Approximation of nD systems using Tensor Decompositions

Femke van Belzen and Siep Weiland

Department of Electrical Engineering
Eindhoven University of Technology

June 29, 2009

Motivation

Model-based process control

- Focus on dynamic behavior of the process
- Examples: crystallization, distillation, glass manufacturing, polymerization, etc.
- Both space and time as independent variables
- Models described by PDEs
  - Accurate
  - Analytical solutions unknown/hard to compute
  - Finite Element (FE) solutions: time-consuming

Problem formulation

Model

$$R\left(\frac{\partial}{\partial x_1}, \ldots, \frac{\partial}{\partial x_N}\right) w = 0.$$

$R \in \mathbb{R}^{\cdot \times 1}[\xi_1, \ldots, \xi_N]$: real matrix-valued polynomial
$w(x_1, \ldots, x_N) \in \mathbb{R}$: signal, $(x_1, \ldots, x_N) \in \mathbb{X}$

Model reduction method should:

- Preserve nD structure of original system
- Capture system dynamics relevant for control

Problem formulation

FEM/FVM gives

$$D(\varsigma_1, \ldots, \varsigma_N)w = 0.$$

$D \in \mathbb{R}^{\cdot \times 1}[\xi_1, \ldots, \xi_N]$: real matrix-valued polynomial
ς_n: forward shift operator, acting on nth mode
$w(x_1, \ldots, x_N) \in \mathbb{R}$: signal, $(x_1, \ldots, x_N) \in \mathbb{X}$

Model reduction method should:

- Preserve nD structure of original system
- Capture system dynamics relevant for control
- Retain original mesh configuration

Proper Orthogonal Decompositions

- Assume domain has Cartesian structure $\mathbb{X} = \mathbb{X}' \times \mathbb{X}''$ (Space $\times$ Time)
- Truncated spectral expansion

$$w_r(x', x'') = \sum_{n=1}^{r} a_n(x'')\xi_n(x')$$

- Reduced model:
  Collection of solutions w_r that satisfy

$$\langle D\left(\varsigma_1, \ldots, \varsigma_N\right) w_r, \xi_n \rangle = 0 \quad n = 1, \ldots, r$$

Quality determined by POD basis functions $\{\xi_n\}$

POD basis choice

POD basis: data dependent

$$
\begin{aligned}
W_{\text{snap}} &= \begin{bmatrix} w(x_1', x_1'') & \cdots & w(x_1', x_M'') \\ \vdots & \ddots & \vdots \\ w(x_L', x_1'') & \cdots & w(x_L', x_M'') \end{bmatrix} \in \mathbb{R}^{L \times M} \\
&= U \Sigma V^T
\end{aligned}
$$

POD basis functions are left singular vectors

$$U = [\xi_1, \ldots, \xi_L]$$

Disadvantages

- Ignores possible structure in $\mathbb{X}'$ by stacking all spatial information
- L is proportional to the number of grid points $\rightarrow$ Huge!

POD basis choice

POD basis: data dependent

$$
\begin{aligned}
W_{\text{snap}} &= \begin{bmatrix} w(x_1', x_1'') & \cdots & w(x_1', x_M'') \\ \vdots & \ddots & \vdots \\ w(x_L', x_1'') & \cdots & w(x_L', x_M'') \end{bmatrix} \in \mathbb{R}^{L \times M} \\
&= U \Sigma V^T
\end{aligned}
$$

POD basis functions are left singular vectors

$$U = [\xi_1, \ldots, \xi_L]$$

Disadvantages

- Ignores possible structure in $\mathbb{X}'$ by stacking all spatial information
- L is proportional to the number of grid points $\rightarrow$ Huge!

POD for nD systems

Assume structure on spatial variables

- Recall $\mathbb{X} = \mathbb{X}' \times \mathbb{X}''$ (Space $\times$ Time)
- Assume

$$\mathbb{X}' = \mathbb{X}_1 \times \cdots \times \mathbb{X}_{N-1}$$

- $\mathbb{X}_n$: finite number of grid points
- Now snapshot data can be stored in multidimensional array

$$W_{\text{snap}} \in \mathbb{R}^{L_1 \times \cdots \times L_N}$$

Basis function computation

- Solution trajectory on $\underbrace{\mathbb{X}_1 \times \cdots \times \mathbb{X}_{N-1}}_{\text{spatial domain}} \times \underbrace{\mathbb{X}_N}_{\text{time}}$
- $\mathbb{X}_n = \{x_1, \ldots, x_{L_n}\}$, $n = 1, \ldots, N$
- Associate inner product space X_n with $\mathbb{X}_n$:

