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

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

\[
\begin{align*}
\text{ODE:} & \quad \frac{dx}{dt} = f(x; t, \mu); \quad x(0, \mu) = x^0(\mu), \quad t \in [0, T_{\text{final}}], \quad \mu \in \mathcal{D} \\
\text{O\Delta E:} & \quad r^n(x^n, \ldots, x^{n-k}; \mu) = 0, \quad n = 1, \ldots, T, \quad \mu \in \mathcal{D}
\end{align*}
\]
Motivation

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

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

**ODE:** \[ r^n(x^n, \ldots, x^{n-k}; \mu) = 0, \quad n = 1, \ldots, T, \quad \mu \in \mathcal{D} \]

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

$$OΔE: \quad r^n(x^n, \ldots, x^{n-k}; \mu) = 0, \quad n = 1, \ldots, T$$
Offline step 1: data collection

\[ \mathcal{OAE}: \quad r^n(x^n, \ldots, x^{n-k}; \mu) = 0, \quad n = 1, \ldots, T \]
Offline step 2: Tensor decomposition (POD)

\[ \text{OΔE: } r^n(x^n, \ldots, x^{n-k}; \mu) = 0, \quad n = 1, \ldots, T \]

Compute dominant left singular vectors of mode-1 unfolding

\[ \mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^T \]
Offline step 2: Tensor decomposition (POD)

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

Compute dominant left singular vectors of mode-1 unfolding

$$\mathbf{X} = \mathbf{X}_{(1)} = \Phi \mathbf{U} \Sigma \mathbf{V}^T$$

\(\Phi\) columns are principal components of the spatial simulation data
Online: LSPG projection [C., Bou-Mosleh, Farhat, 2011]

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

1. Reduce number of spatial unknowns

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

2. Minimize O\Delta E residual

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

LSPG O\Delta E:

$$\hat{x}^n = \arg\min_{\hat{v}} \left\| A r^n(x^0 + \Phi \hat{v}, \tilde{x}^{n-1}, \ldots, \tilde{x}^{n-k}; \mu) \right\|_2$$
Ahmed body [Ahmed, Ramm, Faitin, 1984]

- Unsteady Navier–Stokes
- \( \text{Re} = 4.3 \times 10^6 \)
- \( M_\infty = 0.175 \)

Spatial discretization
- 2nd-order finite volume
- DES turbulence model
- 1.7 \( \times \) 10^7 degrees of freedom

Temporal discretization
- 2nd-order BDF
- Time step \( \Delta t = 8 \times 10^{-5} \text{s} \)
- 1.3 \( \times \) 10^3 time instances
Ahmed body results [C., Farhat, Cortial, Amsallem, 2013]

GNAT ROM \( (A = (P \Phi_r)^+ P) \)
- 4 hours, 4 cores
  - spatial dim: 283
  - temporal dim: \(1.3 \times 10^3\)

high-fidelity model
- 13 hours, 512 cores
  - spatial dim: \(1.7 \times 10^7\)
  - temporal dim: \(1.3 \times 10^3\)

+ 438X computational-cost reduction
+ 60,500X spatial-dimension reduction
- Zero temporal-dimension reduction
B61 captive carry

\( V_\infty \)

- Unsteady Navier–Stokes
  - \( \text{Re} = 6.3 \times 10^6 \)
  - \( M_\infty = 0.6 \)

**Spatial discretization**
- 2nd-order finite volume
- DES turbulence model
- \( 1.2 \times 10^6 \) degrees of freedom

**Temporal discretization**
- 2nd-order BDF
- Verified time step \( \Delta t = 1.5 \times 10^{-3} \)
- \( 8.3 \times 10^3 \) 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

+ 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

- 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

- Requires space–time finite element discretization
- No hyper-reduction
  - Only one space–time basis vector per training simulation

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*

Spatial v. spatiotemporal trial subspaces

**Full-order-model trial subspace**

$$[x^1 \ldots x^T] \in \mathbb{R}^N \otimes \mathbb{R}^T$$

**Spatial trial subspace**

$$[\tilde{x}^1 \ldots \tilde{x}^T] = \Phi [\hat{x}^1 \ldots \hat{x}^T] \in S \otimes \mathbb{R}^T \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

- Spatial dimension reduced
- Temporal dimension large

**Space–time trial subspace**

$$[\tilde{x}^1 \ldots \tilde{x}^T] = \sum_{i=1}^{n_{st}} \pi_i \hat{x}_i(\mu) \in ST \subseteq \mathbb{R}^N \otimes \mathbb{R}^T$$

