AA216/CME345: PROJECTION-BASED MODEL ORDER REDUCTION

Nonlinear Projection-Based Model Order Reduction

Charbel Farhat
Stanford University
cfarhat@stanford.edu

Outline

- Note: The material covered in this chapter is based on the following published documents:
  - K. Carlberg, C. Bou-Mosleh, C. Farhat. Efficient nonlinear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. International Journal for Numerical Methods in Engineering 2011; 86(2):155-181.
  - K. Carlberg, C. Farhat, J. Cortial, D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. Journal of Computational Physics 2013; 242:623-647.
  - K. Washabaugh, M. Zahr, C. Farhat, On the use of discrete nonlinear reduced-order models for the prediction of steady-state flows past parametrically deformed complex geometries. AIAA-2016-1814, AIAA SciTech 2016, San Diego, CA, January 4-8, 2016.
  - D. Amsallem, M. Zahr, C. Farhat. Nonlinear model reduction based on local reduced order bases. International Journal for Numerical Methods in Engineering 2012; 92(12):891-916.

- Note: The material covered in this chapter is based on the following published documents (continue):
  - S. Grimberg, C. Farhat, R. Tezaur, C. Bou-Mosleh. Mesh sampling and weighting for the hyperreduction of nonlinear Petrov-Galerkin reduced-order models with local reduced-order bases. International Journal for Numerical Methods in Engineering 2021; 122:1846-1874.
  - J. Barnett and C. Farhat. Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction. Journal of Computational Physics 2022; 464:111348.
  - J. Barnett, C. Farhat and Y. Maday. Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility. Journal of Computational Physics 2023; 492:112420.
  - A. Cohen, C. Farhat, Y. Maday and A. Somacal. Nonlinear compressive reduced basis approximation for PDEs. Comptes Rendus de l'Académie des Sciences - Mécanique, 2023; 351:357-374.

- Note: The material covered in this chapter is based on the following published documents (continue):
  - M. R. Chmiel, J. L. Barnett and C. Farhat. Unified LSPG model reduction framework and assessment for benchmark hypersonic CFD problem. AIAA Journal 2025; 63(1):72-90.

- Semi-discrete level (parametric dependence not emphasized)

$$\frac{d\mathbf{w}}{dt}(t) = \mathbf{f}(\mathbf{w}(t), t), \quad \mathbf{w} \in \mathbb{R}^N, \ \mathbf{f} \in \mathbb{R}^N$$

- Subspace approximation: $\mathbf{w}(t) \approx \mathbf{V}\mathbf{q}(t) \Rightarrow \mathbf{V}\frac{d\mathbf{q}}{dt}(t) \approx \mathbf{f}(\mathbf{V}\mathbf{q}(t), t)$
- Discrete level (backward Euler implicit time integration scheme)

$$\mathbf{V}\left(\frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t^n}\right) \approx \mathbf{f}\left(\mathbf{V}\mathbf{q}^{n+1}, t^{n+1}\right), \ \mathbf{V} \in \mathbb{R}^{N \times k}, \ \mathbf{q} \in \mathbb{R}^k$$

- Discrete residual

$$\mathbf{r}^{n+1}\left(\mathbf{q}^{n+1}\right) = \mathbf{V}\left(\frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{\Delta t^n}\right) - \mathbf{f}\left(\mathbf{V}\mathbf{q}^{n+1}, t^{n+1}\right) \in \mathbb{R}^N$$

- At each time step, residual minimization in the two-norm

$$\boxed{\mathbf{q}^{n+1} = \arg\min_{\mathbf{y} \in \mathbb{R}^k} \left\| \mathbf{r}^{n+1}(\mathbf{y}) \right\|_2}$$

