Tensorial parametric model order reduction of nonlinear dynamical systems

Alexander V. Mamonov and Maxim A. Olshanskii

University of Houston

Support: ONR N00014-21-1-2370, NSF DMS-2309197

Motivation and overview

- Projection-based **model order reduction** for **nonlinear parametric** dynamical systems
- **Conventional approach:** assemble the snapshots of the linear and nonlinear terms into two big matrices, use SVD to compute projection subspaces + **discrete empirical interpolation method** (DEIM) for the nonlinear term
- **Problem**: loss of information about dependency of snapshots on parameters
- **Solution**: use snapshot **tensors** instead of matrices; use **low-rank tensor decompositions** instead of SVD + DEIM to treat nonlinearity

Parametric nonlinear dynamical system

- Dynamical system for $\mathbf{u} = \mathbf{u}(t; \boldsymbol{\alpha}) : [0, T) \to \mathbb{R}^M$

$$\mathbf{u}_t = \mathrm{A}_{\boldsymbol{\alpha}}\mathbf{u} + \mathbf{f}_{\boldsymbol{\alpha}}(t, \mathbf{u}), \quad t \in (0, T), \quad \mathbf{u}|_{t=0} = \mathbf{u}_0,$$

- Parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)^T$ are from the parameter domain $\mathcal{A} \subset \mathbb{R}^D$
- Parametric system with
  - **Linear term** defined by matrix $\mathrm{A}_{\boldsymbol{\alpha}} \in \mathbb{R}^{M \times M}$,
  - **Nonlinear term** defined by flow field $\mathbf{f}_{\boldsymbol{\alpha}} : (0, T) \times \mathbb{R}^M \to \mathbb{R}^M$
- Sample parameter domain to get the **training set**

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

- Define **solution snapshots**

$$\phi_j(\widehat{\boldsymbol{\alpha}}_k) = \mathbf{u}(t_j, \widehat{\boldsymbol{\alpha}}_k) \in \mathbb{R}^M, \;\; j = 1, \ldots, N, \;\; k = 1, \ldots, K,$$

and **nonlinear snapshots**

$$\psi_j(\widehat{\boldsymbol{\alpha}}_k) = \mathbf{f}_{\widehat{\boldsymbol{\alpha}}_k}(t_j, \mathbf{u}(t_j, \widehat{\boldsymbol{\alpha}}_k)) \in \mathbb{R}^M, \;\; j = 1, \ldots, N, \;\; k = 1, \ldots, K.$$

Conventional POD-ROM with DEIM

- Assemble two **snapshot matrices**:

$$\Phi_{\rm pod} = [\phi_1(\widehat{\boldsymbol\alpha}_1), \dots, \phi_N(\widehat{\boldsymbol\alpha}_1), \dots, \phi_1(\widehat{\boldsymbol\alpha}_K), \dots, \phi_N(\widehat{\boldsymbol\alpha}_K)] \in \mathbb{R}^{M \times NK}$$
$$\Psi_{\rm pod} = [\psi_1(\widehat{\boldsymbol\alpha}_1), \dots, \psi_N(\widehat{\boldsymbol\alpha}_1), \dots, \psi_1(\widehat{\boldsymbol\alpha}_K), \dots, \psi_N(\widehat{\boldsymbol\alpha}_K)] \in \mathbb{R}^{M \times NK}$$

- Choose ROM dimension $n \ll M$ and take U_n and Y_n to be the first n **left singular vectors** of $\Phi_{\rm pod}$ and $\Psi_{\rm pod}$, respectively
- POD-ROM solution $\mathbf{u}^{\rm pod} = \mathrm{U}_n\boldsymbol\beta$ solves the projected system

$$\boldsymbol\beta_t = \underbrace{\mathrm{U}_n^T \mathrm{A}_{\boldsymbol\alpha} \mathrm{U}_n}_{\text{can precompute}} \boldsymbol\beta + \mathrm{U}_n^T \underbrace{\mathbf{f}_{\boldsymbol\alpha}(t, \mathrm{U}_n\boldsymbol\beta)}_{\text{expensive}}, \quad t \in (0, T), \quad \boldsymbol\beta|_{t=0} = \mathrm{U}_n^T \mathbf{u}_0$$