$$X_n := \mathbb{R}^{L_n} \qquad (X_n, \langle \cdot, \cdot \rangle_n)$$

- Solution stored in multidimensional array $W_{\text{snap}} \in \mathbb{R}^{L_1 \times \cdots \times L_N}$
- W_{snap} defines a tensor $\mathcal{W} : \mathbb{R}^{L_1} \times \cdots \times \mathbb{R}^{L_N} \to \mathbb{R}$, represented by

$$\mathcal{W} = \sum_{\ell_1} \cdots \sum_{\ell_N} w_{\ell_1 \cdots \ell_N} e_1^{(\ell_1)} \otimes \cdots \otimes e_N^{(\ell_N)} \tag{1}$$

What is a SVD of $\mathcal{W}$?

Tensors

- Multi-indexed objects are described by Tensors
- Tensor of order-N is a **multi-linear** functional

$$\mathcal{W} : X_1 \times \ldots \times X_N \to \mathbb{R}$$

  defined on N inner product spaces $(X_n, \langle \cdot, \cdot \rangle)$ of dimension L_n
- Represented by N-way array $[[w_{\ell_1,\ldots,\ell_N}]] \in \mathbb{R}^{L_1 \times \cdots \times L_N}$ wrt bases $\{x_n^{(\ell_n)}, \ell_n = 1, \ldots, L_n\}$, $n = 1, \ldots, N$
- $w_{\ell_1 \cdots \ell_N} := \mathcal{W}(x_1^{(\ell_1)}, \ldots, x_N^{(\ell_N)})$
- A matrix is an order-2 tensor
- $U = u_1 \otimes \cdots \otimes u_N$, rank one tensor $U : X_1 \times \cdots \times X_N \to \mathbb{R}$ defined by

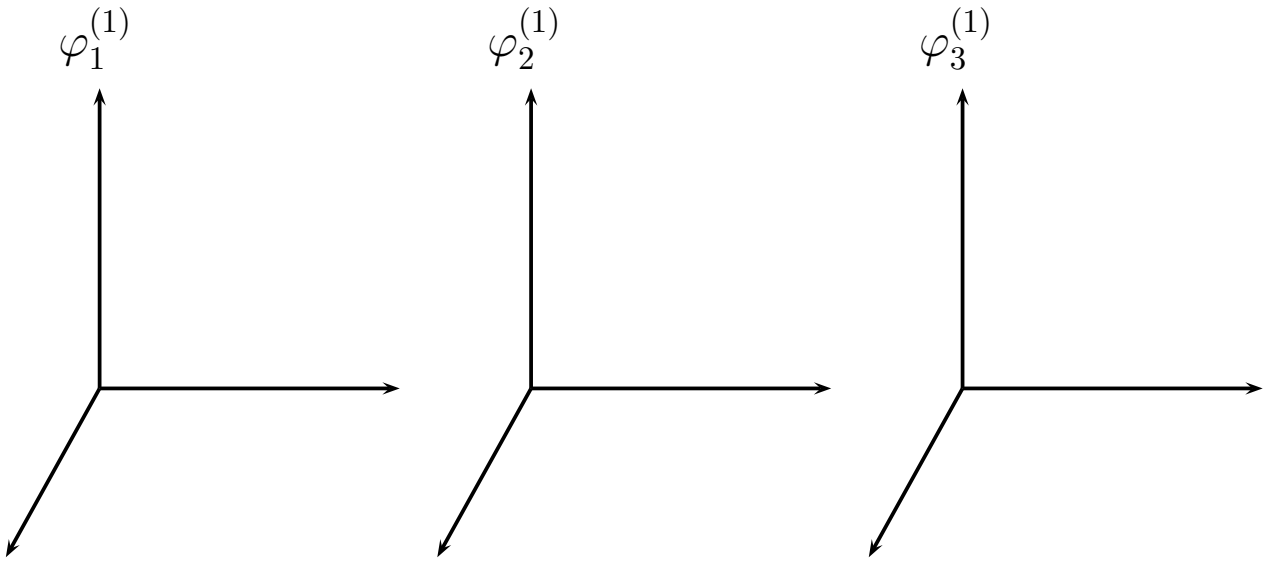
$$U(x_1, \ldots, x_N) := \prod_{n=1}^{N} \langle x_n, u_n \rangle$$

Tensor SVD (1)

