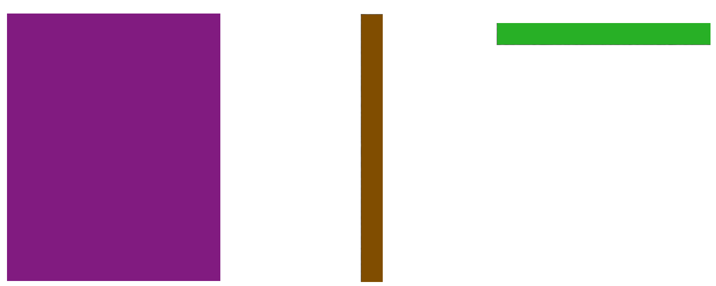
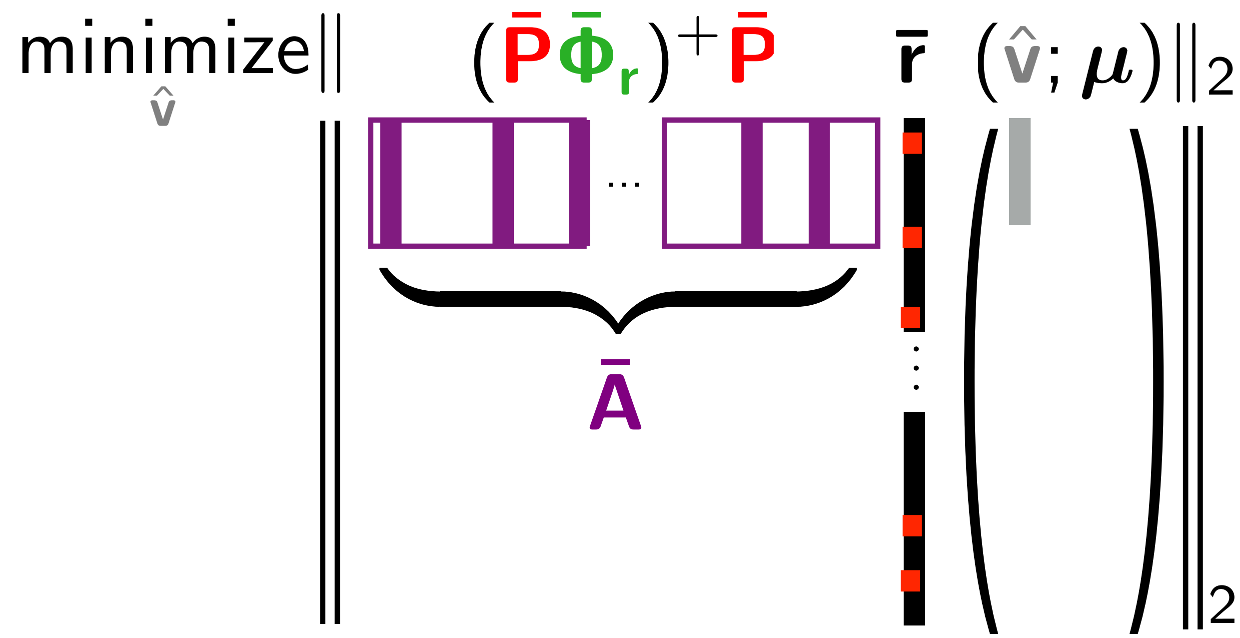
Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction

$\boldsymbol{\pi}_{\mathcal{J}(i,j)} = \phi_i \otimes \xi_j$





Youngsoo Choi, Kevin Carlberg

Sandia National Laboratories

Livermore, California

MoRePaS IV, Nantes, France

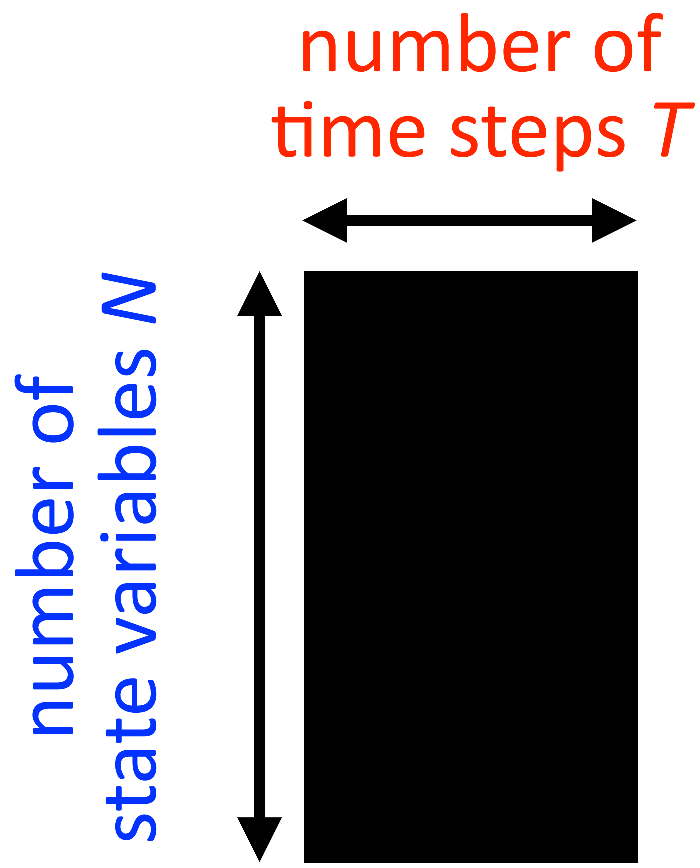
April 12, 2018

Work was performed while Youngsoo Choi was employed in the Extreme-scale Data Science and Analytics Department, Sandia National Laboratories, Livermore, CA 94550. Current affiliation: Lawrence Livermore National Laboratory.

Motivation

ODE: $$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu}); \quad \mathbf{x}(0, \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}), \quad t \in [0, T_{\text{final}}], \quad \boldsymbol{\mu} \in \mathcal{D}$$

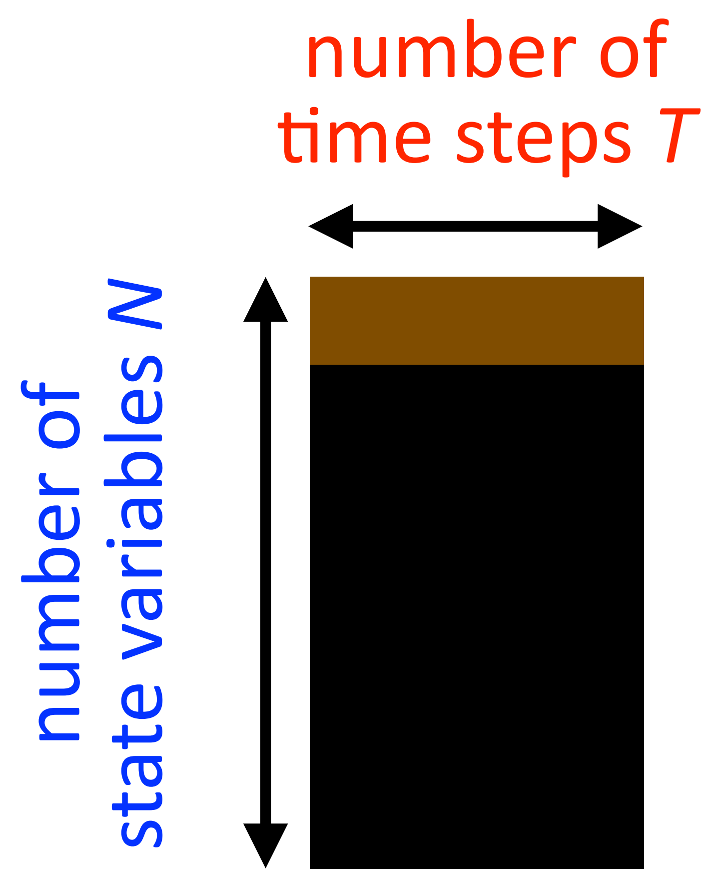
OΔE: $$\mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T, \quad \boldsymbol{\mu} \in \mathcal{D}$$



Motivation

$$\text{ODE:} \quad \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu}); \quad \mathbf{x}(0, \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}), \quad t \in [0, T_{\text{final}}], \quad \boldsymbol{\mu} \in \mathcal{D}$$

$$\text{O}\Delta\text{E:} \quad \mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T, \quad \boldsymbol{\mu} \in \mathcal{D}$$

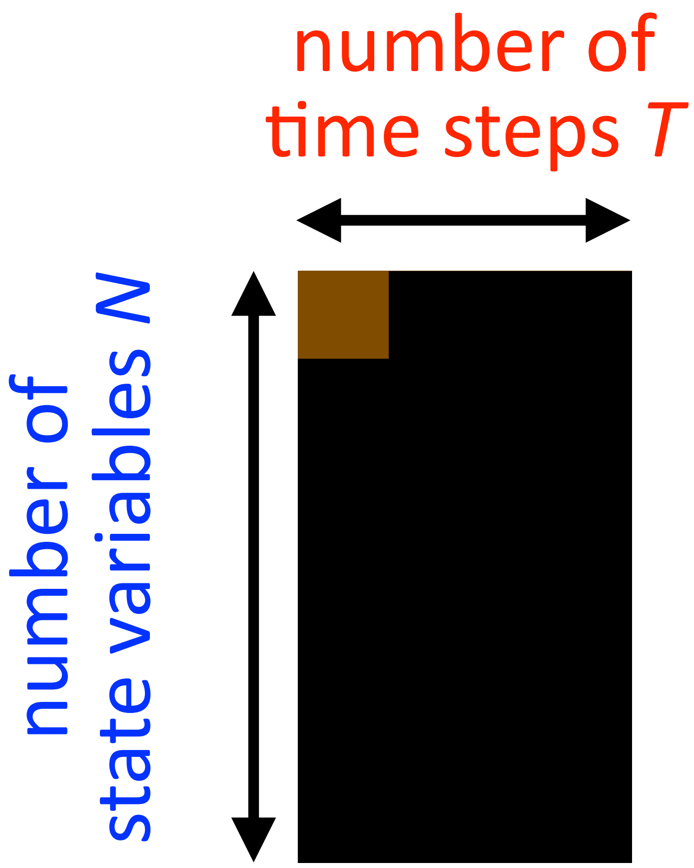


Most ROMs for nonlinear dynamical systems use
spatial simulation data to reduce the spatial dimension and complexity

Motivation

ODE: $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}; t, \boldsymbol{\mu}); \quad \mathbf{x}(0, \boldsymbol{\mu}) = \mathbf{x}^0(\boldsymbol{\mu}), \quad t \in [0, T_{\text{final}}], \quad \boldsymbol{\mu} \in \mathcal{D}$

OΔE: $\mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T, \quad \boldsymbol{\mu} \in \mathcal{D}$

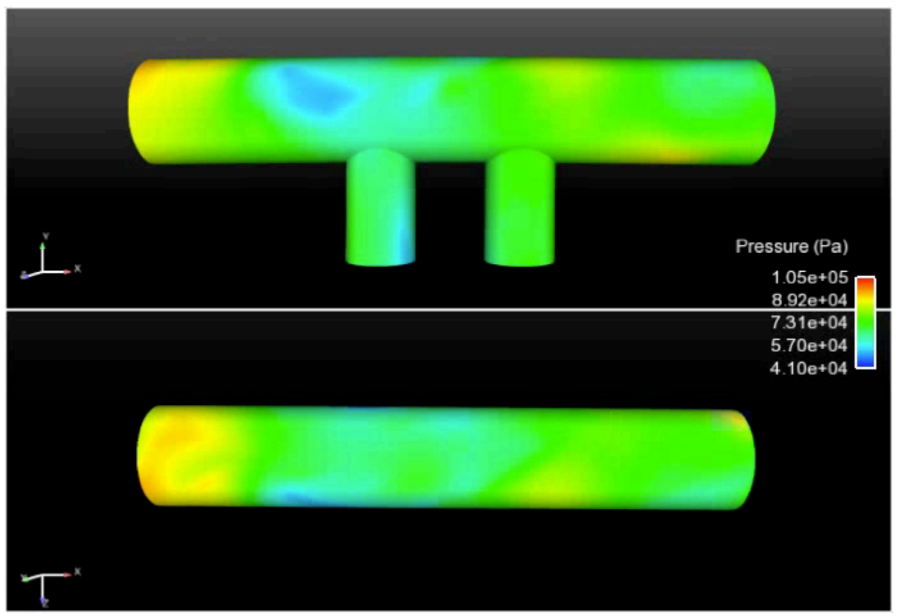


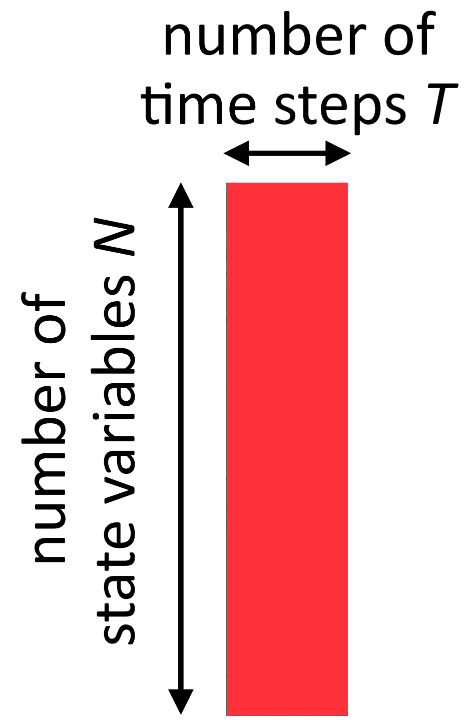
Most ROMs for nonlinear dynamical systems use spatial simulation data to reduce the spatial dimension and complexity

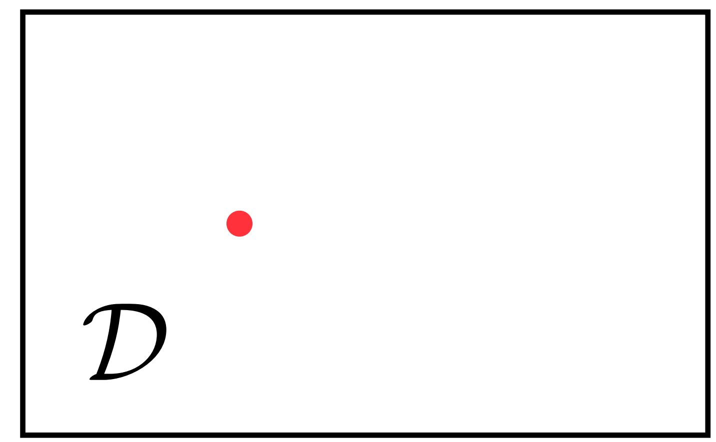
Goal: use temporal simulation data to reduce the temporal dimension and complexity

Offline step 1: data collection

$$\text{O}\Delta\text{E:}\quad \mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T$$