- Reduce computational expense for the nonlinear term: **discrete empirical interpolation method (DEIM)**

DEIM for nonlinear model reduction

- DEIM: a greedy method for choosing a subset of entries of $\mathbf{f}_\alpha$ to reduce the computational cost of evaluating the projected nonlinear term:

$$\mathbf{f}_\alpha(t, \mathbf{u}) \approx \mathrm{Y}_n(\mathrm{P}^T \mathrm{Y}_n)^{-1} \mathrm{P}^T \mathbf{f}_\alpha(t, \mathbf{u}),$$

  where $\mathrm{P} \in \mathbb{R}^{M \times n}$ is the selection matrix, a subset of columns of $M \times M$ identity matrix

- Combining POD with DEIM yields the **POD-DEIM ROM:**

$$\boldsymbol{\beta}_t = \underbrace{\mathrm{U}_n^T \mathrm{A}_\alpha \mathrm{U}_n}_{\text{can precompute}} \boldsymbol{\beta} + \underbrace{(\mathrm{U}_n^T \mathrm{Y}_n)(\mathrm{P}^T \mathrm{Y}_n)^{-1}}_{\text{small, cheap}} \underbrace{\mathrm{P}^T \mathbf{f}_\alpha(t, \mathrm{U}_n \boldsymbol{\beta})}_{\text{cheap}}, \quad t \in (0, T),$$

Approaches to model reduction for parametric systems

- Drawbacks of the conventional POD-DEIM for parametric systems:
  1. Reduced bases U_n and Y_n capture cumulative rather than localized information about the dependence of $\phi_j(\alpha)$ and $\psi_j(\alpha)$ on α, hence POD-DEIM **lacks robustness** for parameter values **outside of the training set** $\widehat{\mathcal{A}}$
  2. **Disregards the tensor product structure** of the parameter space
- Our approach: **interpolatory tensorial ROM (TROM)**
- **Two-stage** process
  1. **Offline stage:** use low-rank **tensor decompositions** to **compress** both snapshot tensors and account for tensor-product structure of the parameter space; then perform **global** DEIM
  2. **Online stage:** for a specific **out-of-sample** $\alpha \in \mathcal{A} \setminus \widehat{\mathcal{A}}$ compute the **reduced basis** using **interpolation**; then perform **local** DEIM or least squares fitting

Interpolatory tensorial ROM

- Assume that the parameter space $\mathcal{A}$ is a D-dimensional box sampled on a **Cartesian grid** $\widehat{\mathcal{A}}$ with **nodes**
$$\{\widehat{\alpha}_i^j\}_{i=1,\dots,D,\; j=1,\dots,K_i},$$
so $K = |\widehat{\mathcal{A}}| = K_1 \times K_2 \times \ldots \times K_D$
- Instead of snapshot matrices, assemble two **snapshot tensors** $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}^{M \times K_1 \times \cdots \times K_D \times N}$ with entries
$$(\mathbf{\Phi})_{:,i_1,\dots,i_D,j} = \phi_j(\widehat{\alpha}_1^{i_1}, \dots, \widehat{\alpha}_D^{i_D}),$$
$$(\mathbf{\Psi})_{:,i_1,\dots,i_D,j} = \psi_j(\widehat{\alpha}_1^{i_1}, \dots, \widehat{\alpha}_D^{i_D}),$$
with $j = 1, \dots, N$
- For the offline stage of TROM we need **low-rank tensor** approximations to both snapshot tensors
$$\left\|\widetilde{\mathbf{\Theta}} - \mathbf{\Theta}\right\|_F \le \varepsilon \left\|\mathbf{\Theta}\right\|_F, \quad \mathbf{\Theta} \in \{\mathbf{\Phi}, \mathbf{\Psi}\}$$