- At each time step, nonlinear least-squares problem of the form $\min_{\mathbf{y}} \|\mathbf{r}(\mathbf{y})\|_2$, where $\mathbf{r} \in \mathbb{R}^N$, $\mathbf{y} \in \mathbb{R}^k$, and $k \ll N$
- Equivalent function to minimize: $\phi(\mathbf{y}) = \frac{1}{2}\|\mathbf{r}(\mathbf{y})\|_2^2 = \frac{1}{2}\mathbf{r}^T(\mathbf{y})\mathbf{r}(\mathbf{y})$
- Gradient: $\nabla\phi(\mathbf{y}) = \mathbf{J}_\mathbf{r}^T(\mathbf{y})\mathbf{r}(\mathbf{y})$, where $\mathbf{J}_\mathbf{r}(\mathbf{y}) = \frac{\partial \mathbf{r}}{\partial \mathbf{y}}(\mathbf{y}) \in \mathbb{R}^{N\times k}$
- Iterative solution of equivalent minimization problem using the Gauss-Newton method

$$\mathbf{y}^{(\ell+1)} = \mathbf{y}^{(\ell)} + \Delta\mathbf{y}^{(\ell+1)}$$

where

$$\nabla^2\phi\left(\mathbf{y}^{(\ell)}\right)\Delta\mathbf{y}^{(\ell+1)} = -\nabla\phi\left(\mathbf{y}^{(\ell)}\right)$$

- What is $\nabla^2\phi(\mathbf{y})$?

$$\nabla^2\phi(\mathbf{y}) = \mathbf{J}^T(\mathbf{y})\mathbf{J}(\mathbf{y}) + \sum_{i=1}^{N}\frac{\partial^2 r_i}{\partial \mathbf{y}^2}(\mathbf{y})r_i(\mathbf{y})$$

- Gauss-Newton method with $\nabla^2\phi(\mathbf{y}) \approx \mathbf{J}^T(\mathbf{y})\mathbf{J}(\mathbf{y})$

- At each time step, Gauss-Newton method can be written as

$$\mathbf{y}^{(\ell+1)} = \mathbf{y}^{(\ell)} + \Delta\mathbf{y}^{(\ell+1)}$$

where

$$\mathbf{J}_\mathbf{r}^T\left(\mathbf{y}^{(\ell)}\right)\mathbf{J}_\mathbf{r}\left(\mathbf{y}^{(\ell)}\right)\Delta\mathbf{y}^{(\ell+1)} = -\mathbf{J}_\mathbf{r}^T\left(\mathbf{y}^{(\ell)}\right)\mathbf{r}\left(\mathbf{y}^{(\ell)}\right) \tag{1}$$

- This is the normal equation for

$$\Delta\mathbf{y}^{(\ell+1)} = \frac{1}{2}\arg\min_{\mathbf{z}}\left\|\mathbf{J}_\mathbf{r}\left(\mathbf{y}^{(\ell)}\right)\mathbf{z} + \mathbf{r}\left(\mathbf{y}^{(\ell)}\right)\right\|_2^2$$

- QR decomposition of the Jacobian

$$\mathbf{J}_\mathbf{r}\left(\mathbf{y}^{(\ell)}\right) = \mathbf{Q}^{(\ell)}\mathbf{R}^{(\ell)},\ \mathbf{Q}^{(\ell)} \in \mathbb{R}^{N\times N},\ \left(\mathbf{Q}^{(\ell)}\right)^T\mathbf{Q}^{(\ell)} = \mathbf{I}_N$$
$$\mathbf{R}^{(\ell)} \in \mathbb{R}^{N\times k} \text{ upper triangular}$$

- Equivalent solution using the QR decomposition (assuming that $\mathbf{R}^{(\ell)}$ has full column rank)

$$\Delta\mathbf{y}^{(\ell+1)} = -\mathbf{J}_\mathbf{r}\left(\mathbf{y}^{(\ell)}\right)^\dagger\mathbf{r}\left(\mathbf{y}^{(\ell)}\right) = -\left(\mathbf{R}^{(\ell)}\right)^{-1}\left(\mathbf{Q}^{(\ell)}\right)^T\mathbf{r}\left(\mathbf{y}^{(\ell)}\right)$$

