**

$$\left(\mathbf{n} \cdot \nabla w + \alpha_1 (w - 1)\right)\big|_{\Gamma_o} = 0,$$

$$\left(\mathbf{n} \cdot \nabla w + \frac{1}{2} w\right)\bigg|_{\partial\Omega_i} = \frac{1}{2}\alpha_{i+1}, \quad i = 1, 2, 3,$$

  the rest of the boundary is **insulated**

Heat equation: out-of-sample TROM performance

- Given true $w(t_k, \mathbf{x}, \boldsymbol{\alpha})$ and ROM $\widetilde{w}(t_k, \mathbf{x}, \boldsymbol{\alpha})$ solutions, define **relative ROM solution error**

$$R_{\mathrm{X}}(\boldsymbol{\alpha}) = \frac{\max\limits_{k=1,\ldots,N} \left\| \widetilde{w}(t_k, \mathbf{x}, \boldsymbol{\alpha}) - w(t_k, \mathbf{x}, \boldsymbol{\alpha}) \right\|_{L^2(\Omega)}}{\max\limits_{k=1,\ldots,N} \left\| w(t_k, \mathbf{x}, \boldsymbol{\alpha}) \right\|_{L^2(\Omega)}},$$

for X$\in$ {CP, HOSVD, TT, POD}, and **relative gain**

$$G_{\mathrm{X}} = R_{\mathrm{POD}}(\boldsymbol{\alpha}) / R_{\mathrm{X}}(\boldsymbol{\alpha}),$$

- Report **mean** of G_{X} over 200 random realizations $\boldsymbol{\alpha}^{(r)} \in \mathcal{A} \setminus \widehat{\mathcal{A}}$

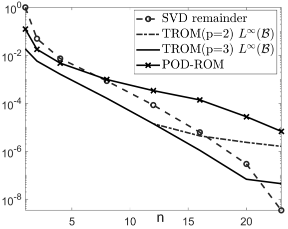
$\widetilde{\varepsilon} = 10^{-5}$	mean G_{X}		
K	CP	HOSVD	TT
$135 = 5 \times 3^3$	**24.76**	**25.08**	**25.08**
$1000 = 8 \times 5^3$	**35.21**	**35.52**	**35.51**
$3430 = 10 \times 7^3$	**37.80**	**38.80**	**38.80**

$\widetilde{\varepsilon} = 10^{-7}$	mean G_{X}		
n	CP	HOSVD	TT
10	**38.66**	**38.80**	**38.80**
20	**49.80**	**155.65**	**154.03**

Heat equation: out-of-sample predictive power

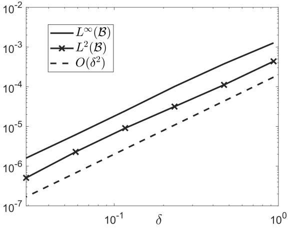
- We use
$$E_{L^\infty(\mathcal{A})} = \sup_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\frac{1}{MN} \left\| (\mathrm{I} - \mathrm{Z}_n \mathrm{Z}_n^T) \Phi_e(\boldsymbol{\alpha}) \right\|_F^2 \right)^{1/2}$$
and its $L^2(\mathcal{A})$ analogue to check **predictive power** of **HOSVD-TROM** for out-of-sample $\boldsymbol{\alpha}$, where $\mathrm{Z}_n = [\mathbf{z}_1, \dots, \mathbf{z}_n]$
- The study is for $D = 2$ parameters: $\alpha_1 = \alpha_2 = \alpha_3$ and α_4

TROM and POD prediction error vs n



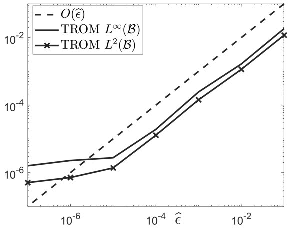
$\widetilde{\varepsilon} = 10^{-7}$

TROM prediction error vs δ



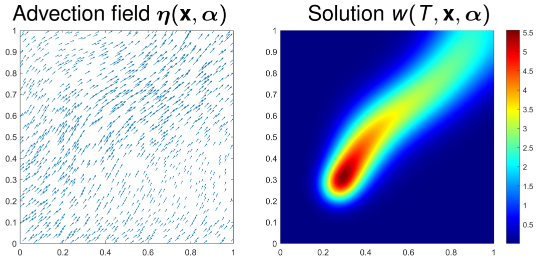
$\widetilde{\varepsilon} = 10^{-7}$, $p = 2$