Low-rank tensor decompositions

Tensor rank definition is **not unique** and neither are the low-rank decomposition formats. Popular choices include:

1. **Canonical polyadic (CP)** decomposition and rank R:

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

2. **High order SVD (HOSVD)** and Tucker ranks $\widetilde{M}, \widetilde{K}_1, \ldots, \widetilde{K}_D, \widetilde{N}$:

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

3. **Tensor train (TT)** decomposition and ranks $\widetilde{r}_1, \ldots, \widetilde{r}_{D+1}$:

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

TROM offline stage

- For the TT variant of TROM, compute the decompositions for a chosen accuracy ε:

$$\mathbf{\Theta} \approx \widetilde{\mathbf{\Theta}} = \sum_{j_1=1}^{\widetilde{R}_1^{\Theta}} \cdots \sum_{j_{D+1}=1}^{\widetilde{R}_{D+1}^{\Theta}} \mathbf{u}_{\Theta}^{j_1} \circ \boldsymbol{\sigma}_{1,\Theta}^{j_1,j_2} \circ \cdots \circ \boldsymbol{\sigma}_{D,\Theta}^{j_D,j_{D+1}} \circ \mathbf{v}_{\Theta}^{j_{D+1}}, \quad \mathbf{\Theta} \in \{\mathbf{\Phi}, \mathbf{\Psi}\}$$

- For the online stage we need the **universal bases**

$$\mathrm{U} = \left[\mathbf{u}_{\Phi}^{1}, \ldots, \mathbf{u}_{\Phi}^{\widetilde{R}_1^{\Phi}}\right] \in \mathbb{R}^{M \times \widetilde{R}_1^{\Phi}}, \quad \mathrm{Y} = \left[\mathbf{u}_{\Psi}^{1}, \ldots, \mathbf{u}_{\Psi}^{\widetilde{R}_1^{\Psi}}\right] \in \mathbb{R}^{M \times \widetilde{R}_1^{\Psi}},$$

  and also

$$\left[\mathbf{S}_i^{\Theta}\right]_{jkq} = \left[\boldsymbol{\sigma}_{i,\Theta}^{jq}\right]_k, \;\; j = 1, \ldots, \widetilde{R}_i^{\Theta}, \;\; k = 1, \ldots, K_i, \;\; q = 1, \ldots, \widetilde{R}_{i+1}^{\Theta},$$

$$\mathrm{W}^{\Theta} = \mathrm{diag}\left(\|\mathbf{v}_{\Theta}^{1}\|, \ldots, \|\mathbf{v}_{\Theta}^{\widetilde{R}_{D+1}^{\Theta}}\|\right)$$

- Apply DEIM to the columns of Y to obtain the selection matrix P

TROM online stage

- Online stage is **parameter-specific**, it takes $\boldsymbol{\alpha} \in \mathcal{A} \cap \widehat{\mathcal{A}}$ as an input
- Compute two **core matrices**

$$\mathrm{C}^{\Theta}(\boldsymbol{\alpha}) = \prod_{i=1}^{D} \left(\mathbf{S}_i^{\Theta} \times_2 \mathbf{e}^i(\boldsymbol{\alpha}) \right),$$

where $\mathbf{e}^i(\boldsymbol{\alpha})$, $i = 1, \ldots, D$, are (Lagrange) interpolants; and SVDs

$$\mathrm{C}^{\Theta}(\boldsymbol{\alpha})\mathrm{W}^{\Theta} = \mathrm{X}^{\Theta}\Sigma^{\Theta}(\mathrm{Z}^{\Theta})^T$$

- Choose the **ROM dimensions** $n_\Phi, n_\Psi \ll M$ and take the **local bases** U_n and Y_n to be the first n_Θ columns of X^{Θ} for $\Theta \in \{\Phi, \Psi\}$, respectively
- Can avoid local DEIM by using the least squares fitting