First singular value

$$\sigma_1 = \max_{\substack{x_n \\ \|x_n\|=1, n=1,\ldots,N}} |\mathcal{W}(x_1, \ldots, x_N)|$$

- Maximum exists
- Yields unit vectors: $\varphi_n^{(1)} \in X_n, n = 1, \ldots, N$

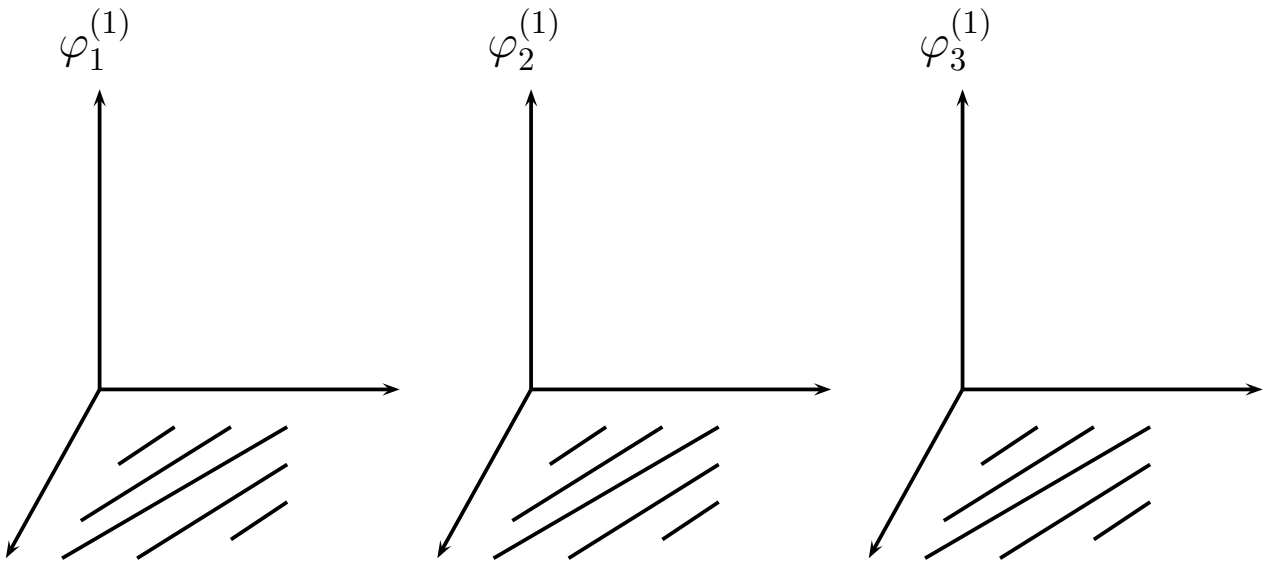


Tensor SVD (2)

Second singular value

$$\sigma_2 = \max_{\substack{x_n \\ \|x_n\|=1, n=1,\ldots,N}} |\mathcal{W}(x_1, \ldots, x_N)|$$

subject to the constraint $\left\langle x_n, \varphi_n^{(1)} \right\rangle = 0, \; n = 1, \ldots, N$



Tensor SVD (3)

Results in orthonormal bases

$$\{\varphi_n^{(\ell_n)}, \ell_n = 1, \ldots, L_n\} \quad \text{for } X_n, \quad n = 1, \ldots, N \qquad \text{(SVDbasis)}$$

Definition

A singular value decomposition of the tensor $\mathcal{W}$ is a representation of $\mathcal{W}$ with respect to the basis (SVDbasis), i.e.,

$$w_{\ell_1 \cdots \ell_N} = \mathcal{W}(\varphi_1^{(\ell_1)}, \cdots, \varphi_N^{(\ell_N)})$$

- singular values $\sigma_1, \ldots, \sigma_K$.
- singular vectors of order k are the maxima $(\varphi_1^{(k)}, \ldots, \varphi_N^{(k)})$

$\rightarrow$ Tensor SVD for $\mathcal{W} \in \mathcal{T}_2$: gives matrix SVD

Signal Reduction

- Tensor SVD gives POD basis

$$\{\varphi_n^{(\ell_n)}\}_{\ell_n=1}^{r_n} \quad n = 1, \ldots, N-1$$

- Spectral expansion for $w(x_1, \ldots, x_N)$ becomes

$$w(x_1, \ldots, x_N) = \sum_{\ell_1} \cdots \sum_{\ell_{N-1}} a_{\ell_1 \cdots \ell_{N-1}}(x_N) \varphi_1^{(\ell_1)}(x_1) \otimes \cdots \otimes \varphi_{N-1}^{\ell_{N-1}}(x_{N-1})$$