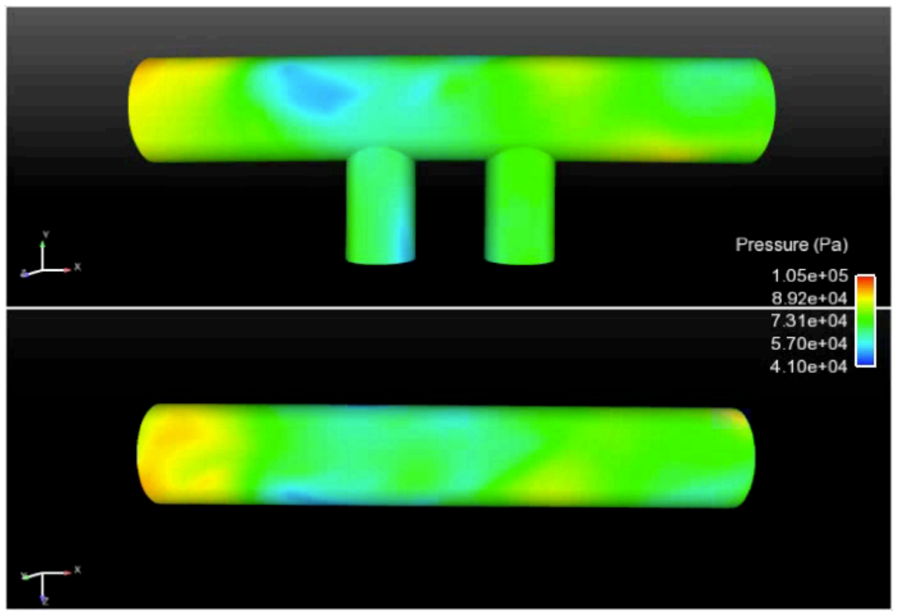




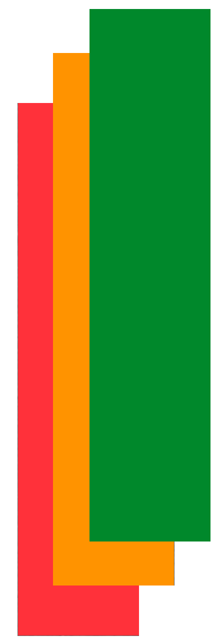


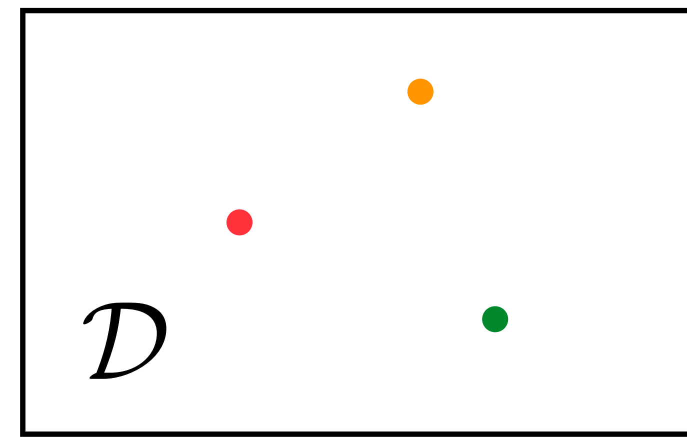
Offline step 1: data collection

$$\text{O}\Delta\text{E:} \quad \mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T$$



$\mathcal{X} =$

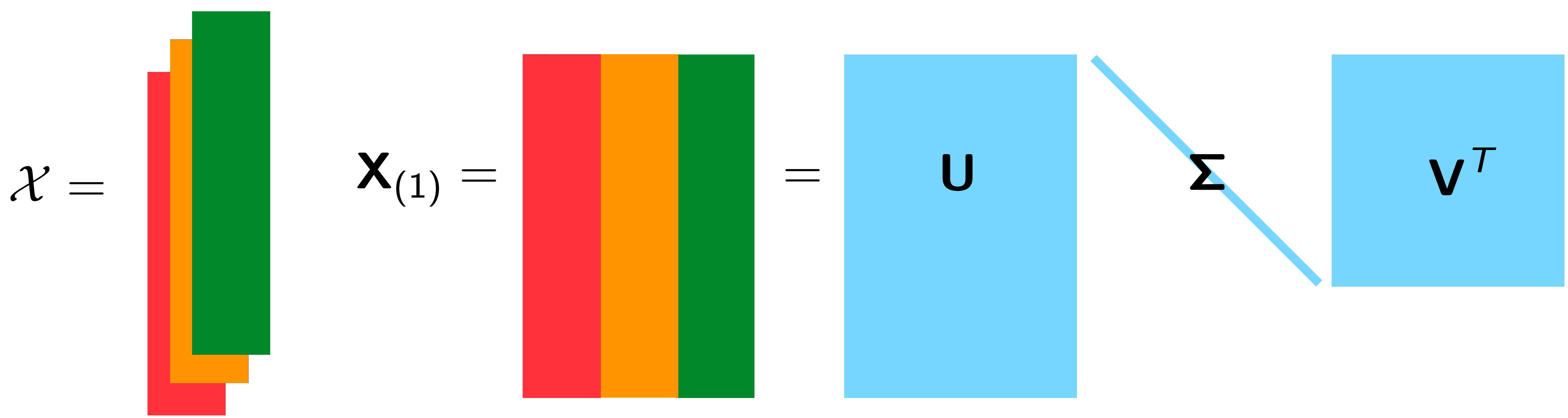




Offline step 2: Tensor decomposition (POD)

$$\text{O}\Delta\text{E:} \quad \mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T$$

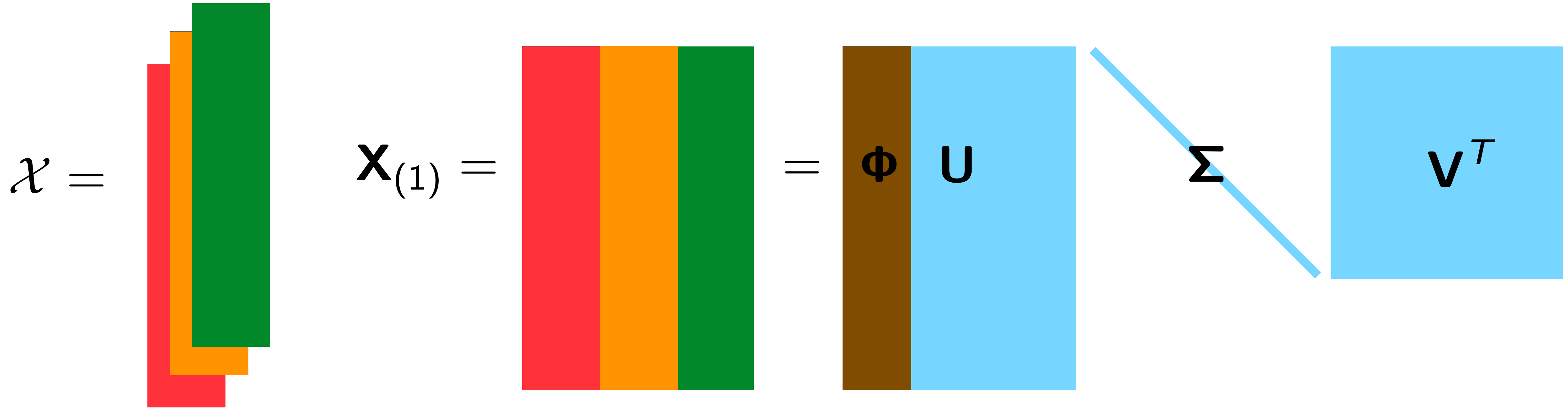
Compute dominant left singular vectors of mode-1 unfolding



Offline step 2: Tensor decomposition (POD)

$$\text{O}\Delta\text{E:} \quad \mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T$$

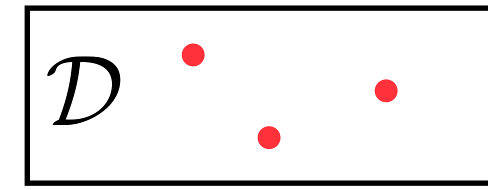
Compute dominant left singular vectors of mode-1 unfolding



$\boldsymbol{\Phi}$ *columns are principal components of the spatial simulation data*

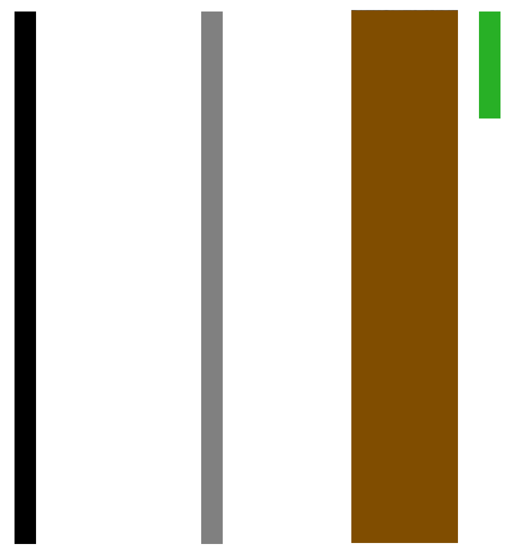
Online: LSPG projection [C., Bou–Mosleh, Farhat, 2011]

OΔE: $\mathbf{r}^n(\mathbf{x}^n, \ldots, \mathbf{x}^{n-k}; \boldsymbol{\mu}) = \mathbf{0}, \quad n = 1, \ldots, T$



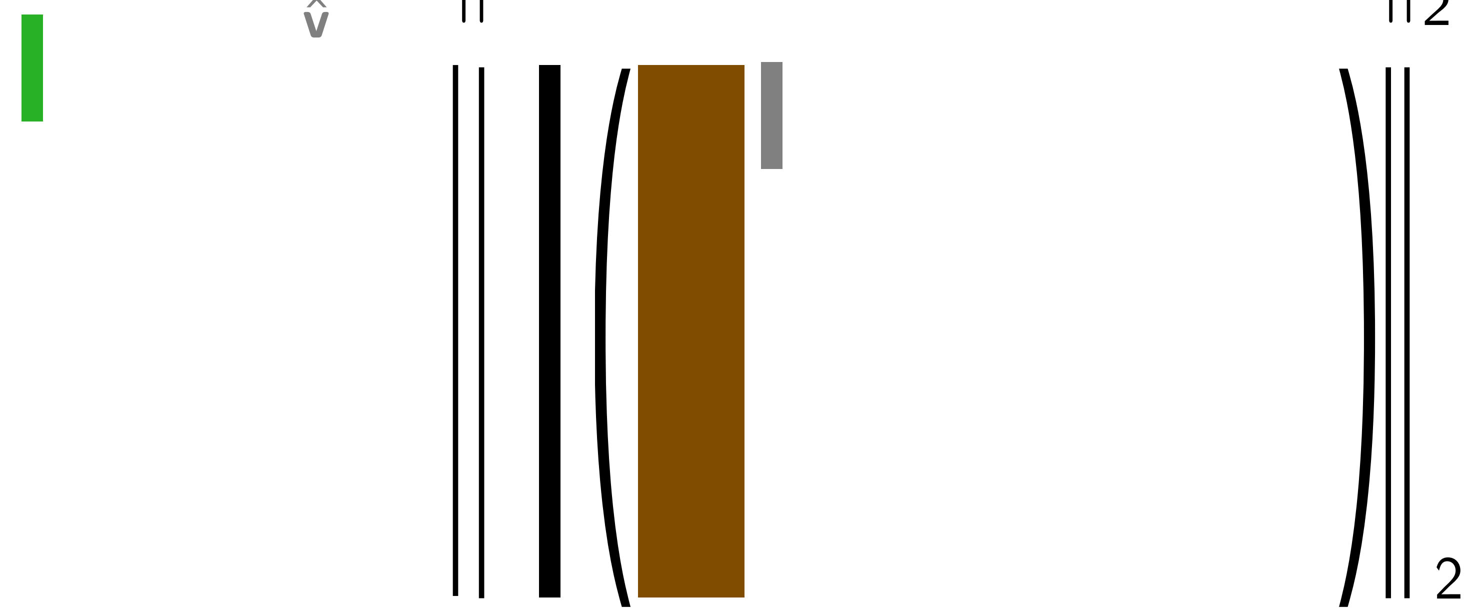
1. Reduce number of spatial unknowns

$$\mathbf{x}^n \approx \tilde{\mathbf{x}}^n = \boldsymbol{\Phi}\, \hat{\mathbf{x}}^n$$

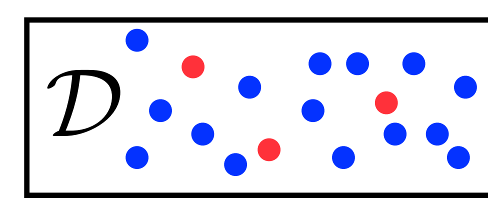


2. Minimize OΔE residual

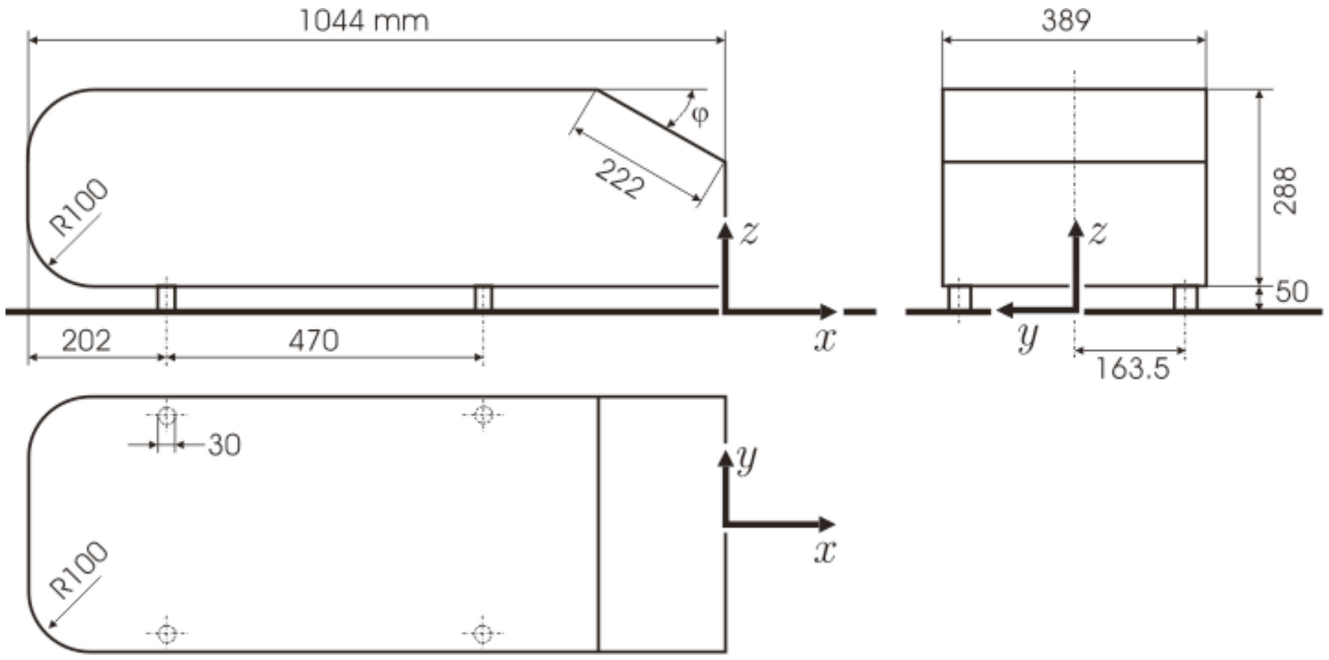
$$\hat{\mathbf{x}}^n = \arg\min_{\hat{\mathbf{v}}} \left\| \mathbf{A}\mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2$$