Final TROM form

- Recall the full order model (FOM):

$$\mathbf{u}_t = \mathrm{A}_\alpha \mathbf{u} + \mathbf{f}_\alpha(t, \mathbf{u})$$

- TROM takes the form

$$\beta_t = \mathrm{A}_n \beta + \mathbf{f}_n(t, \beta),$$

where

$$\mathrm{A}_n = \mathrm{U}_n^T \underbrace{(\mathrm{U}^T \mathrm{A}_\alpha \mathrm{U})}_{\text{offline}} \mathrm{U}_n,$$

$$\mathbf{f}_n(t, \beta) = \mathrm{U}_n^T \underbrace{(\mathrm{U}^T \mathrm{Y})}_{\text{offline}} \mathrm{Y}_n \Big(\underbrace{(\mathrm{P}^T \mathrm{Y})}_{\text{offline}} \mathrm{Y}_n \Big)^\dagger \underbrace{\mathrm{P}^T \mathbf{f}_\alpha(t, \mathrm{U}}_{\text{offline}} \mathrm{U}_n \beta),$$

and the approximate solution is

$$\mathbf{u}^{\mathrm{trom}}(t) = \mathrm{U}\mathrm{U}_n \beta(t) \approx \mathbf{u}(t)$$

Numerical experiments: Burgers equation

- **Burgers equation** in 1D:

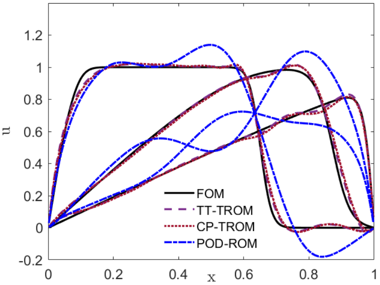
$$u_t = \alpha_1 u_{xx} - u u_x, \quad x \in (0,1), \quad t \in (0,T), \quad u(t,0) = u(t,1) = 0,$$

where $\alpha_1 > 0$ is the viscosity parameter; the second parameter is in the initial condition

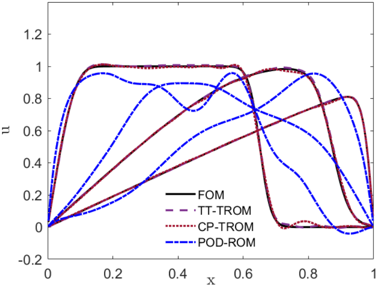
$$u_{\boldsymbol{\alpha}}(0,x) = u_0(x,\alpha_2) = \begin{cases} 1, & x \in (0,\alpha_2) \\ 0, & x \in [\alpha_2, 1) \end{cases}, \quad \text{for } \alpha_2 \in (0,1)$$

- **Two-parameter** system: $\boldsymbol{\alpha} = (\alpha_1, \alpha_2) \in \mathcal{A} = [0.01, 0.5] \times [0.2, 0.8]$
- Discretization $M = 400$, $N = 200$, training set $\widehat{\mathcal{A}}$ is a 16×32 grid
- We compare out-of-sample performance of TROM to POD-DEIM ROM for various ROM dimensions n_Φ, n_Ψ