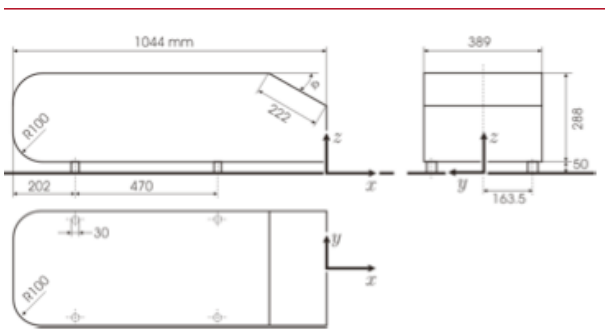
- Recall that for the backward Euler implicit time integration scheme
$$\mathbf{r}\left(\mathbf{q}^{n+1}\right) = \mathbf{V}\left(\frac{\mathbf{q}^{n+1}-\mathbf{q}^{n}}{\Delta t^{n}}\right) - \mathbf{f}\left(\mathbf{V}\mathbf{q}^{n+1}, t^{n+1}\right)$$
- For an arbitrary *implicit* time integration scheme
$\mathbf{r}\left(\mathbf{q}^{n+1}\right) = \mathbf{g}\left(\mathbf{V}\mathbf{q}^{n+1}, \mathbf{V}\mathbf{q}^{n}, \ldots, \mathbf{V}\mathbf{q}^{m}\right) - \mathbf{f}\left(\mathbf{V}\mathbf{q}^{n+1}, t^{n+1}\right), \; m < n,$
which is an approximation of $\mathbf{r}(\mathbf{q}) = \mathbf{V}\frac{d\mathbf{q}}{dt}(t) - \mathbf{f}(\mathbf{V}\mathbf{q}(t), t)$
- Recall $\tilde{\mathbf{w}} = \mathbf{V}\mathbf{q}$: From the above expression of $\mathbf{r}(\mathbf{q})$, it follows that
$$\mathbf{J}_{\mathbf{r}}(\mathbf{q}) = \frac{\partial \mathbf{r}}{\partial \mathbf{q}}(\mathbf{q}) = -\frac{\partial \mathbf{f}}{\partial \tilde{\mathbf{w}}}(\tilde{\mathbf{w}})\mathbf{V} = -\mathbf{J}_{\mathbf{f}}(\tilde{\mathbf{w}})\mathbf{V}$$
- Therefore, minimizing
$\phi(\mathbf{q}) = \frac{1}{2}\|\mathbf{r}(\mathbf{q})\|_2^2 \Leftrightarrow \mathbf{J}_{\mathbf{r}}^{T}(\mathbf{q})\mathbf{r}(\mathbf{q}) = -\left(\mathbf{J}_{\mathbf{f}}(\tilde{\mathbf{w}})\mathbf{V}\right)^{T}\mathbf{r}(\mathbf{q}) = \mathbf{0}$ is equivalent to solving the nonlinear rectangular problem
$$\mathbf{W}^{T}(\tilde{\mathbf{w}})\mathbf{r}(\tilde{\mathbf{w}}) = \mathbf{0}, \text{ where } \mathbf{W}(\tilde{\mathbf{w}}) = \mathbf{J}_{\mathbf{f}}(\tilde{\mathbf{w}})\mathbf{V} \in \mathbb{R}^{N\times k}$$
- Hence, residual minimization in the two-norm is a Petrov-Galerkin projection method

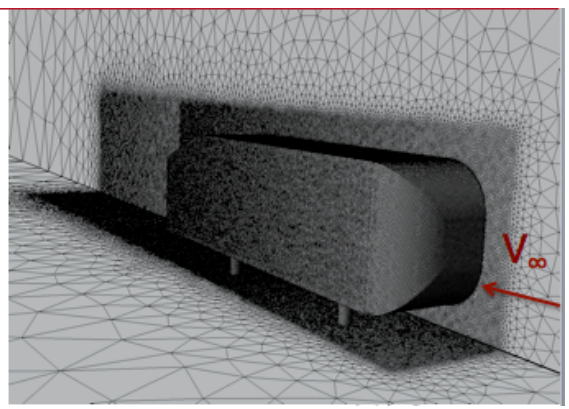
- In summary, the projection-based model order reduction (PMOR) method based on the minimization in the two-norm of the discrete residual is the Petrov-Galerkin PMOR method with $\mathbf{W}(\tilde{\mathbf{w}}) = \mathbf{J_f}(\tilde{\mathbf{w}})\mathbf{V}$
- At each time step, the solution of the resulting nonlinear rectangular problem $\mathbf{W}^T(\tilde{\mathbf{w}})\mathbf{r}(\tilde{\mathbf{w}}) = \mathbf{0}$ by the Gauss-Newton method leads to the same system of equations as (1), with