- Define truncation level $r = (r_1, \ldots, r_{N-1})$
- Signal reduction

$$w_r(x_1, \ldots, x_N) = \sum_{\ell_1=1}^{r_1} \cdots \sum_{\ell_{N-1}=1}^{r_{N-1}} a_{\ell_1 \cdots \ell_{N-1}}(x_N) \varphi_1^{(\ell_1)}(x_1) \otimes \cdots \otimes \varphi_{N-1}^{\ell_{N-1}}(x_{N-1})$$

Model Reduction

- The reduced model is defined using a Galerkin projection
- Reduced model:

  Trajectories w_r that satisfy

$$\left\langle D(\varsigma_1, \ldots, \varsigma_N) w_r, \varphi_n^{(\ell_n)} \right\rangle_n = 0$$

for $n = 1, \ldots, N-1$ and $\ell_n = 1, \ldots, r_n$

Model reduction framework summarized:

1. Determine POD basis through tensor SVD: computable!
2. Signal reduction: truncate spectral expansion
3. Model reduction: Galerkin projection

Model Reduction

- The reduced model is defined using a Galerkin projection
- Reduced model:

  Trajectories w_r that satisfy

$$\left\langle D(\varsigma_1, \ldots, \varsigma_N) w_r, \varphi_n^{(\ell_n)} \right\rangle_n = 0$$

for $n = 1, \ldots, N-1$ and $\ell_n = 1, \ldots, r_n$

Model reduction framework summarized:

1. Determine POD basis through tensor SVD: computable!
2. Signal reduction: truncate spectral expansion
3. Model reduction: Galerkin projection

Model reduction example

- 2D heat diffusion:

$$\rho c_p \frac{\partial w}{\partial t} = \kappa_{x_1} \frac{\partial^2 w}{\partial x_1^2} + \kappa_{x_2} \frac{\partial^2 w}{\partial x_2^2}$$

- $w(x_1, x_2, t)$: Temperature on position (x_1, x_2) and time t

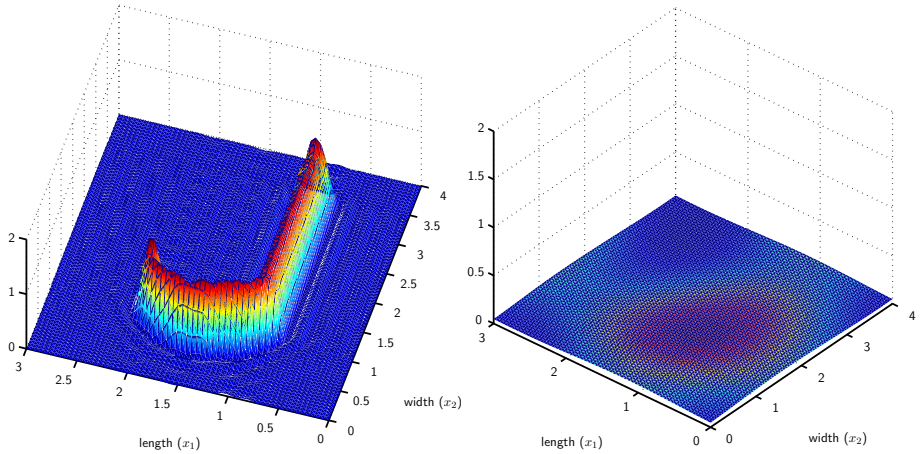


Figure: Snapshots of original data t_1 (left) and t_{L_3}

- Discretize time and space and compute a solution
- Simulation data is stored in an array: $[[w_{\ell_1 \ell_2 \ell_3}]] \in \mathbb{R}^{L_1 \times L_2 \times L_3}$

Solution using spectral expansions

- Compute TSVD for measurement $[[w_{\ell_1\ell_2\ell_3}]]$
  $\rightarrow$ gives orthonormal bases $\{\varphi_1^{(\ell_1)}\}$ for X_1,$\{\varphi_2^{(\ell_2)}\}$ for X_2, $\{\varphi_3^{(3)}\}$ for $T := \mathbb{R}^{L_3}$
- Use these in spectral decomposition to separate time and space

$$w_r(x_1, x_2, t) = \sum_{\ell_1=1}^{r_1} \sum_{\ell_2=1}^{r_2} a_{\ell_1\ell_2}(t)\varphi_1^{(\ell_1)}(x_1)\varphi_2^{(\ell_2)}(x_2)$$