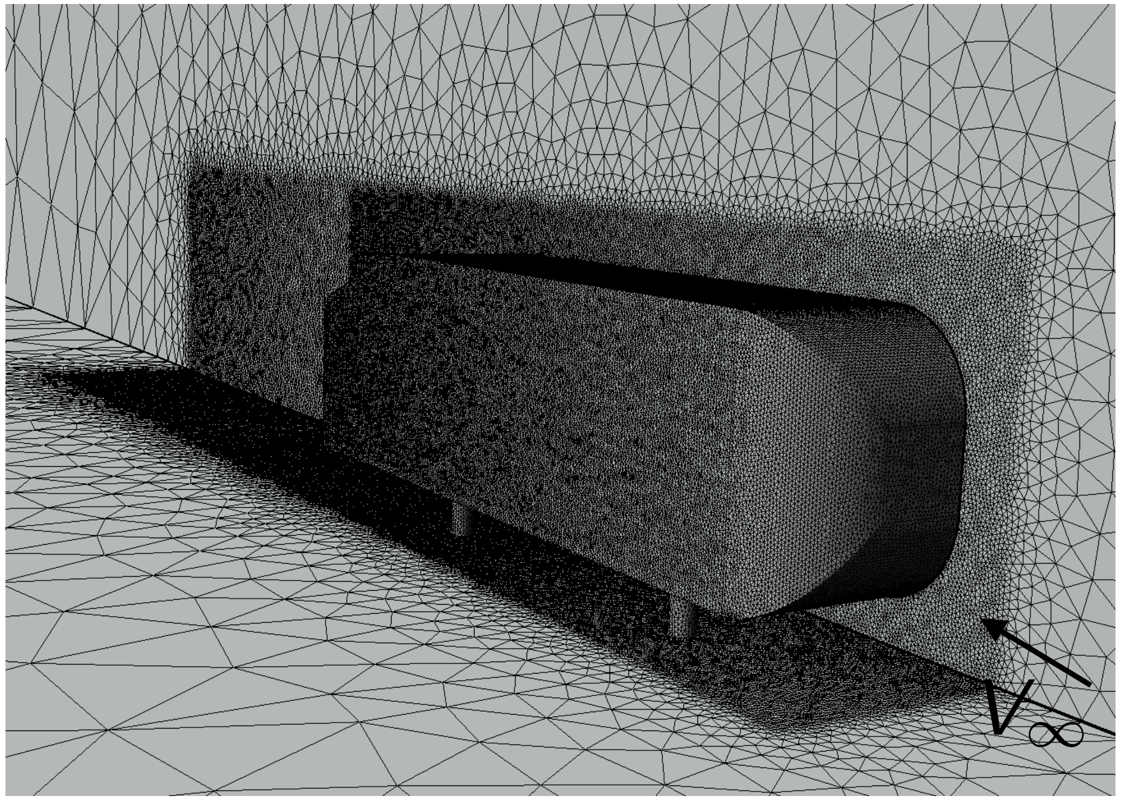


LSPG OΔE: $\hat{\mathbf{x}}^n = \arg\min_{\hat{\mathbf{v}}} \left\| \mathbf{A}\mathbf{r}^n(\mathbf{x}^0 + \boldsymbol{\Phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2^2$



Ahmed body [Ahmed, Ramm, Faitin, 1984]





- Unsteady Navier–Stokes
- Re = 4.3 x 10^6
- M_∞ = 0.175

Spatial discretization

- 2nd-order finite volume
- DES turbulence model
- 1.7×10^7 degrees of freedom

Temporal discretization

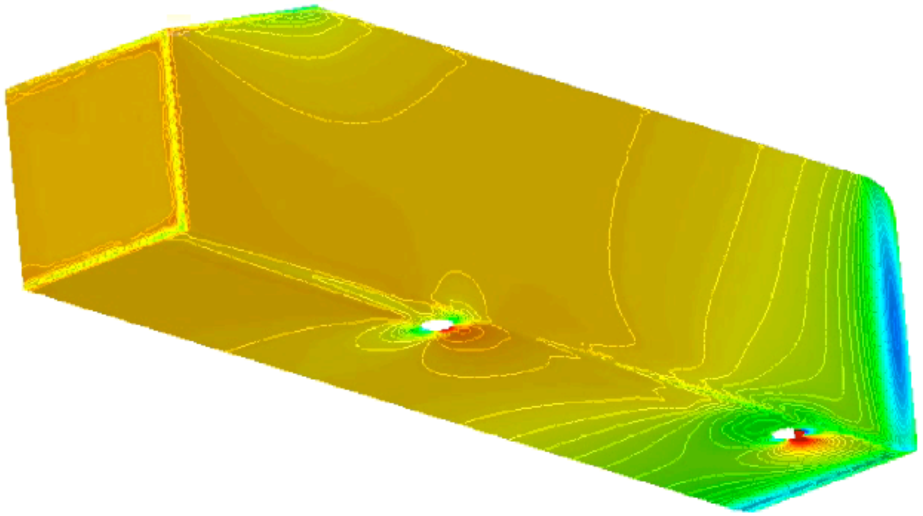
- 2nd-order BDF
- Time step $\Delta t = 8 \times 10^{-5}\mathrm{s}$
- 1.3×10^3 time instances

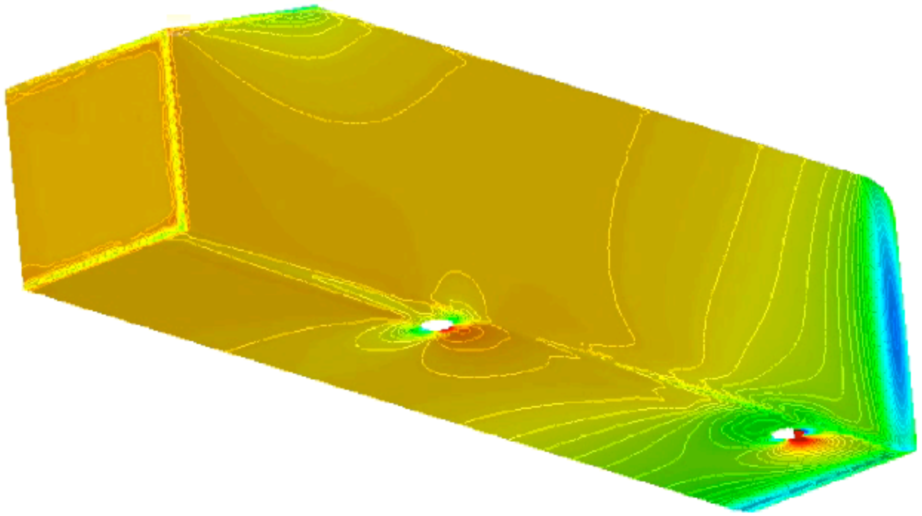
Ahmed body results [C., Farhat, Cortial, Amsallem, 2013]

GNAT ROM ($\mathbf{A} = (\mathbf{P}\boldsymbol{\Phi}_r)^{+}\mathbf{P}$)
4 hours, 4 cores
spatial dim: 283
temporal dim: 1.3 x 10^3

high-fidelity model
13 hours, 512 cores
spatial dim: 1.7 x 10^7
temporal dim: 1.3 x 10^3

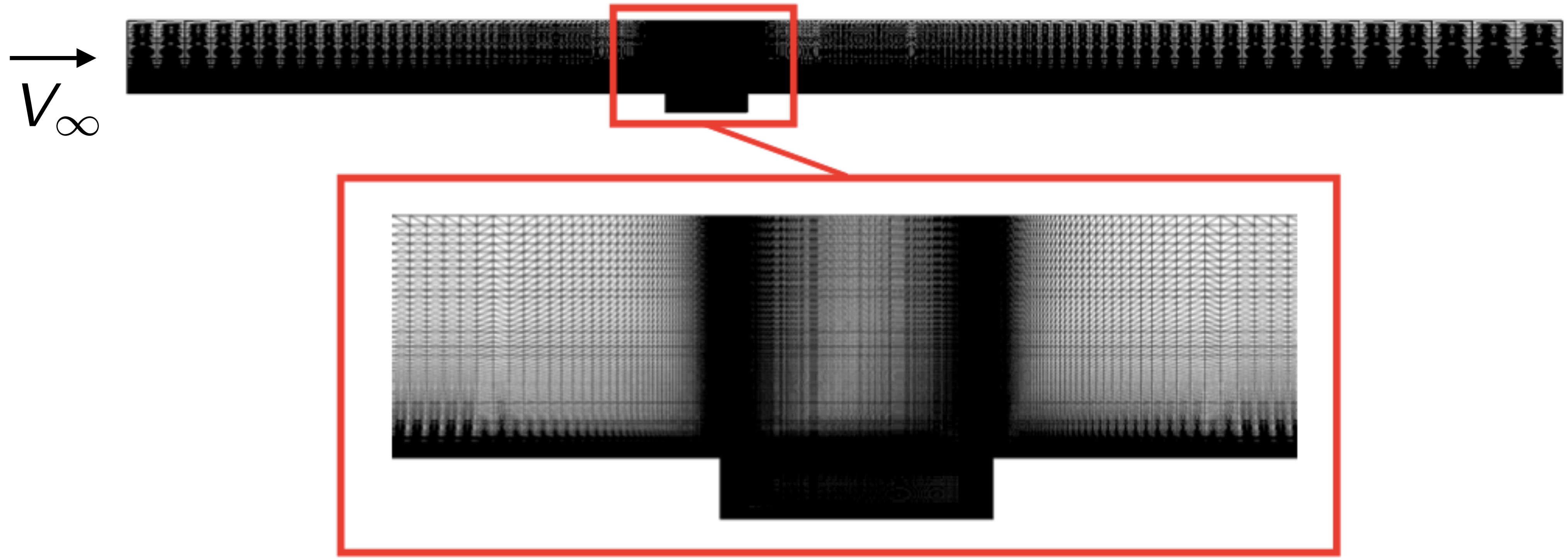
pressure field





+ *438X computational-cost reduction*
+ *60,500X spatial-dimension reduction*
- *Zero temporal-dimension reduction*

B61 captive carry



- Unsteady Navier–Stokes
- Re = 6.3 x 10^6
- $M_\infty = 0.6$

Spatial discretization

- 2nd-order finite volume
- DES turbulence model
- 1.2×10^6 degrees of freedom

Temporal discretization

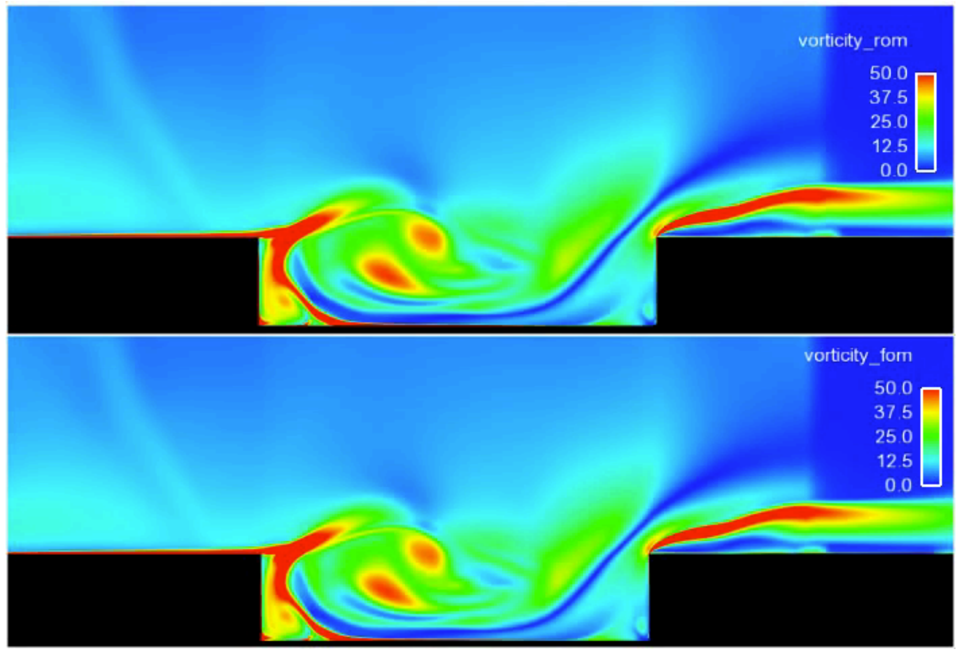
- 2nd-order BDF
- Verified time step $\Delta t = 1.5 \times 10^{-3}$
- 8.3×10^3 time instances

Turbulent-cavity results [C., Barone, Antil, 2017]

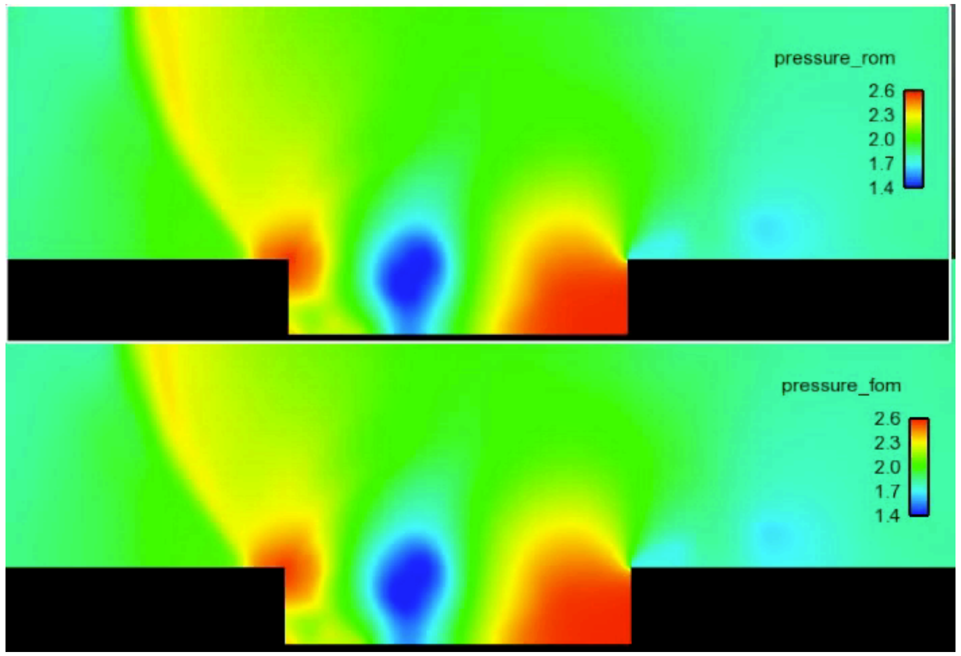
GNAT ROM
32 min, 2 cores
spatial dim: 179
temporal dim: 458

high-fidelity
5 hours, 48 cores
spatial dim: 1.2M
temporal dim: 3,700

vorticity field



pressure field



\+ *229X computational-cost reduction*
\+ *6,500X spatial-dimension reduction*
\- *8X temporal-dimension reduction*

How can we significantly reduce the temporal dimensionality?