TROM prediction error vs $\widehat{\varepsilon}$



$p = 2$

Numerical experiments: advection-diffusion, $D = 9$



- Advection-diffusion equation for $w(t, \mathbf{x}, \boldsymbol{\alpha})$:

$$w_t = \nu \Delta w - \boldsymbol{\eta}(\mathbf{x}, \boldsymbol{\alpha}) \cdot \nabla w + f(\mathbf{x}), \quad t \in (0, T]$$

  with Gaussian source $f(\mathbf{x})$ in a unit square with insulated boundary
- Divergence-free **advection field** parametrized with $D = 9$ parameters

$$\boldsymbol{\eta}(\mathbf{x}, \boldsymbol{\alpha}) = \begin{pmatrix} \eta_1(\mathbf{x}, \boldsymbol{\alpha}) \\ \eta_2(\mathbf{x}, \boldsymbol{\alpha}) \end{pmatrix} = \begin{pmatrix} \cos \alpha_9 \\ \sin \alpha_9 \end{pmatrix} + \frac{1}{\pi} \begin{pmatrix} \partial_{x_2} h(\mathbf{x}, \boldsymbol{\alpha}) \\ -\partial_{x_1} h(\mathbf{x}, \boldsymbol{\alpha}) \end{pmatrix}, \text{ with}$$

$$\begin{aligned} h(\mathbf{x}, \boldsymbol{\alpha}) = \quad & \alpha_1 \cos(\pi x_1) + \alpha_2 \cos(\pi x_2) + \alpha_3 \cos(\pi x_1) \cos(\pi x_2) \\ & + \alpha_4 \cos(2\pi x_1) + \alpha_5 \cos(2\pi x_2) + \alpha_6 \cos(2\pi x_1) \cos(\pi x_2) \\ & + \alpha_7 \cos(\pi x_1) \cos(2\pi x_2) + \alpha_8 \cos(2\pi x_1) \cos(2\pi x_2). \end{aligned}$$

Advection-diffusion equation: TROM performance

- Numerical results: diffusion coefficient $\nu = 0.01$, $M = 4797$, $K = 20 \times 2^8 = 5120$
- Report mean of G_{X} for HOSVD- and TT-TROM, CP decomposition is too memory intensive
- Good results for $\widetilde{\varepsilon} = 10^{-3}$, $n = 10$, no need to use more expensive options

$\widetilde{\varepsilon} = 10^{-3}$	n	5	8	10
mean G_{X}	HOSVD	**6.95**	**22.56**	**32.66**
	TT	**6.95**	**22.54**	**31.83**

$\widetilde{\varepsilon} = 10^{-5}$	n	5	10	15
mean G_{X}	HOSVD	**6.95**	**33.34**	**19.45**
	TT	**6.95**	**33.34**	**19.45**

$\widetilde{\varepsilon} = 10^{-7}$	n	5	10	15
mean G_{X}	HOSVD	**6.95**	**33.34**	**19.45**
	TT	**6.95**	**33.34**	**19.45**

Conclusions and future work

- Framework for model reduction for parametric dynamical systems based on low-rank tensor approximation of snapshots (**offline**); tensor decompositions provide a universal basis that retains information about solution variation with respect to parameters
- Information from compressed tensor representation is used to compute the (coefficients of) ROM basis for any incoming parameter, including out-of-sample (**online**)
- Prediction power analysis

Future work

- Model reduction for non-linear dynamical systems, DEIM in tensor framework, etc.

Reference

Interpolatory tensorial reduced order models for parametric dynamical systems. A.V. Mamonov, M.A. Olshanskii, Preprint: **arXiv:2211.00649 [math.NA]**