$$\mathbf{J_r}\left(\mathbf{q}^{(\ell)}\right) = \mathbf{J_f}\left(\tilde{\mathbf{w}}^{(\ell)}\right)\mathbf{V} = \mathbf{J_f}\left(\mathbf{V}\mathbf{q}^{(\ell)}\right)\mathbf{V}$$

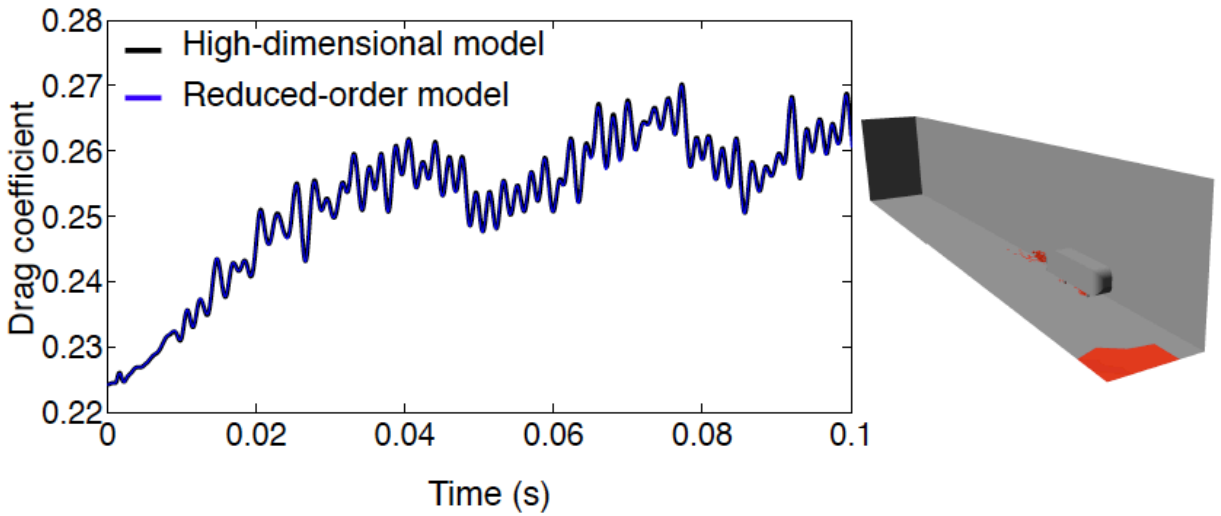
- This Petrov-Galerkin PMOR method is known today as the Least-Squares Petrov-Galerkin (LSPG) method: it represents the state of the art of PMOR for transport problems (first-order hyperbolic problems) and particularly for convection-dominated turbulent flow problems
- This is because LSPG is numerically stable for such problems, whereas the standard Galerkin projection method is unstable and thus requires stabilization
- LSPG is equally applicable to steady-state (time-independent) problems
- For computational efficiency, LSPG has been equipped in the literature with the hyperreduction methods DEIM (in which case it was called the Gauss-Newton with Approximated Tensors (GNAT) method) and ECSW, and demonstrated for real-world applications

- Benchmark CFD problem in the automotive industry: slant angle = 20°, $V_\infty = 60$ m/s (216 km/h), zero angle of attack, $\mathrm{Re} = 4.29 \times 10^6$
- RANS (Reynolds-Averaged Navier-Stokes) model based on the Spalart-Allmaras turbulence model
- HDM: second-order in space and time with $N \approx 1.7 \times 10^7$





- PMOR: POD + GNAT, $k = 283$, $k_{\mathbf{f}} = 1\,514$, and $k_i = 2\,268$ (Circa 2011)



Method	CPU time	Number of CPUs	Relative error
HDM	13.28 h	512	–
PROM with GNAT	3.88 h	4	0.68%