- Traditional spectral expansion

$$w_r(x_1, x_2, t) = \sum_{k=1}^{r} b_k(t)\xi_k(x_1, x_2)$$

- $\{\xi_k\}_1^r$: orthonormal basis for $\mathbb{X} = \mathbb{R}^{L_1 \cdot L_2}$, computed by re-arranging $[[w_{\ell_1\ell_2\ell_3}]]$ into a matrix

Tensor SVD results

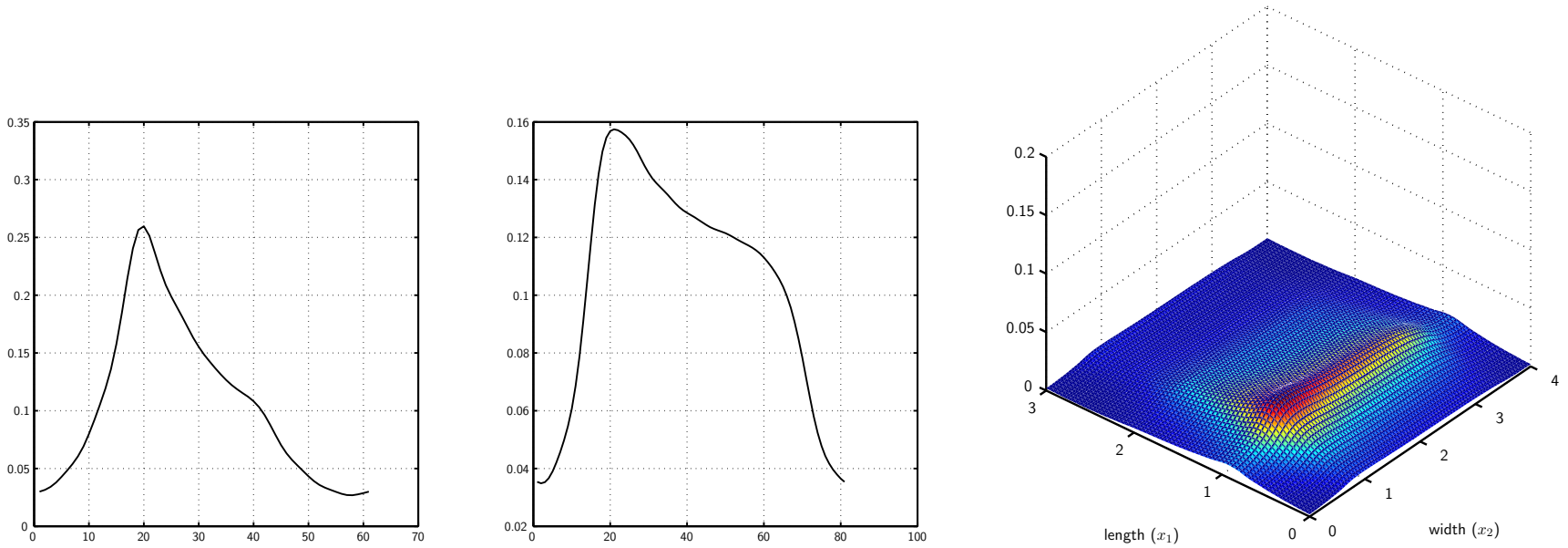


Figure: First basis function for X_1 (left), X_2 (middle) and $X_1 \times X_2$ (right)

Model Reduction: Galerkin projection

Reduced model is defined as:

$$w_r(x_1, x_2, t) = \sum_{\ell_1=1}^{r_1} \sum_{\ell_2=1}^{r_2} a_{\ell_1 \ell_2}(t) \varphi_1^{(\ell_1)}(x_1) \varphi_2^{(\ell_2)}(x_2)$$

with $a_{kl}(t) = [A(t)]_{kl}$ a solution of

$$\rho c_p \dot{A} = \kappa_{x_1} F A + \kappa_{x_2} A P$$

with F and P defined as:

$$F = \begin{bmatrix} \left\langle \varphi_1^{(1)}, \ddot{\varphi}_1^{(1)} \right\rangle & \dots & \left\langle \varphi_1^{(1)}, \ddot{\varphi}_1^{(r_1)} \right\rangle \\ \vdots & & \vdots \\ \left\langle \varphi_1^{(r_1)}, \ddot{\varphi}_1^{(1)} \right\rangle & \dots & \left\langle \varphi_1^{(r_1)}, \ddot{\varphi}_1^{(r_1)} \right\rangle \end{bmatrix}; \quad P = \begin{bmatrix} \left\langle \varphi_2^{(1)}, \ddot{\varphi}_2^{(1)} \right\rangle & \dots & \left\langle \varphi_2^{(1)}, \ddot{\varphi}_2^{(r_2)} \right\rangle \\ \vdots & & \vdots \\ \left\langle \varphi_2^{(r_2)}, \ddot{\varphi}_2^{(1)} \right\rangle & \dots & \left\langle \varphi_2^{(r_2)}, \ddot{\varphi}_2^{(r_2)} \right\rangle \end{bmatrix}$$

Obtained through a Galerkin projection of the PDE residual

$$\left\langle D(\varsigma_1, \varsigma_2, \varsigma_3) w_r, \varphi_n^{(\ell_n)} \right\rangle = 0; \quad 1 \leq \ell_n \leq r_n \quad n = 1, 2$$

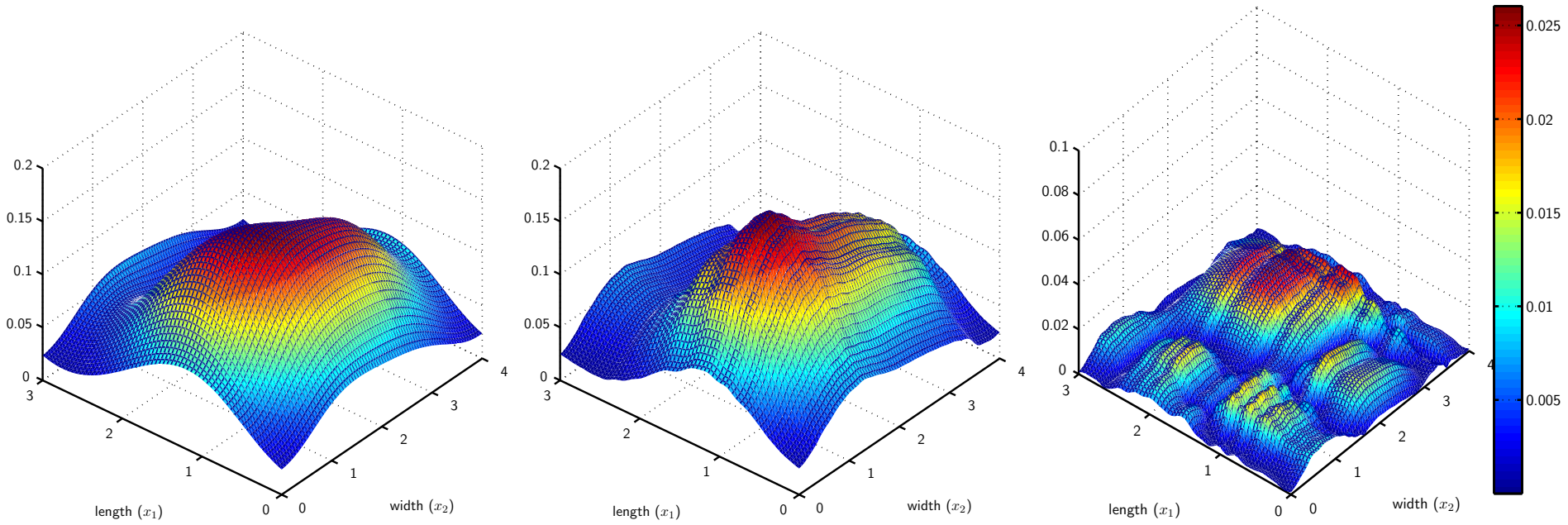


Figure: Time slice of original data at time t_{40}(left), time slice of reduced model of order $(7, 7)$ at time t_{40} (middle) and time slice of absolute error at time t_{40} (right).

Summary and Conclusions

Summary

- Adapted POD to nD systems
- Projection-based
- Projection spaces are computed using tensor decompositions

What did we gain?

- Basis-independent tensor SVD: computable!
- nD structure preserved in reduced model
- Reduction in each vector space separately
  $\rightarrow$ Truncation level $r = (r_1, \ldots, r_{N-1})$

Future work

Future work

- Include multiple dependent variables in this framework
- Conservation of system properties:
  - Stability
  - Dissipativity
  - Conservation of mass in flow models
- Test industrial benchmarks
- Develop tensor methods further:

  Many algebraic concepts do not exist for tensors!