Reducing temporal complexity: existing work

Larger time steps with ROM

[Krysl et al., 2001; Lucia et al., 2004; Taylor et al., 2010; C. et al., 2017]

- Developed for explicit and implicit integrators
- – Limited reduction of time dimension: <10X reductions typical

Forecasting using gappy POD in time

- Accurate Newton-solver initial guess [C., Ray, van Bloemen Waanders, 2015]
- Coarse propagator in time-parallel setting [C., Brencher, Haasdonk, Barth, 2016]
- + No error incurred and wall-time improvements observed
- – No reduction of time dimension

Space–time ROMs

- Reduced basis [Urban, Patera, 2012; Yano, 2013; Urban, Patera, 2014; Yano, Patera, Urban, 2014]
- POD–Galerkin [Volkwein, Weiland, 2006; Baumann, Benner, Heiland, 2016]
- ODE-residual minimization [Constantine, Wang, 2012]
- + Reduction of time dimension
- + Linear time-growth of error bounds ˆ
- – Requires space–time finite element discretization ˆ
- – No hyper-reduction *
- – Only one space–time basis vector per training simulation †

ˆ Only reduced-basis methods * Except [Constantine, Wang, 2012] † Except [Baumann, Benner, Heiland, 2016]

Goals

Preserve attractive properties of existing space–time ROMs

+ Reduce both space and time dimensions
+ Slow time-growth of error bound

Overcome shortcomings of existing space–time ROMs

+ Applicability to general nonlinear dynamical systems
+ Hyper-reduction to reduce complexity of nonlinearities
+ Extract multiple space–time basis vectors from each training simulation

Space–time least-squares Petrov–Galerkin (ST-LSPG) projection

Reference: Choi and C. Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction. *arXiv e-print*, (1703.04560), 2017.

Spatial v. spatiotemporal trial subspaces

Full-order-model trial subspace

$$\begin{bmatrix}\mathbf{x}^1 & \cdots & \mathbf{x}^T\end{bmatrix} \in \mathbb{R}^N \otimes \mathbb{R}^T$$

Spatial trial subspace

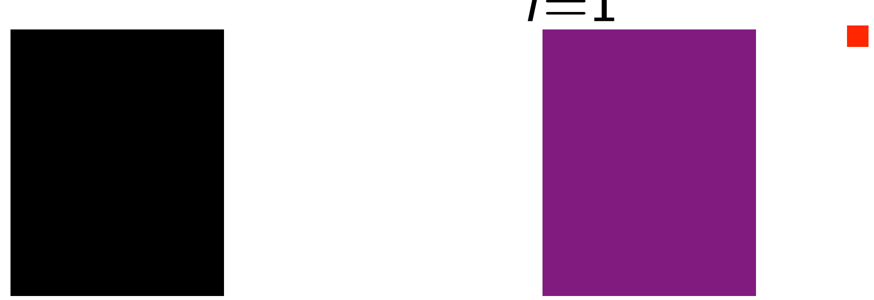
$$\begin{bmatrix}\tilde{\mathbf{x}}^1 & \cdots & \tilde{\mathbf{x}}^T\end{bmatrix} = \mathbf{\Phi} \begin{bmatrix}\hat{\mathbf{x}}^1 & \cdots & \hat{\mathbf{x}}^T\end{bmatrix} \in \mathcal{S} \otimes \mathbb{R}^T \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

+ Spatial dimension reduced
- Temporal dimension large

Space–time trial subspace

$$\begin{bmatrix}\tilde{\mathbf{x}}^1 & \cdots & \tilde{\mathbf{x}}^T\end{bmatrix} = \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i \hat{x}_i(\boldsymbol{\mu}) \in \mathcal{S}\mathcal{T} \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

How to compute space–time bases $\boldsymbol{\pi}_i$ ***?***



+ Spatial dimension reduced
+ Temporal dimension reduced
- Additional approximation

Space–time basis computation

$\mathcal{X} =$ 

Tensor slices

[Urban, Patera, 2012; Yano, 2013; Urban, Patera, 2014; Yano, Patera, Urban, 2014; Volkwein, Weiland, 2006; Constantine, Wang, 2012]

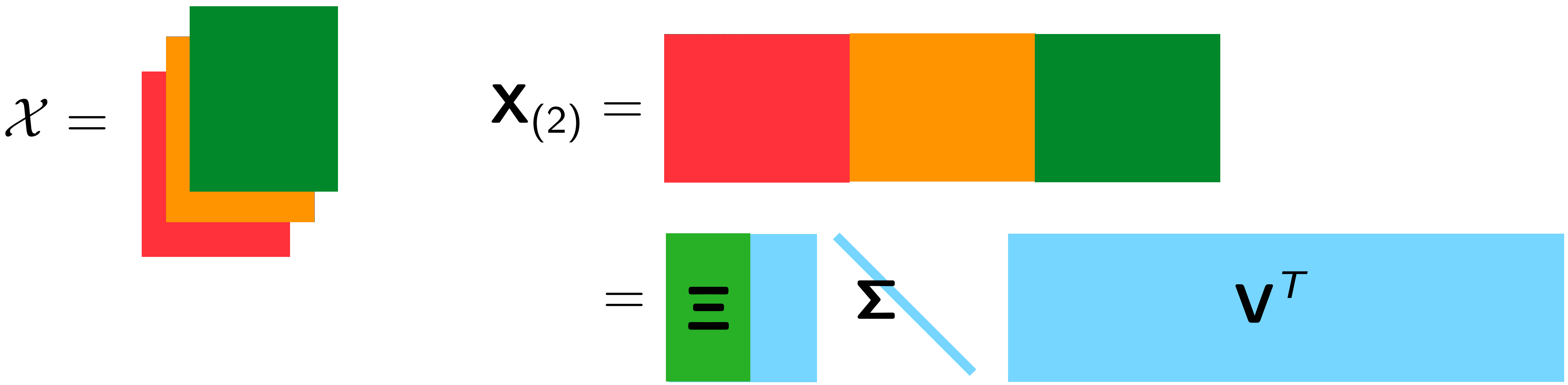
$$\boldsymbol{\pi}_i = \mathbf{X}_i$$

+ General space–time structure
- Only one basis vector per training simulation
- NT storage per basis vector

Space–time basis computation

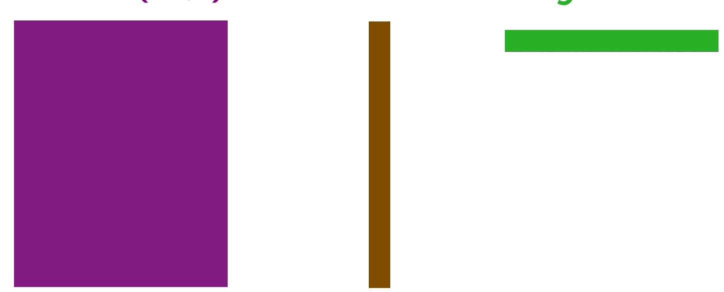
Truncated high-order SVD (T-HOSVD) [Baumann, Benner, Heiland, 2016]

- Compute dominant left singular vectors of **mode-2** unfolding



$\boldsymbol{\Xi}$ *columns are principal components of the* ***temporal*** *simulation data*

$$\boldsymbol{\pi}_{\mathcal{J}(i,j)} = \boldsymbol{\phi}_i \otimes \boldsymbol{\xi}_j$$

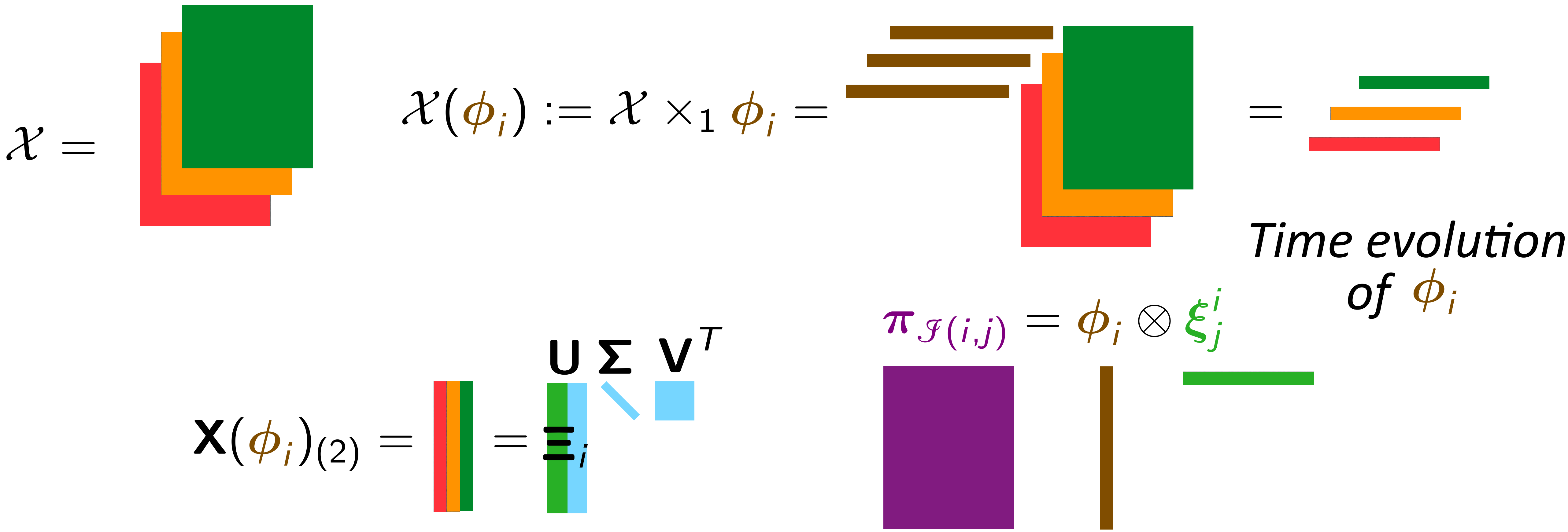


+ Multiple basis vectors per training simulation
+ N+T storage per basis vector
- Enforces Kronecker–product structure
- Same temporal modes for each spatial mode

Space–time basis computation

Sequentially truncated high-order SVD (ST-HOSVD)

[C., Ray, van Bloemen Waanders, 2015; C., Brencher, Haasdonk, Barth, 2016]



$\mathbf{\Xi}_i$ columns are principal components of the ***temporal*** *simulation data of ϕ_i*

- \+ Multiple basis vectors per training simulation
- \+ N+T storage per basis vector
- \+ Tailored temporal modes for each spatial mode
- – Enforces Kronecker-product structure

How to project governing equations?

Space–time LSPG projection

LSPG

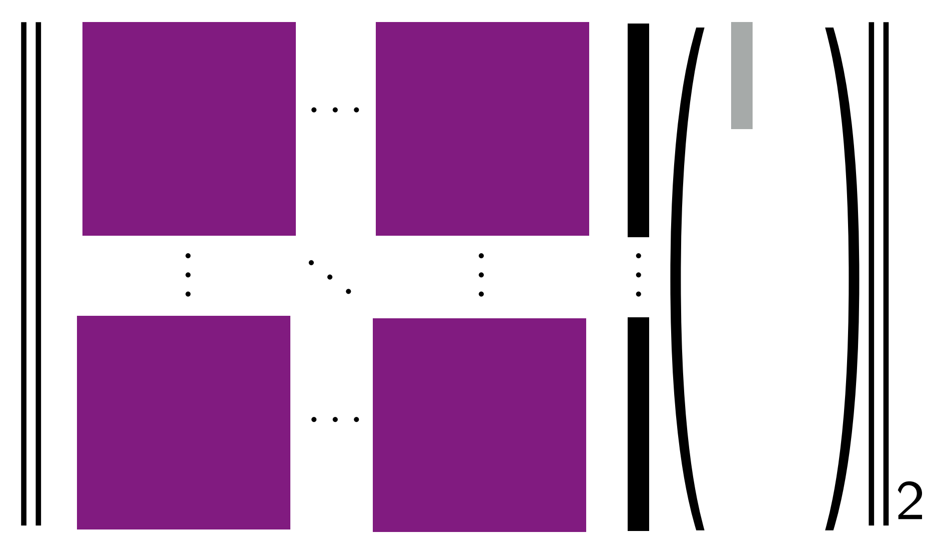
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2, \quad n = 1, \ldots, T$$

\+ efficient: time-sequential solve

ST-LSPG

$$\bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) := \begin{bmatrix} \mathbf{r}^1\left(\sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^1)\hat{v}_i, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^0)\hat{v}_i; \boldsymbol{\mu}\right) \\ \vdots \\ \mathbf{r}^T\left(\sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^T)\hat{v}_i, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^{T-1})\hat{v}_i, \ldots, \sum_{i=1}^{n_{st}} \boldsymbol{\pi}_i(t^{T-k})\hat{v}_i; \boldsymbol{\mu}\right) \end{bmatrix}$$

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}}; \boldsymbol{\mu}) \right\|_2$$



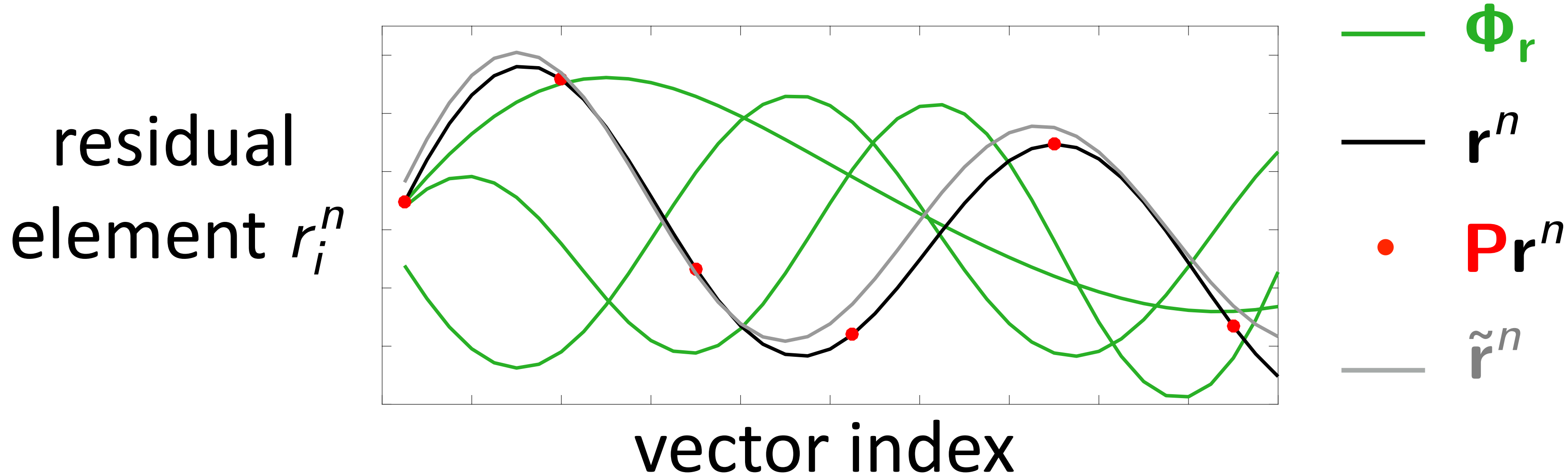
\- costly: minimizing residual simultaneously over space and time