- PMOR has been performed for a long time using the classical linear or affine subspace approximation $\tilde{\mathbf{w}} = \mathbf{V}\mathbf{q} + \mathbf{w}_{\text{ref}}$, $\mathbf{V} \in \mathbb{R}^{N \times k}$, $\mathbf{q} \in \mathbb{R}^{k}$, $\mathbf{w}_{\text{ref}} \in \mathbb{R}^{N}$
- For highly nonlinear and transport (first-order hyperbolic) problems in general, and for convection-dominated flow problems in particular, the convergence of a subspace approximation is limited by the slow decay of the Kolmogorov k-width $d_k(\mathcal{M})$
- $d_k(\mathcal{M})$ is the worst-case error resulting from projection onto an optimal subspace $\mathcal{M}$ of dimension $k \ll N$
- Recent strategies for mitigating this issue share the abandonment of the traditional affine approximation in favor of a nonlinear one such as: a **piecewise linear or affine** approximation; a **quadratic** approximation; and an **arbitrarily nonlinear** approximation grounded in **deep learning**

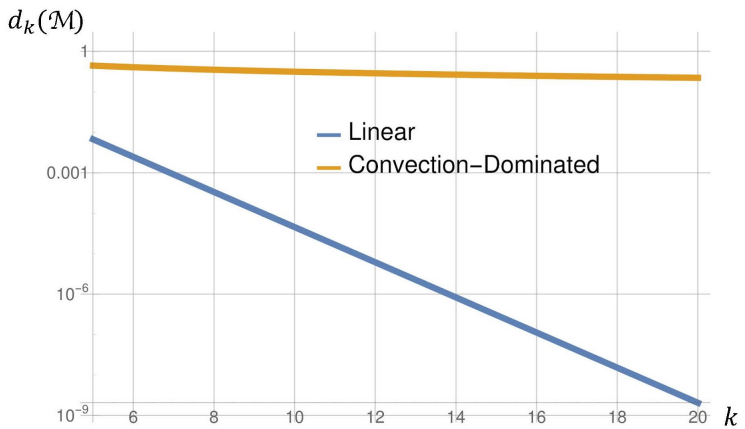
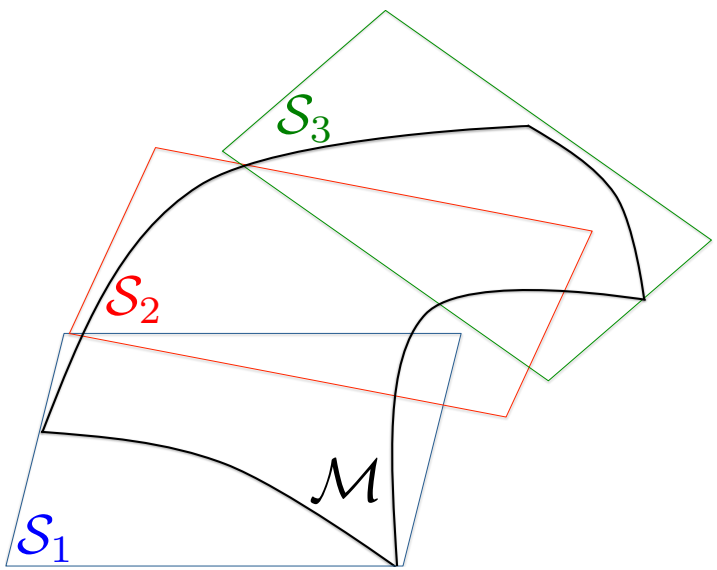


Figure: For most linear problems, $d_k(\mathcal{M})$ exhibits exponential decay; for convection-dominated flow problems, its exhibits a decay of $\mathcal{O}\left(k^{-1/2}\right)$.