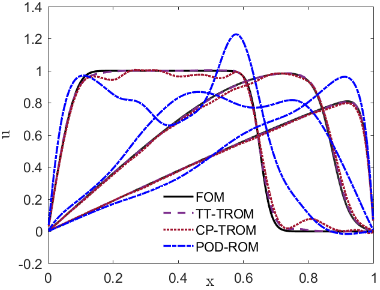
$n_\Phi = 5,\ n_\Psi = 10$



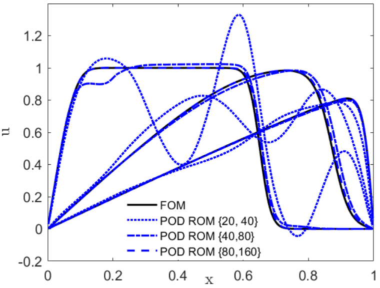
$n_\Phi = 10,\ n_\Psi = 20$



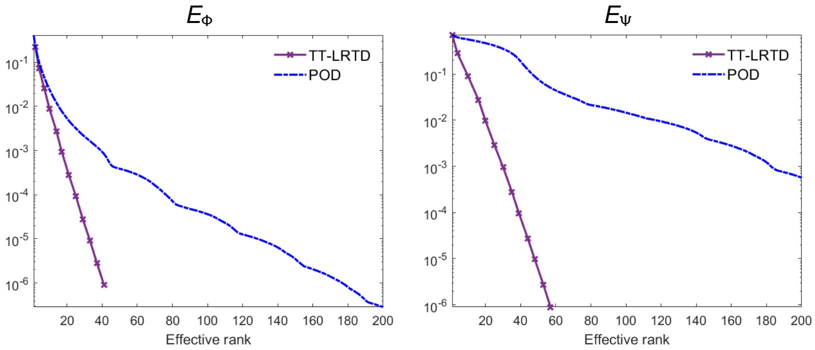
$n_\Phi = 15,\ n_\Psi = 30$



POD–DEIM ROM for $\{n_\Phi, n_\Psi\}$



TT vs POD snapshot approximation



- Relative errors

$$E_\Theta = \|\boldsymbol{\Theta} - \widetilde{\boldsymbol{\Theta}}\|_F / \|\boldsymbol{\Theta}\|_F, \quad \boldsymbol{\Theta} \in \boldsymbol{\Phi}, \boldsymbol{\Psi}$$

of snapshot tensor approximations for TT compared to POD as functions of **effective ranks**: $\widetilde{R}^{\Theta}_{D+1}$ for TT and the numbers of singular vectors for POD of $\Theta_{\rm pod}$

Numerical experiments: 2D Allen-Cahn equation

- **Allen-Cahn equation** in 2D:

$$u_t = \alpha_1^2 \Delta u - f(u), \quad \mathbf{x} \in (0,1)^2, \quad t \in (0,T),$$

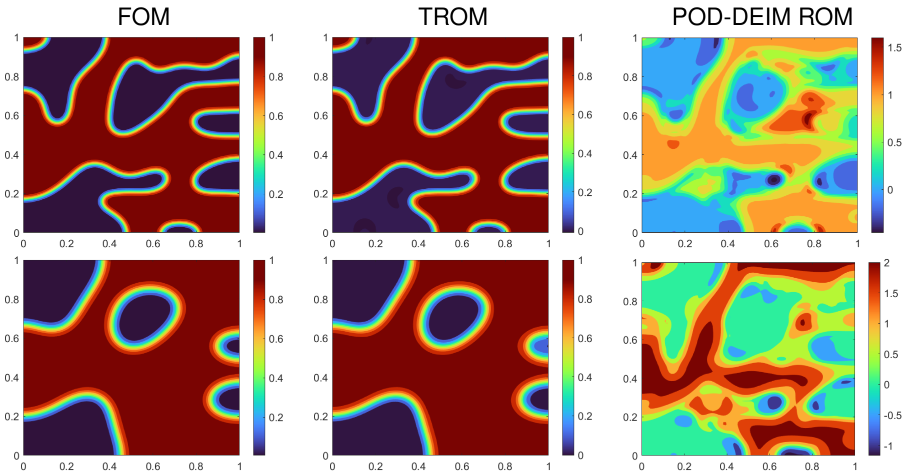
with

$$f(u) = \frac{d}{du}\left(u^2(1-u)^2 + \frac{\alpha_2}{10}(u^4 - \frac{1}{2}u)\right),$$