- Spatial dimension reduced
- Temporal dimension reduced
- Additional approximation

How to compute space–time bases $\pi_i$?
Space–time basis computation

Tensor slices

\[ \mathbf{\pi}_i = \mathbf{X}_i \]

+ 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

\[
\chi = X_{(2)} = \Xi \Sigma V^T
\]

\(\Xi\) columns are principal components of the **temporal** simulation data

\[\pi_{\mathcal{F}(i,j)} = \phi_i \otimes \xi_j\]

- 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]

\[
\mathbf{X} = \mathbf{X}(\phi_i) := \mathbf{X} \times_1 \phi_i = \mathbf{U} \Sigma \mathbf{V}^T
\]

\[
\mathbf{X}(\phi_i)_{(2)} = \Xi_i
\]

\[
\pi_{\mathcal{J}(i,j)} = \phi_i \otimes \xi_j
\]

\[
\Xi_i \text{ columns are principal components of the temporal simulation data of } \phi_i
\]

- 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**

\[
\text{minimize} \quad \| \begin{pmatrix} \hat{v} \\ \bar{A} \end{pmatrix} \begin{pmatrix} r^n(\Phi \hat{v}, \tilde{x}^{n-1}, \ldots, \tilde{x}^{n-k}; \mu) \end{pmatrix} \|_2, \quad n = 1, \ldots, T
\]

+ efficient: time-sequential solve

**ST-LSPG**

\[
\bar{r}(\hat{v}; \mu) := \begin{bmatrix} r^1(\sum_{i=1}^{n_{st}} \pi_i(t^1) \hat{v}_i, \sum_{i=1}^{n_{st}} \pi_i(t^0) \hat{v}_i; \mu) \\ \vdots \\ r^T(\sum_{i=1}^{n_{st}} \pi_i(t^T) \hat{v}_i, \sum_{i=1}^{n_{st}} \pi_i(t^{T-1}) \hat{v}_i, \ldots, \sum_{i=1}^{n_{st}} \pi_i(t^{T-k}) \hat{v}_i; \mu) \end{bmatrix}
\]

\[
\text{minimize} \quad \| \begin{pmatrix} \hat{v} \\ \bar{A} \end{pmatrix} \begin{pmatrix} \bar{r}(\hat{v}; \mu) \end{pmatrix} \|_2
\]

- costly: minimizing residual simultaneously over space and time
LSPG hyper-reduction

Select $A$ to make this less expensive

$$r^n \approx \tilde{r}^n = \Phi_r (P\Phi_r)^+ P r^n$$

residual element $r^n_i$

vector index

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

```
minimize $\| \tilde{r}^n (\Phi \hat{v}) \|_2$
```
LSPG hyper-reduction

minimize \[ \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} r^n(\Phi \hat{\mathbf{x}}, \tilde{x}^{n-1}, \ldots, \tilde{x}^{n-k}; \mu) \end{bmatrix}^2 \]

Select A to make this less expensive
\[ r^n \approx \tilde{r}^n = \Phi_r(P\Phi_r)^+Pr^n \]

residual element \( r^n_i \)

vector index

LSPG-collocation

minimize \[ \begin{bmatrix} P \end{bmatrix} \begin{bmatrix} r^n(\Phi \hat{\mathbf{x}}) \end{bmatrix}^2 \]

GNAT

minimize \[ \begin{bmatrix} (P\Phi_r)^+P \end{bmatrix} \begin{bmatrix} r^n(\Phi \hat{\mathbf{x}}) \end{bmatrix}^2 \]

+ Residual computed at a few spatial degrees of freedom
ST-LSPG hyper-reduction

\[
\begin{align*}
\text{minimize } & \| \begin{array}{ccc}
\hat{\mathbf{v}} \\
\bar{\mathbf{A}} \\
\bar{\mathbf{r}}(\hat{\mathbf{v}}; \mu)
\end{array} \|_2 \\
\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} &= \Phi_r (\bar{\mathbf{P}} \Phi_r)^+ \bar{\mathbf{P}} \tilde{\mathbf{r}}
\end{align*}
\]