LSPG hyper-reduction

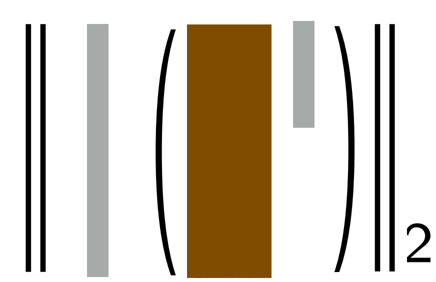
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2$$

Select **A** *to make this less expensive*

$$\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_{\mathbf{r}}(\mathbf{P}\boldsymbol{\Phi}_{\mathbf{r}})^{+}\mathbf{P}\mathbf{r}^n$$



$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \tilde{\mathbf{r}}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \right\|_2$$

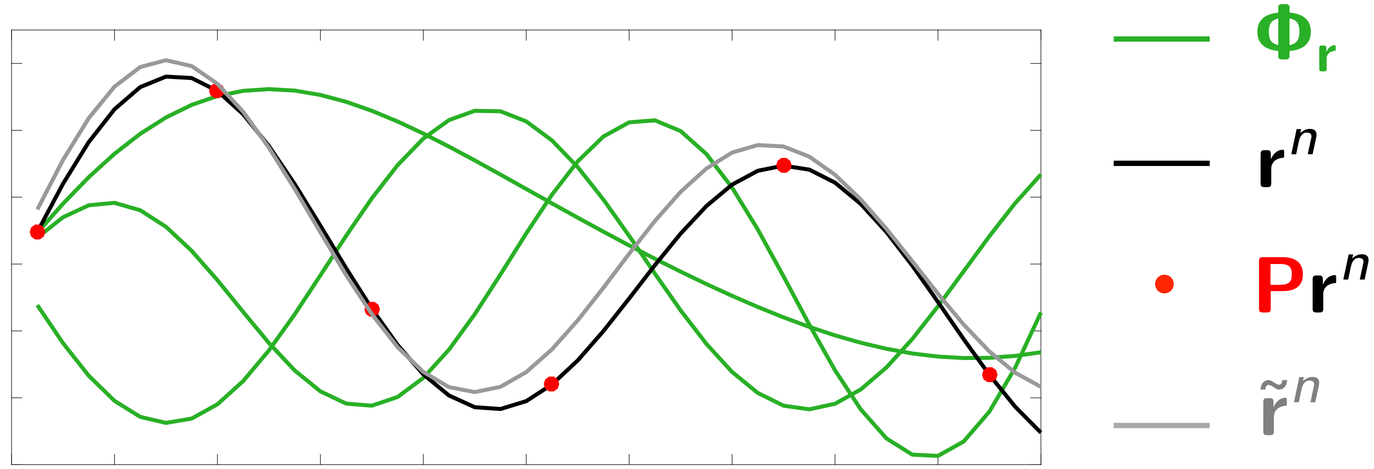


LSPG hyper-reduction

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{A} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}, \tilde{\mathbf{x}}^{n-1}, \ldots, \tilde{\mathbf{x}}^{n-k}; \boldsymbol{\mu}) \right\|_2$$

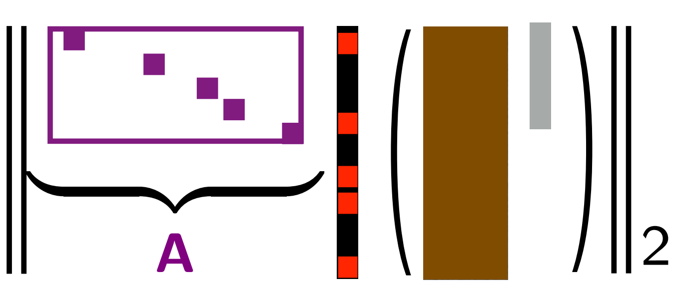
Select **A** *to make this less expensive*

$$\mathbf{r}^n \approx \tilde{\mathbf{r}}^n = \boldsymbol{\Phi}_\mathbf{r}(\mathbf{P}\boldsymbol{\Phi}_\mathbf{r})^+\mathbf{P}\mathbf{r}^n$$



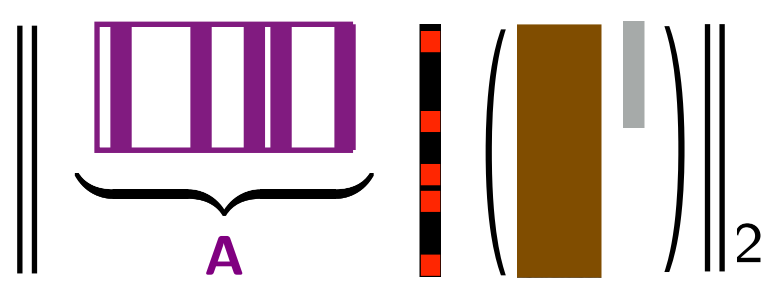
LSPG-collocation

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| \mathbf{P} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \right\|_2$$



GNAT

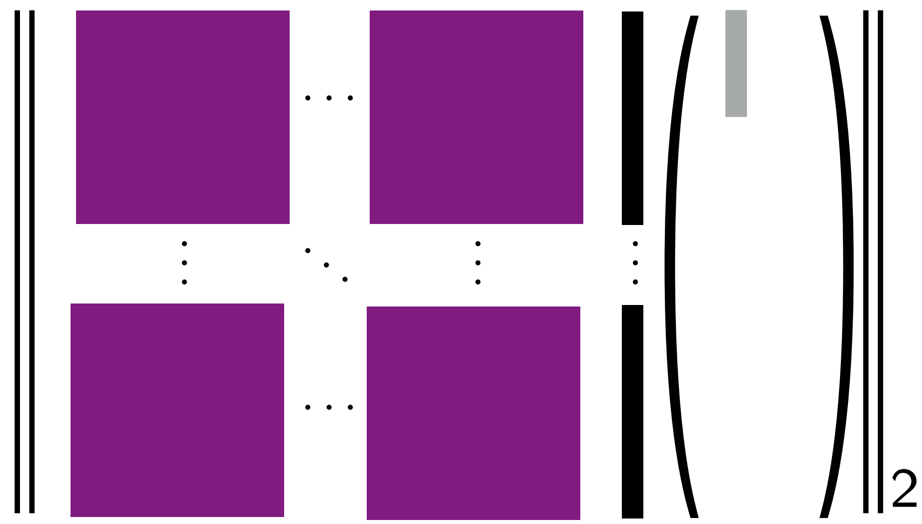
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \left\| (\mathbf{P}\boldsymbol{\Phi}_\mathbf{r})^+\mathbf{P} \, \mathbf{r}^n(\boldsymbol{\Phi}\hat{\mathbf{v}}) \right\|_2$$



+ Residual computed at a few spatial degrees of freedom

ST-LSPG hyper-reduction

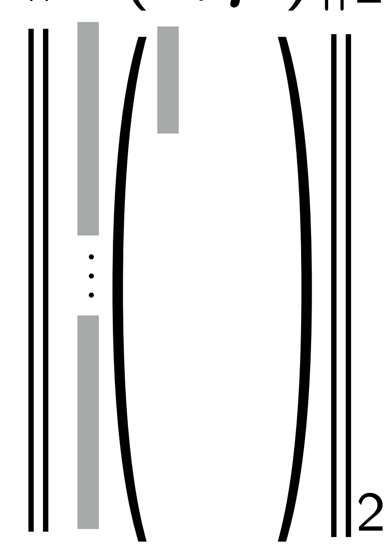
$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \, \| \bar{\mathbf{A}} \, \bar{\mathbf{r}}(\hat{\mathbf{v}};\boldsymbol{\mu}) \|_2$$



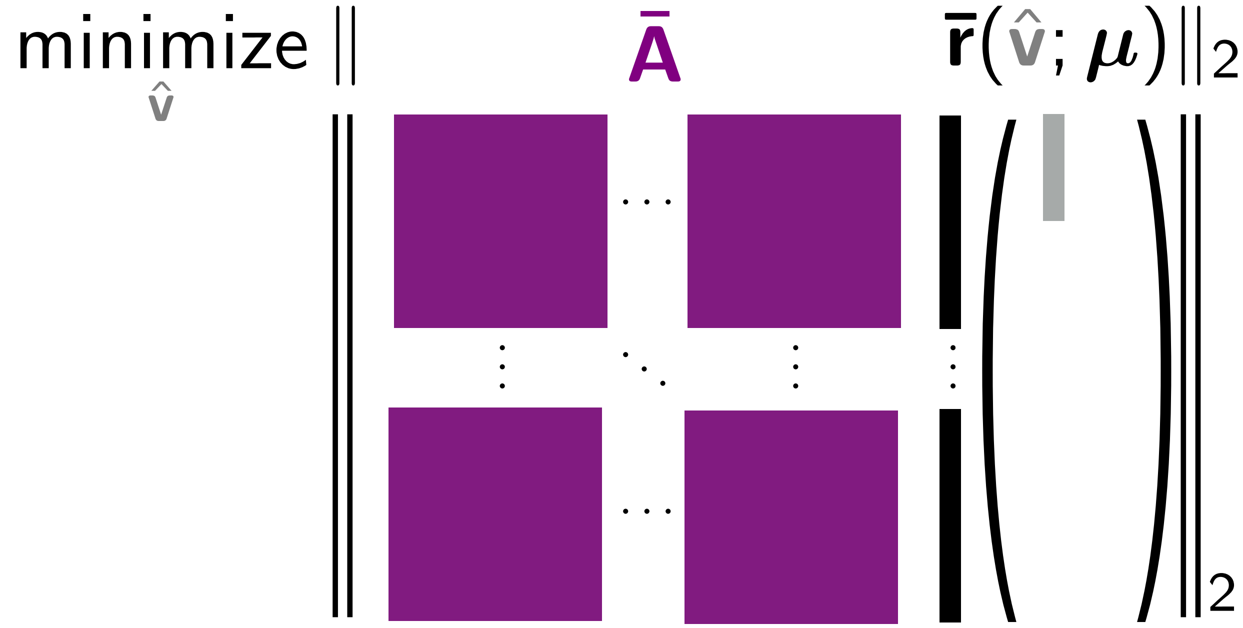
$$\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} = \bar{\boldsymbol{\Phi}}_r (\bar{\mathbf{P}} \bar{\boldsymbol{\Phi}}_r)^{+} \bar{\mathbf{P}} \bar{\mathbf{r}}$$

- space–time residual basis $\bar{\boldsymbol{\Phi}}_r$ via tensor decomposition
- space–time sampling $\bar{\mathbf{P}}$ via sequential greedy

$$\underset{\hat{\mathbf{v}}}{\text{minimize}} \, \| \tilde{\mathbf{r}}(\hat{\mathbf{v}};\boldsymbol{\mu}) \|_2$$



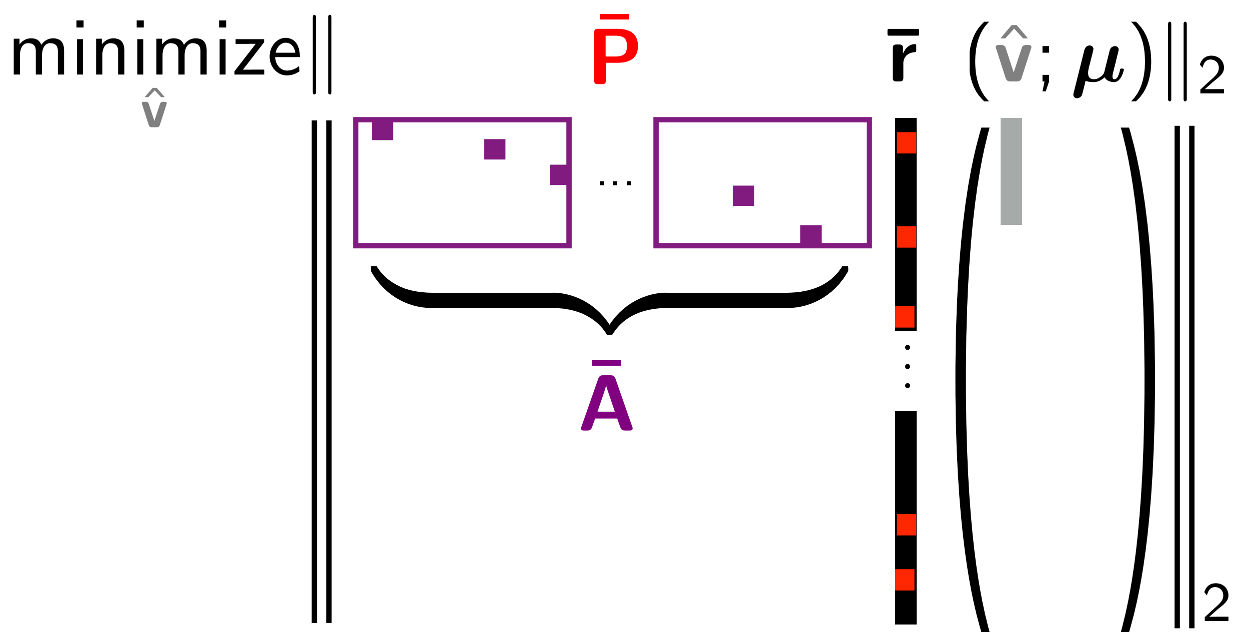
ST-LSPG hyper-reduction



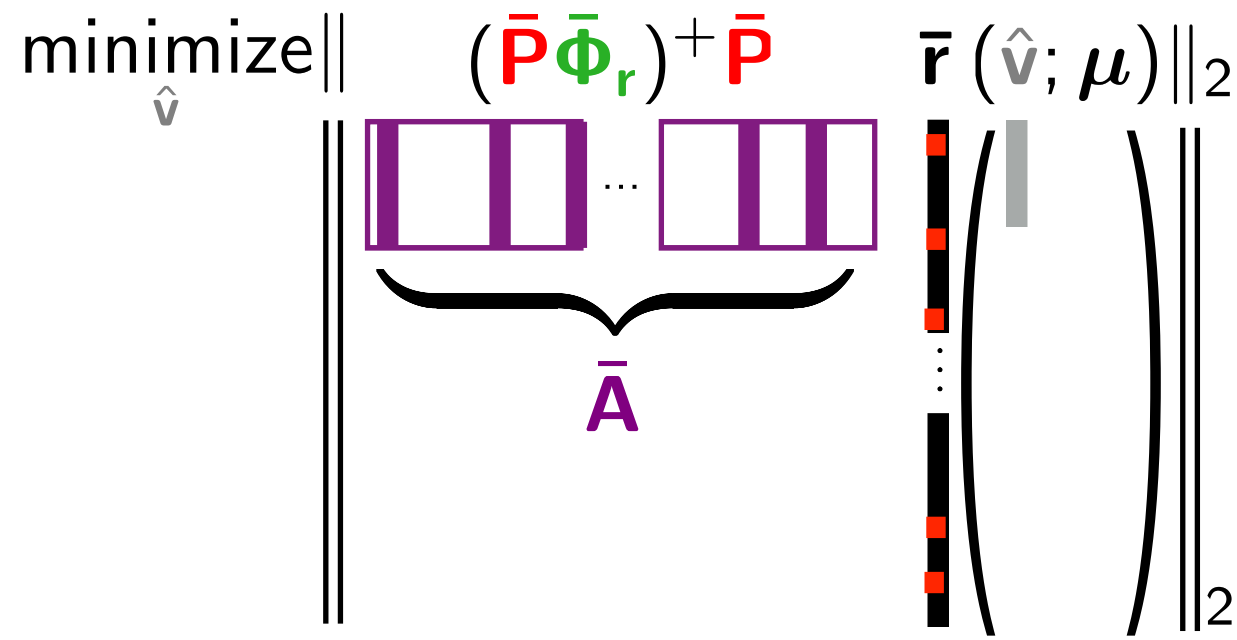
$$\bar{r} \approx \tilde{r} = \bar{\Phi}_r(\bar{P}\bar{\Phi}_r)^+\bar{P}\bar{r}$$