where α_1 is transition region width, and α_2 is the energy level of pure phases

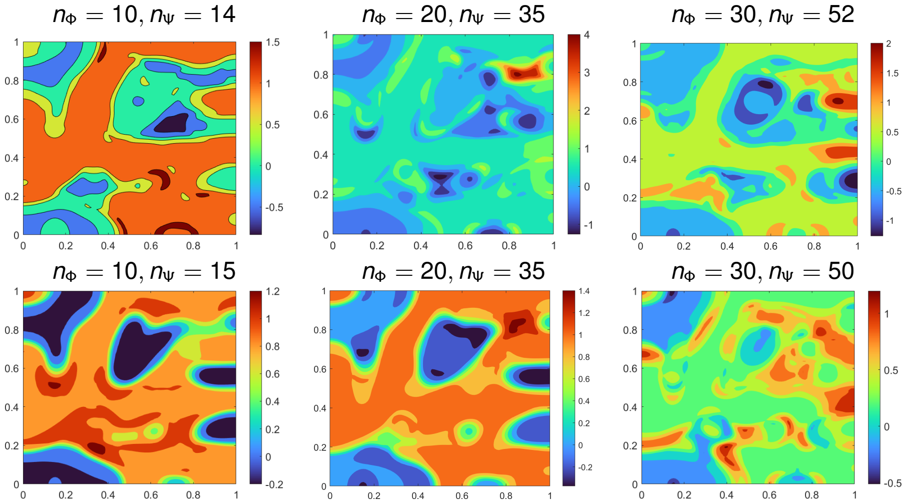
- **Initial condition** is computed from a random Bernoulli distribution with probability of $u = 1$ equal to α_3
- **Three-parameter** system with $\mathcal{A} = [0.01, 0.025] \times [0, 1] \times [0.5, 0.52]$
- Discretization with $M = 22,500$, $N = 200$, training set $\widehat{\mathcal{A}}$ is a $8 \times 3 \times 3$ grid

Out-of-sample FOM vs ROM solutions



- Solutions for two out-of-sample $\alpha \notin \widehat{\mathcal{A}}$ for $n_\Phi = n_\Psi = 20$
- TROM tensor compression accuracy is $\varepsilon = 10^{-5}$

POD-DEIM ROM solutions



- POD-DEIM ROM struggles to approximate FOM solutions for higher n_Φ and n_Ψ

Conclusions and future work

- Generalization of TROM for **parametric model reduction** from linear dynamical systems to the **nonlinear** case using **DEIM**
- The framework relies on **low-rank tensor decompositions** (LRTD) to exploit the parametric nature of the problem
- **Two-stage** approach: **offline** stage LRTD+DEIM to construct **universal ROM bases**; **online** stage for parameter-specific ROM construction and further dimension reduction
- TROM computes ROMs of **modest dimension** at a level of **accuracy** unattainable by the conventional POD-DEIM approach

Future work:

- TROM suffers from the **curse of dimensionality** due to tensor-product sampling of the training set $\widehat{\mathcal{A}}$
- Solution: employ **low-rank tensor completion** to drastically reduce the number of FOM evaluations needed for training

References

1. **This work:** *Tensorial parametric model order reduction of nonlinear dynamical systems.* A.V. Mamonov, M.A. Olshanskii, SIAM Journal on Scientific Computing 46(3):10.1137/23M1553789, 2024.
2. **TROM for linear parametric systems:** *Interpolatory tensorial reduced order models for parametric dynamical systems.* A.V. Mamonov, M.A. Olshanskii, Computer Methods in Applied Mechanics and Engineering, 397:115122, 2022.
3. **Linear TROM analysis**: *A Priori Analysis of a Tensor ROM for Parameter Dependent Parabolic Problems.* A.V. Mamonov, M.A. Olshanskii, SIAM Journal on Numerical Analysis 63(1):10.1137/23M1616844, 2025.
4. **TROM + low-rank tensor completion**: *Model order reduction of parametric dynamical systems by slice sampling tensor completion.* A.V. Mamonov, M.A. Olshanskii, arXiv:2411.07151 [math.NA]