- A piecewise linear or affine approximation is the simplest nonlinear approximation
- Additional benefit: Approximating the solution manifold $\mathcal{M}$ with a single linear or affine subspace $\mathcal{S}$ may result in a large-dimensional subspace, which can hinder computational efficiency; in contrast, using local subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^{L}$ to approximate $\mathcal{M}$ allows for tailoring the approximation to different physics regimes, leading to improved computational efficiency



- In practice, the local approximation of the state takes place at the *discrete* level
- Each subspace $\mathcal{S}_\ell$ is associated with a pre-computed *local* Reduced-Order Basis (ROB) $\mathbf{V}_\ell$
- At each time step n, the state $\mathbf{w}^n$ is computed as

$$\mathbf{w}^n = \mathbf{w}^{n-1} + \Delta\mathbf{w}^n$$

- The increment $\Delta\mathbf{w}^n$ is then approximated in a subspace $\mathcal{S}_{\ell(n)} = \text{range}\left(\mathbf{V}_{\ell(n)}\right)$ as

$$\Delta\mathbf{w}^n \approx \mathbf{V}_{\ell(n)}\tilde{\mathbf{q}}^n$$

- The choice of the pre-computed ROB $\mathbf{V}_{\ell(n)}$ is specified later
- By induction, the state $\mathbf{w}^n$ is computed as

$$\mathbf{w}^n = \mathbf{w}^0 + \sum_{i=1}^{n} \mathbf{V}_{\ell(i)}\tilde{\mathbf{q}}^i$$

- The state $\mathbf{w}^n$ is computed as

$$\mathbf{w}^n = \mathbf{w}^0 + \sum_{i=1}^{n} \mathbf{V}_{\ell(i)} \tilde{\mathbf{q}}^i$$

- In practice, the ROBs $\left\{\mathbf{V}_{\ell(i)}\right\}_{i=1}^{n}$ are chosen among a finite set of pre-computed local ROBs $\left\{\mathbf{V}_\ell\right\}_{\ell=1}^{L}$
- Hence

$$\mathbf{w}^n = \mathbf{w}^0 + \sum_{\ell=1}^{L} \mathbf{V}_\ell \mathbf{q}_\ell^n$$

- This shows that

$$\mathbf{w}^n \in \mathbf{w}^0 + \text{range}([\mathbf{V}_1 \quad \cdots \quad \mathbf{V}_L])$$

- Note that each local ROB can be of a different dimension

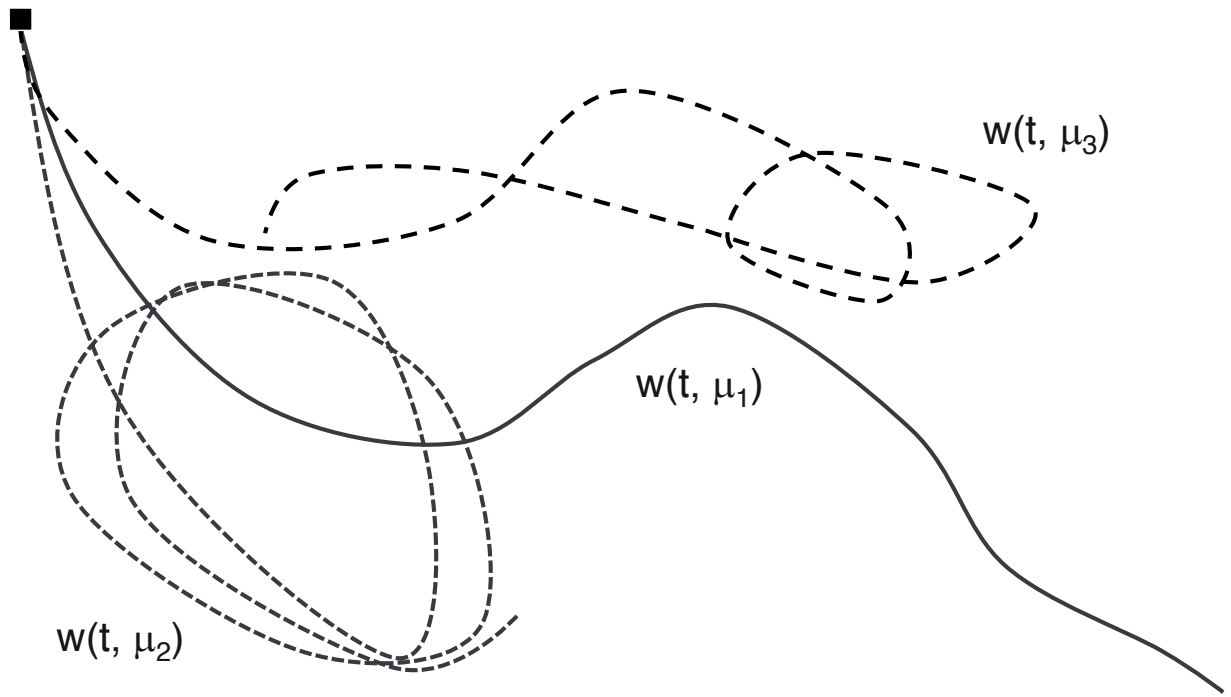
$$\mathbf{V}_\ell \in \mathbb{R}^{N \times k_\ell}$$