- space–time residual basis $\bar{\Phi}_r$ via tensor decomposition
- space–time sampling $\bar{P}$ via sequential greedy

ST-LSPG-collocation



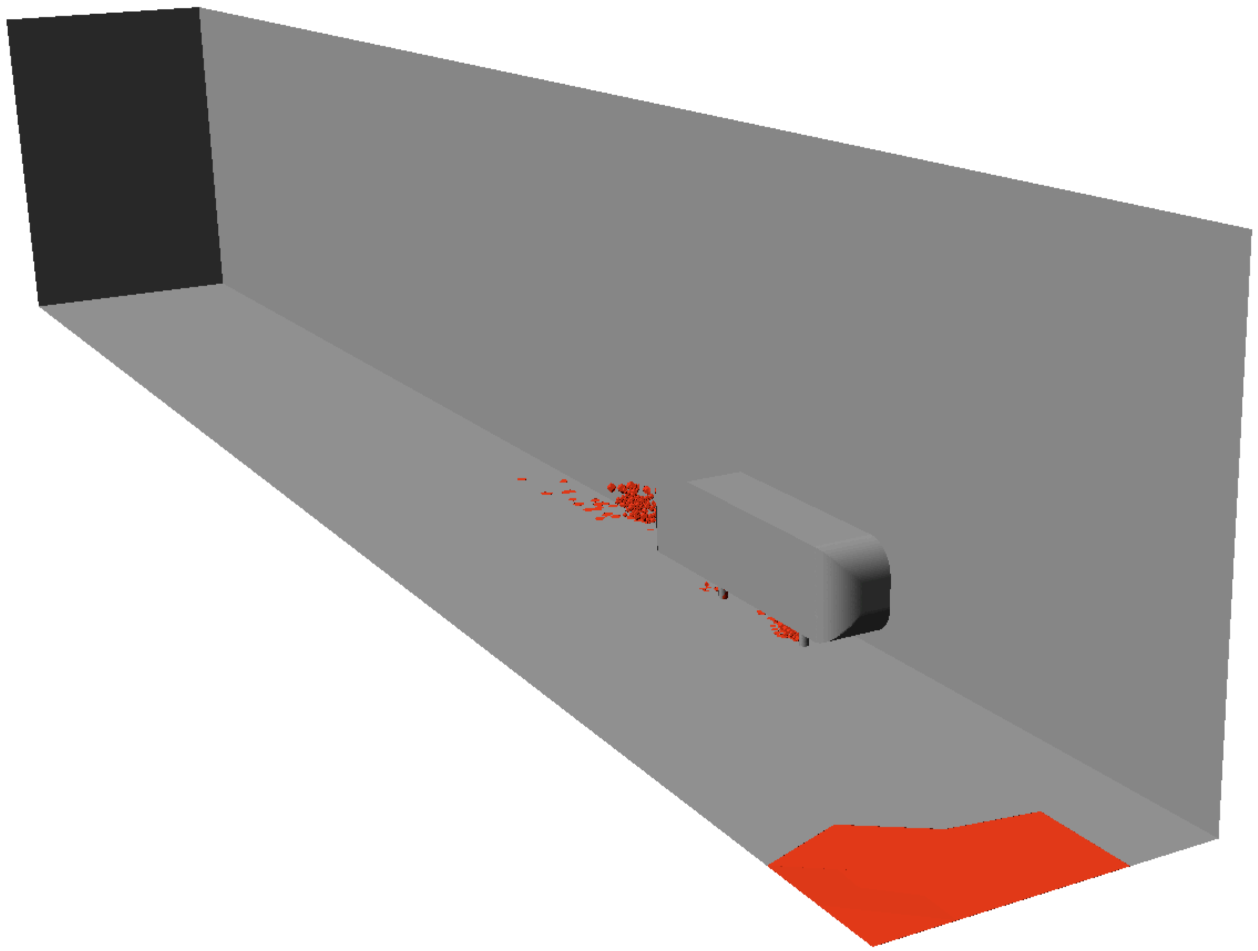
ST-GNAT



+ Residual computed at a few space–time degrees of freedom

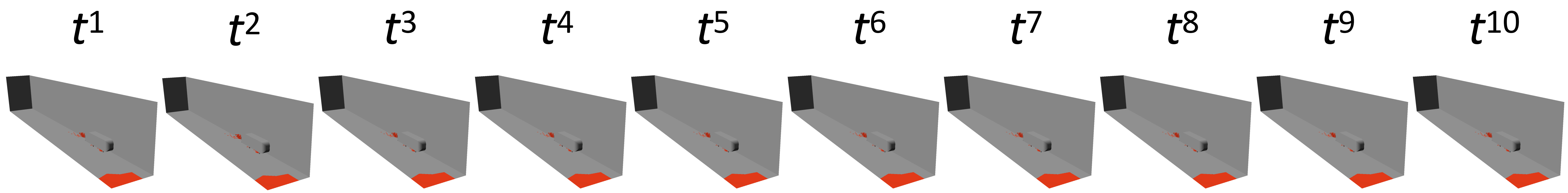
Sample mesh

LSPG



Sample mesh

LSPG

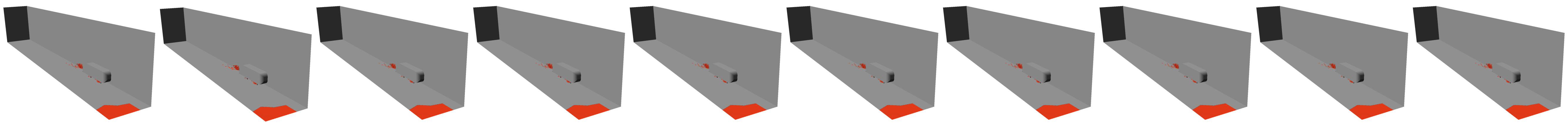


+ Residual computed at a few spatial degrees of freedom
- Residual computed at all time instances

Sample mesh

LSPG

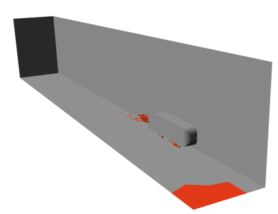
t^1 t^2 t^3 t^4 t^5 t^6 t^7 t^8 t^9 t^{10}



+ Residual computed at a few spatial degrees of freedom
- Residual computed at all time instances

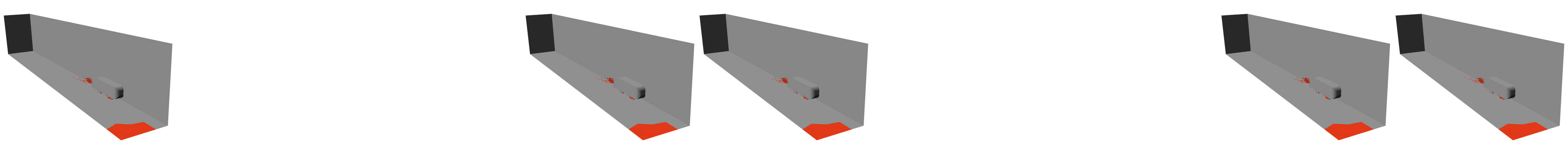
ST-LSPG

- $\bar{\mathbf{P}}$: Kronecker product of space sampling and time sampling



t^1, t^5, t^9

t^1 t^4 t^5 t^8 t^9



+ Residual computed at a few space—time degrees of freedom

Error bound

LSPG

- *Sequential solves*: sequential accumulation of time-local errors

$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n_{\mathrm{LSPG}}\|_2 \leq \frac{\gamma_1(\gamma_2)^n \exp(\gamma_3 t^n)}{\gamma_4 + \gamma_5 \Delta t} \underbrace{\max_{j\in\{1,\ldots,n\}} \min_{\hat{\mathbf{v}}} \|\mathbf{r}^j_{\mathrm{LSPG}}(\boldsymbol{\Phi}\hat{\mathbf{v}})\|_2}_{\text{worst best time-local approximation residual}}$$

- *Stability constant*: exponential time growth
- bounded by the worst (over time) best residual

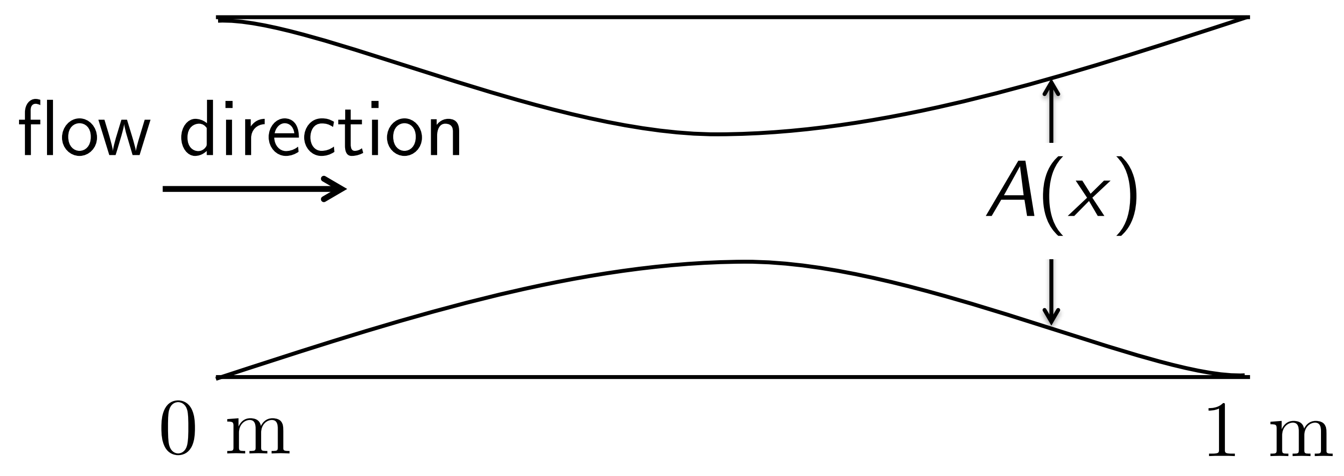
ST-LSPG

+ *Single solve*: no sequential error accumulation

$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n_{\text{ST-LSPG}}\|_2 \leq \sqrt{T}(1+\Lambda) \underbrace{\min_{\mathbf{w}\in\mathcal{ST}} \max_{j\in\{1,\ldots,T\}} \|\mathbf{x}^n - \mathbf{w}^n\|_2}_{\text{best space-time approximation error}}$$

+ *Stability constant*: polynomial growth in time with degree 3/2
+ bounded by best space–time approximation error

Quasi-1D Euler equation



$$\frac{\partial \mathbf{w}}{\partial t} + \frac{1}{A}\frac{\partial(\mathbf{f}(\mathbf{w})A)}{\partial x} = \mathbf{q}(\mathbf{w}), \quad \forall x \in [0, 1] \text{ m}, \quad \forall t \in [0, T_{\text{final}} = 0.6 \text{ sec}]$$