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

\[
\begin{align*}
\text{minimize } & \| \begin{array}{ccc}
\hat{\mathbf{v}} \\
\bar{\mathbf{r}}(\hat{\mathbf{v}}; \mu)
\end{array} \|_2 \\
\bar{\mathbf{r}} \approx \tilde{\mathbf{r}} &= \Phi_r (\bar{\mathbf{P}} \Phi_r)^+ \bar{\mathbf{P}} \tilde{\mathbf{r}}
\end{align*}
\]
ST-LSPG hyper-reduction

\[
\begin{align*}
\min_{\tilde{\mathbf{v}}} & \quad \|\tilde{\mathbf{A}} \tilde{\mathbf{v}} - \tilde{\mathbf{r}}(\tilde{\mathbf{v}}; \mu)\|_2^2 \\
\tilde{\mathbf{r}} & \approx \tilde{\mathbf{r}} = \Phi_r (\tilde{\mathbf{P}} \Phi_r)^+ \tilde{\mathbf{P}} \tilde{\mathbf{r}}
\end{align*}
\]

- space–time residual basis $\Phi_r$ via tensor decomposition
- space–time sampling $\tilde{\mathbf{P}}$ via sequential greedy

**ST-LSPG-collocation**

\[
\min_{\tilde{\mathbf{v}}} \|\tilde{\mathbf{P}} \tilde{\mathbf{v}} - \tilde{\mathbf{r}}(\tilde{\mathbf{v}}; \mu)\|_2^2
\]

**ST-GNAT**

\[
\min_{\tilde{\mathbf{v}}} \|\tilde{\mathbf{P}} \Phi_r(\tilde{\mathbf{v}}; \mu)\|_2^2
\]

\[
\min_{\tilde{\mathbf{v}}} \|\tilde{\mathbf{P}} \Phi_r(\tilde{\mathbf{v}}; \mu)\|_2^2
\]

+ 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

\( t^1 \quad t^2 \quad t^3 \quad t^4 \quad t^5 \quad t^6 \quad t^7 \quad t^8 \quad t^9 \quad t^{10} \)

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

**ST-LSPG**

\( \mathbf{P} \): Kronecker product of **space sampling and time sampling**

\( t^1, t^5, t^9 \)

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

**LSPG**

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

\[
\|x^n - \Phi \hat{x}_{LSPG}^n\|_2 \leq \frac{\gamma_1 (\gamma_2)^n \exp(\gamma_3 t^n)}{\gamma_4 + \gamma_5 \Delta t} \max_{j \in \{1, \ldots, n\}} \min_{\hat{v}} \|r_j^{LSPG}(\Phi \hat{v})\|_2
\]

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

**ST-LSPG**

+ **Single solve**: no sequential error accumulation

\[
\|x^n - \Phi \hat{x}_{ST-LSPG}^n\|_2 \leq \sqrt{T}(1 + \Lambda) \min_{w \in S} \max_{j \in \{1, \ldots, T\}} \|x^n - w^n\|_2
\]

+ **Stability constant**: polynomial growth in time with degree 3/2
+ bounded by best space–time approximation error
Quasi-1D Euler equation

![Flow direction diagram](image)

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

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

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

Temporal discretization
- 1st-order backward Euler
- Time step \( \Delta t = 1 \times 10^{-3} \text{ s} \)
- space–time dimension \( NT = 90,000 \)

Parameters:
- \( \mu_1 = \) middle Mach number
- \( \mu_2 = \) Exit-pressure factor \( P_{\text{exit}} \)

Offline training:
- \( |\mathcal{D}_{\text{train}}| = 8 \)
Performance Pareto front

![Graph showing relative error vs. relative wall time for different methods: LSPG, GNAT, ST-LSPG (tailored), ST-LSPG (fixed), ST-GNAT (tailored), ST-GNAT (fixed). The overall Pareto front is marked by a solid line.](image)
Performance Pareto front

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


\[
\mathcal{X} = \mathcal{X} \times_1 \phi_i = \begin{bmatrix} \bar{r} \\ \tilde{\mathcal{P}} \bar{P} \phi_r \end{bmatrix}
\]

\[
\| x^n - \Phi \hat{x}^n_{\text{ST-LSPG}} \|_2 \leq \sqrt{T(1 + \lambda)} \times \min_{w \in S} \max_{j \in \{1, \ldots, T\}} \| x^n - w^n \|_2
\]

\[
\min_{\hat{\sigma}} \| (\bar{P} \phi_r)^+ \bar{P} \hat{\sigma} - \tilde{\mathcal{A}} \|_2
\]