- Shock placed at x = 0.85 m
- Exit pressure increased by factor P_{exit}

Spatial discretization

- 1st-order finite volume (Roe)
- $\Delta x = 2 \times 10^{-2}$ m

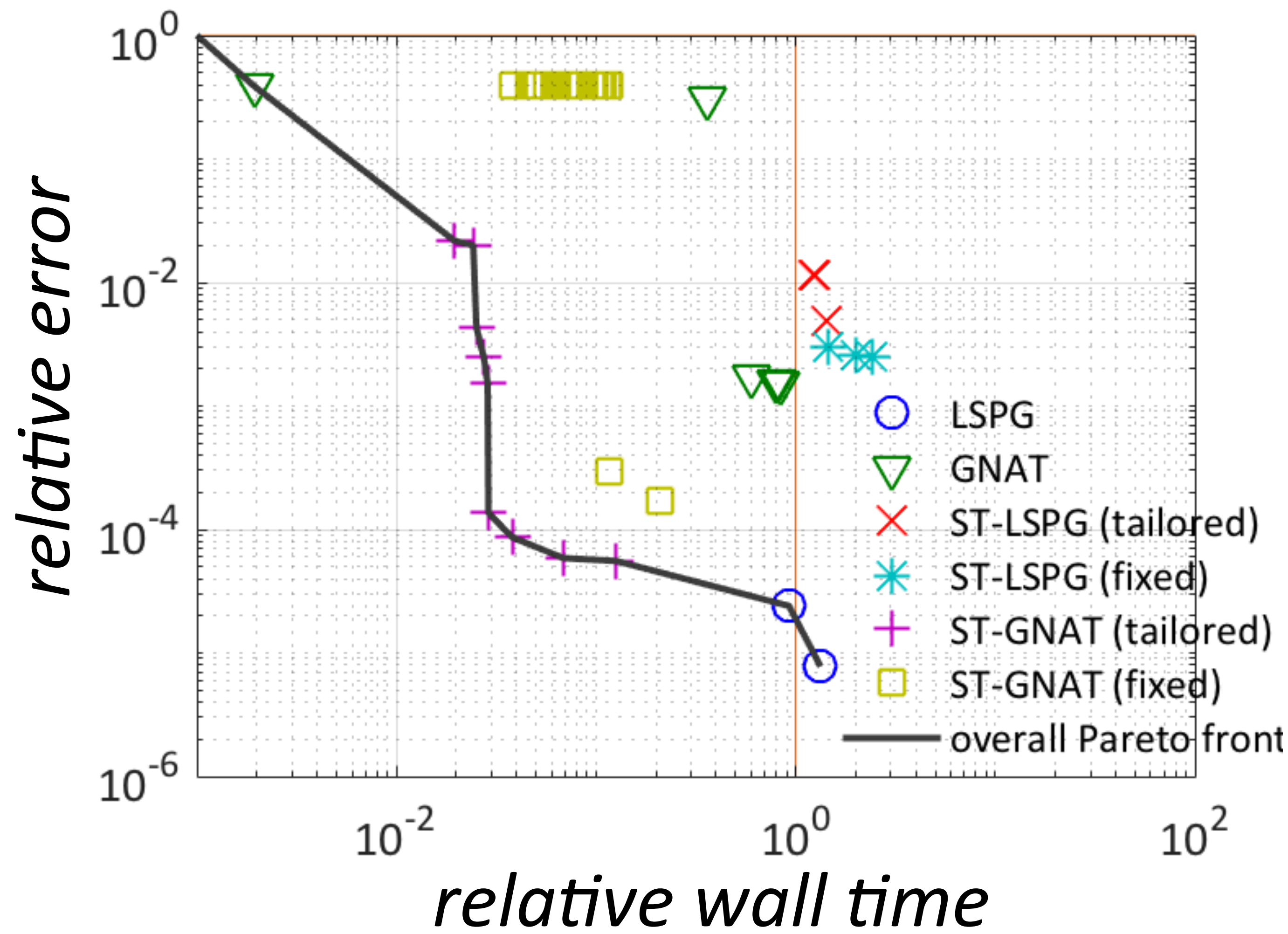
Temporal discretization

- 1st-order backward Euler
- Time step $\Delta t = 1 \times 10^{-3}$ s

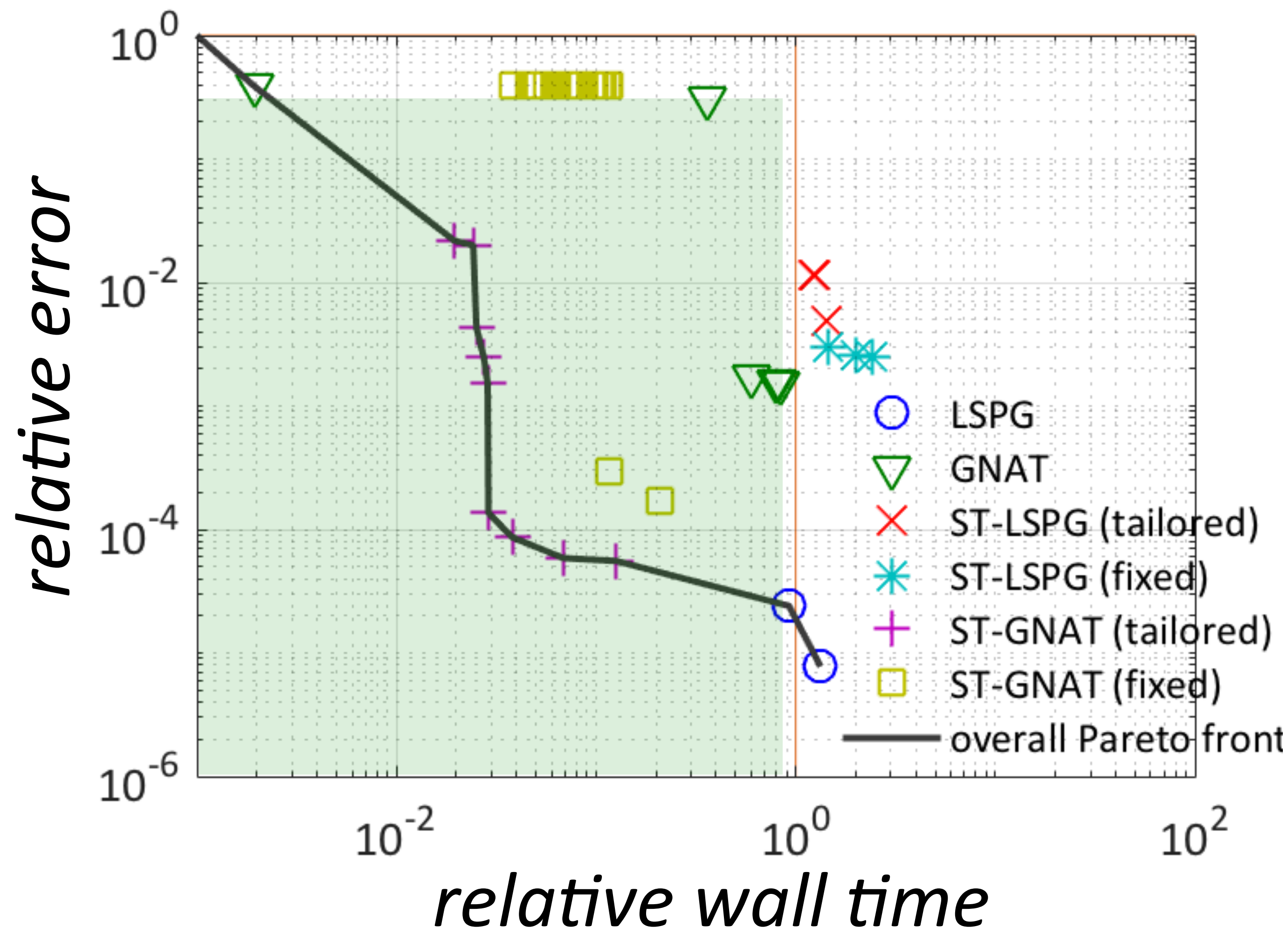
- space–time dimension $NT = 90,000$

- **Parameters**: $\mu_1 =$ middle Mach number, $\mu_2 =$ Exit-pressure factor P_{exit}
- **Offline training**: $|\mathcal{D}_{\text{train}}| = 8$

Performance Pareto front

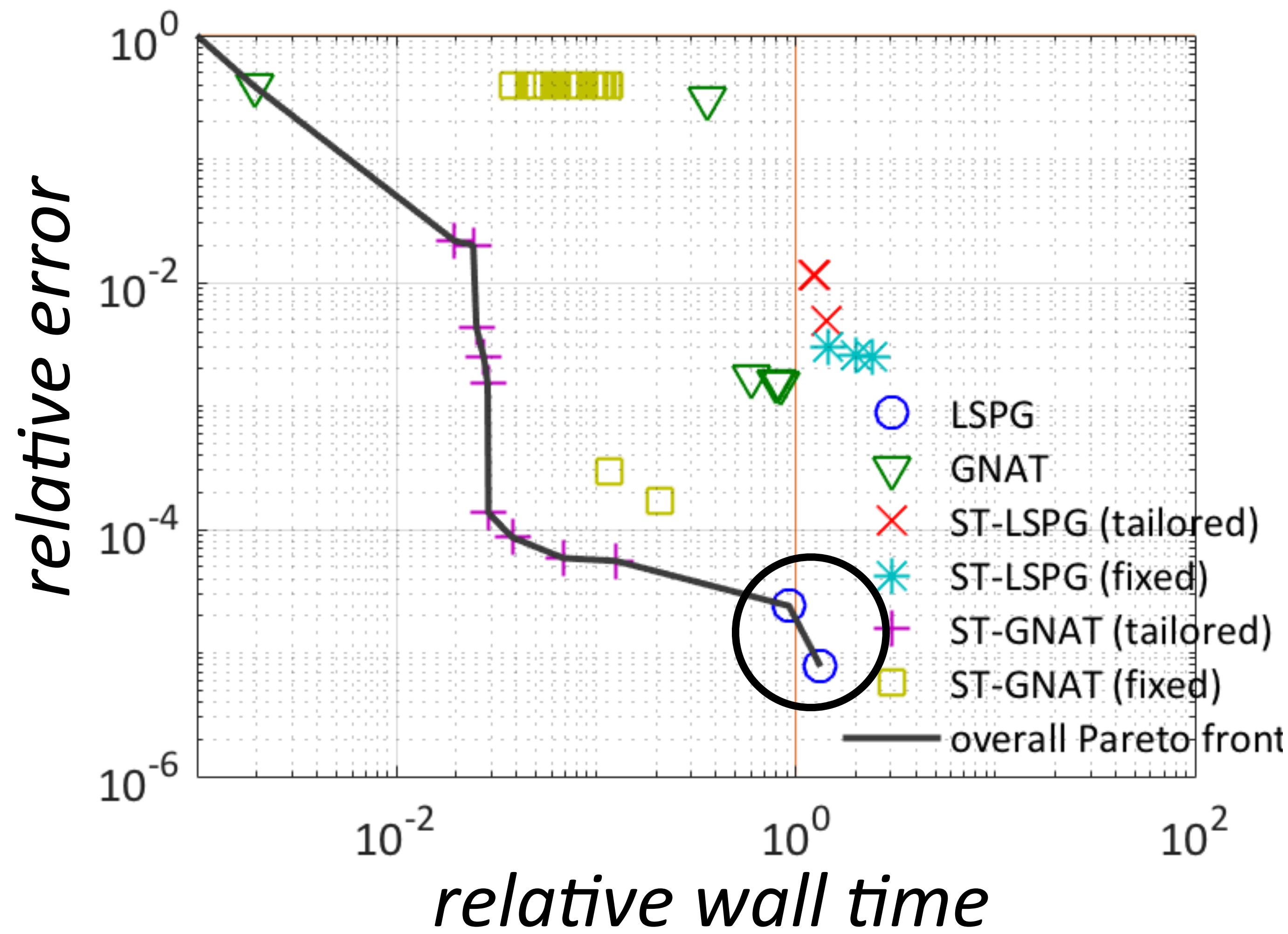


Performance Pareto front



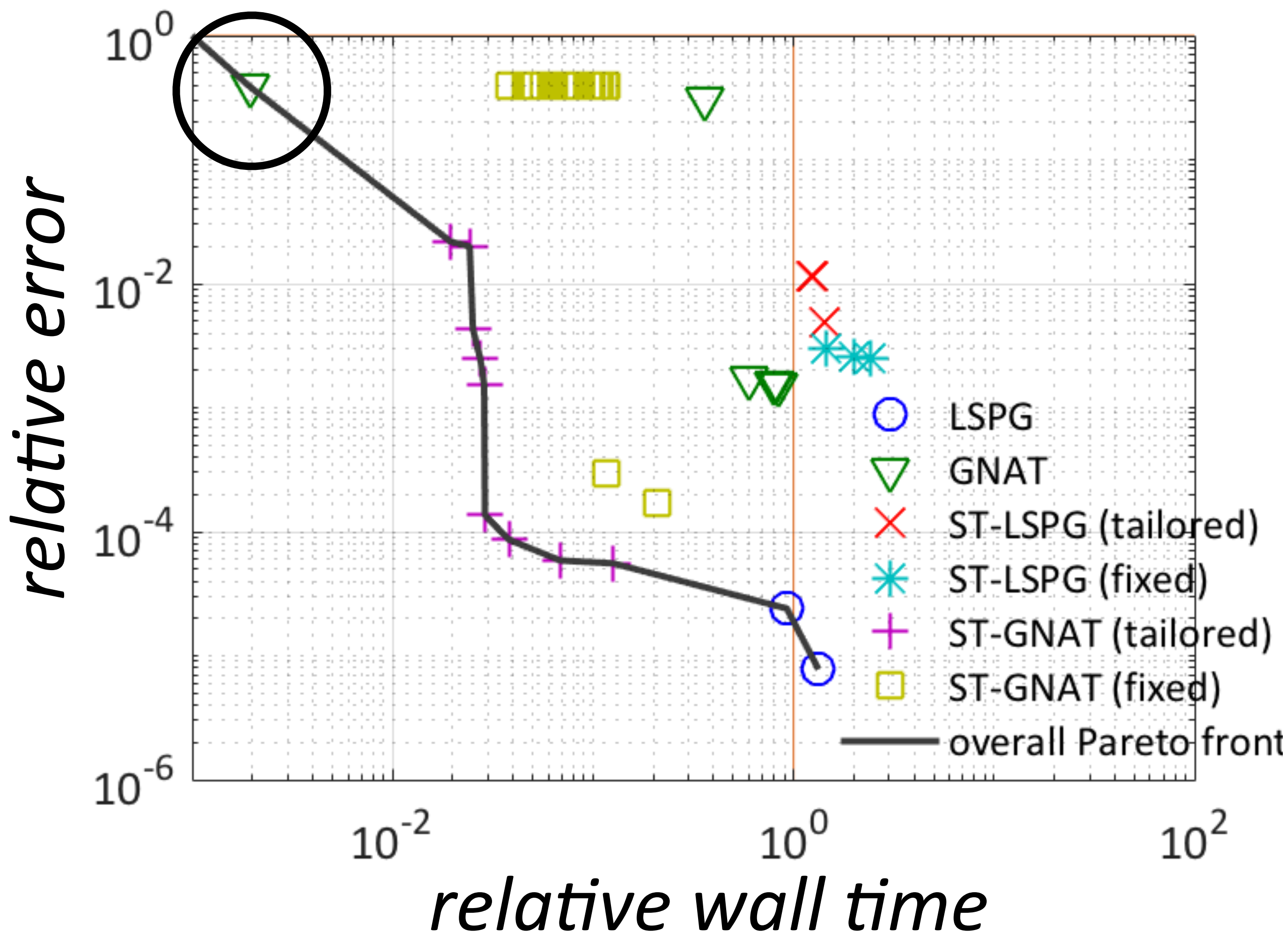
+ *ST-GNAT (tailored):* Pareto optimal for <35% rel errors, <90% rel wall time

Performance Pareto front



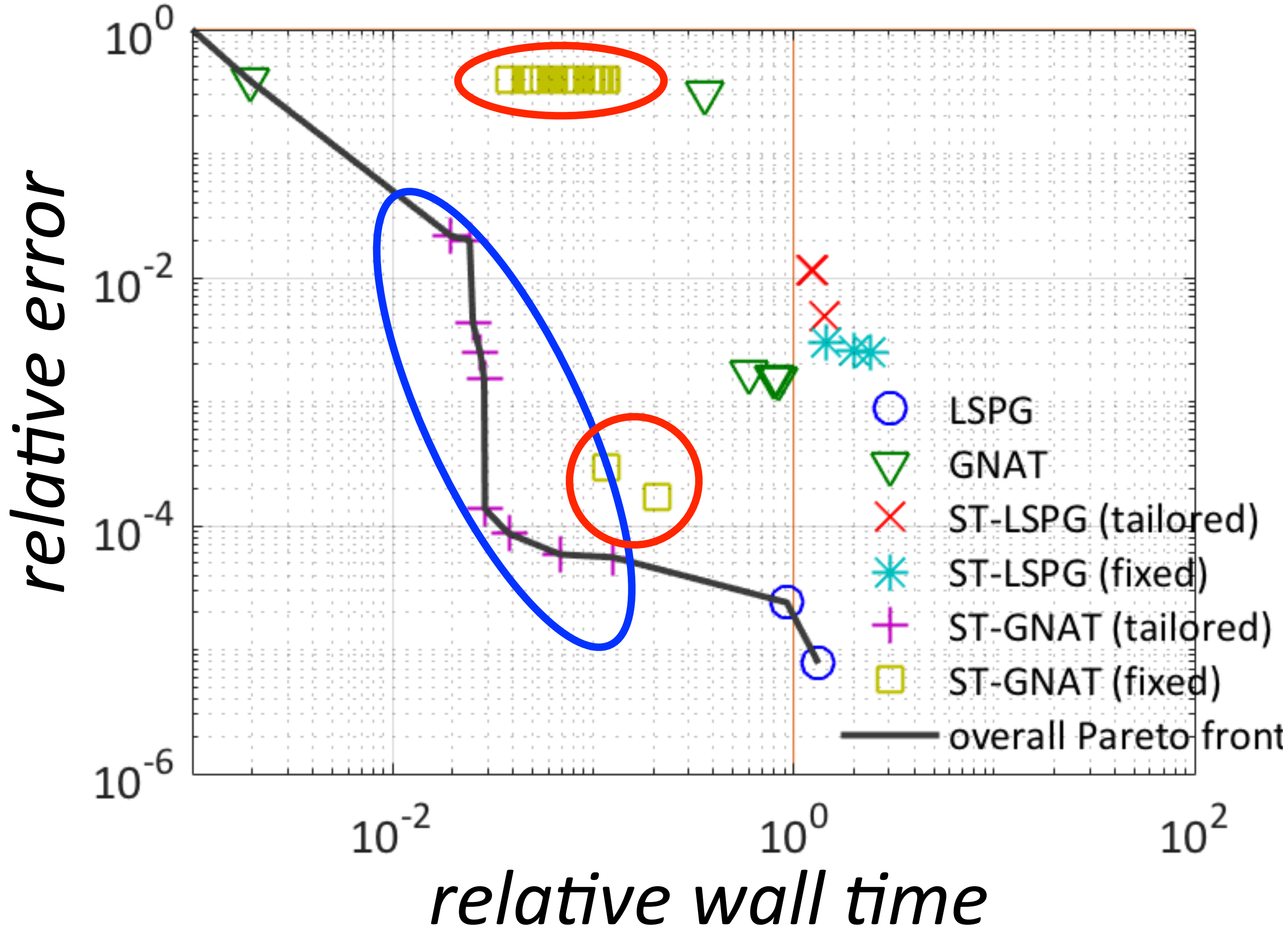
+ *ST-GNAT (tailored):* Pareto optimal for <35% rel errors, <90% rel wall time
- *LSPG*: can produce smaller errors, but incurs >90% relative wall time

Performance Pareto front



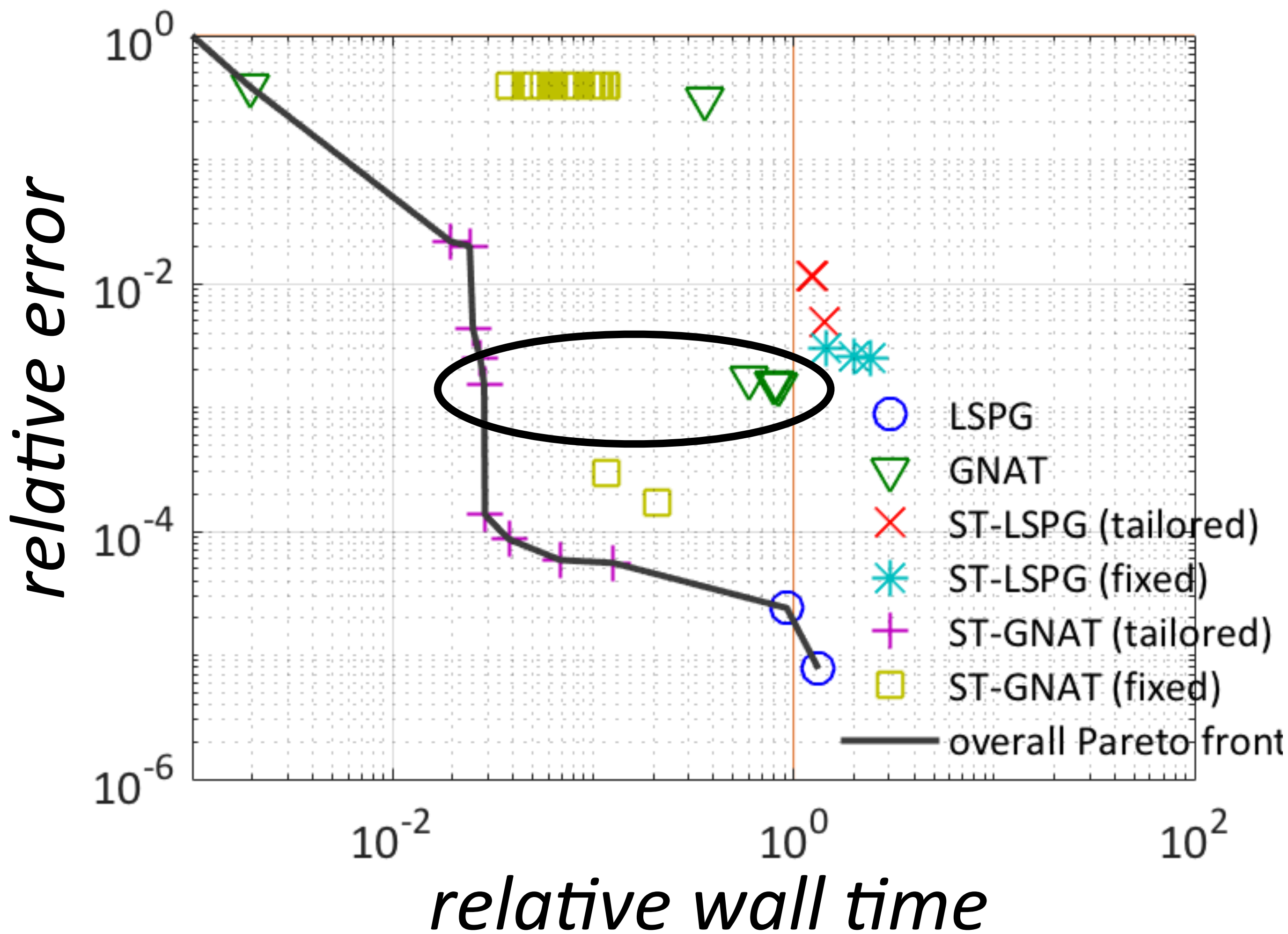
+ *ST-GNAT (tailored):* Pareto optimal for <35% rel errors, <90% rel wall time
- *LSPG*: can produce smaller errors, but incurs >90% relative wall time
- *GNAT:* can produce smaller wall times, but incurs >35% relative error

Performance Pareto front



- *ST-GNAT (tailored):* Pareto optimal for <35% rel errors, <90% rel wall time
- *LSPG*: can produce smaller errors, but incurs >90% relative wall time
- *GNAT:* can produce smaller wall times, but incurs >35% relative error
- Tailored temporal modes significantly outperform fixed temporal modes

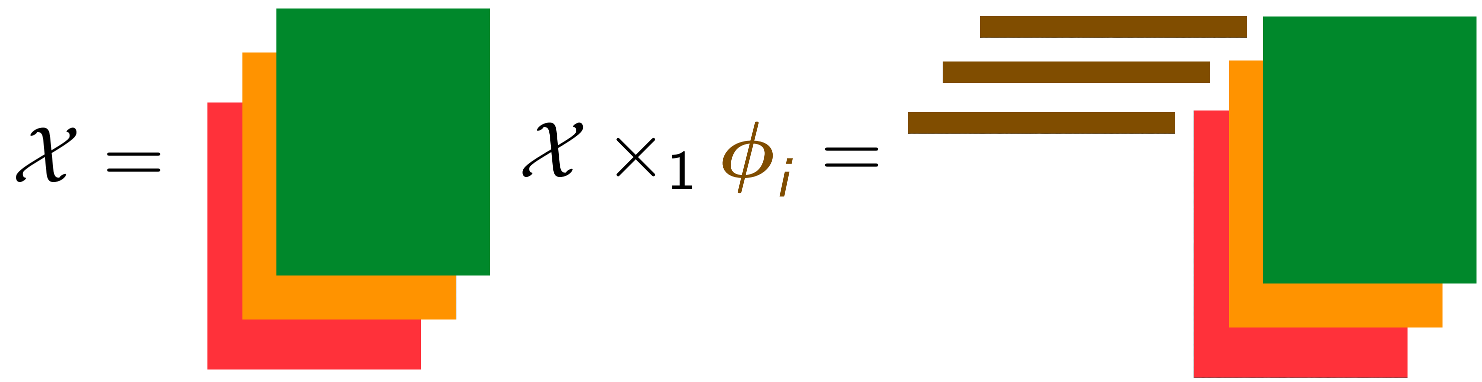
Performance Pareto front



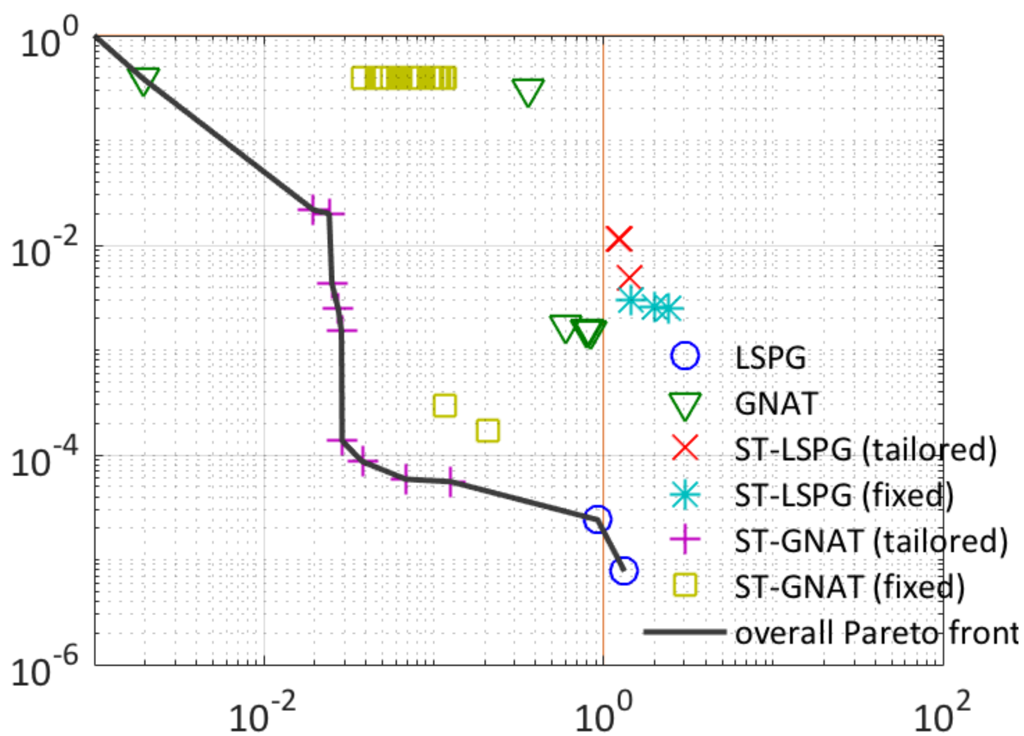
+ *ST-GNAT (tailored):* Pareto optimal for <35% rel errors, <90% rel wall time
  - *LSPG*: can produce smaller errors, but incurs >90% relative wall time
  - *GNAT:* can produce smaller wall times, but incurs >35% relative error
  - Tailored temporal modes significantly outperform fixed temporal modes
+ For fixed error, ST-GNAT (tailored) almost 100X faster than GNAT

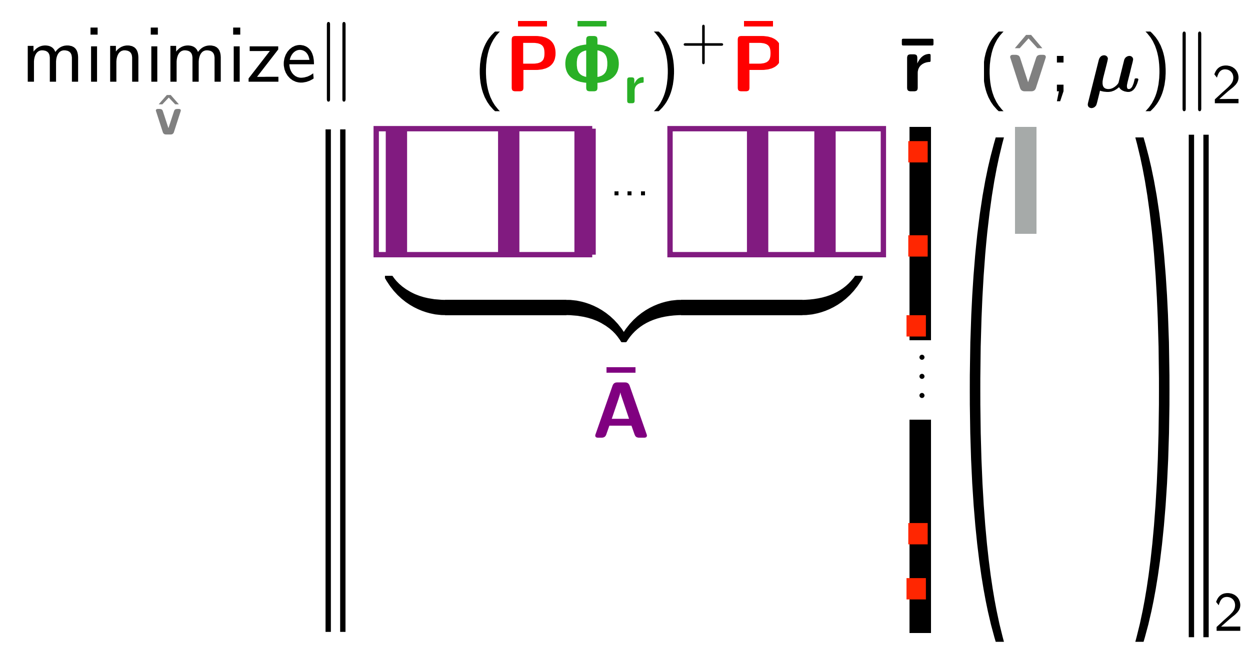
Questions?

Reference: Choi and C. Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction. *arXiv e-print*, (1703.04560), 2017.



$$\|\mathbf{x}^n - \boldsymbol{\Phi}\hat{\mathbf{x}}^n_{\text{ST-LSPG}}\|_2 \leq \sqrt{T}(1+\Lambda)\times \min_{\mathbf{w}\in\mathcal{ST}} \max_{j\in\{1,\ldots,T\}} \|\mathbf{x}^n - \mathbf{w}^n\|_2$$





Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344.