- Intuitively, a given local subspace $\mathcal{S}_\ell$ should approximate only a portion of the solution manifold $\mathcal{M}$
- The solution manifold is a subset of the solution space $\mathbb{R}^N$

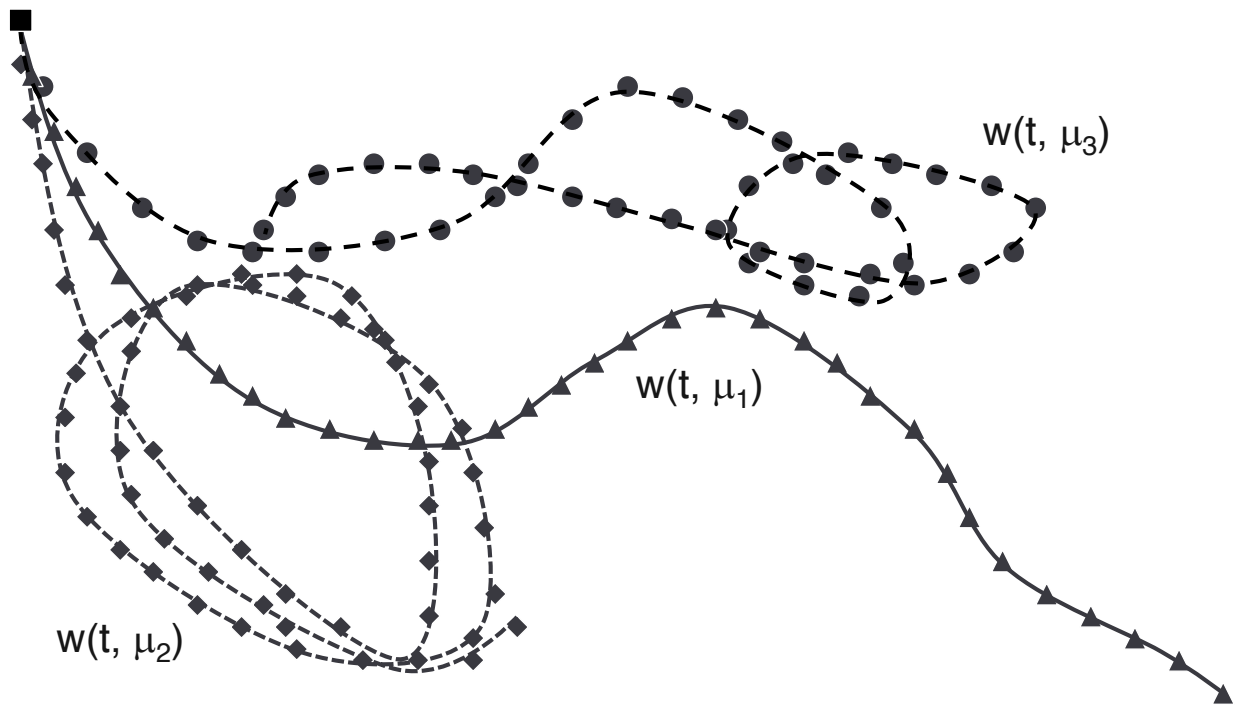
$$\mathcal{M} \subset \mathbb{R}^N$$

- $\mathbb{R}^N$ is partitioned into L subdomains, where each subdomain is associated with a local approximation subspace $\mathcal{S}_\ell = \text{range}(\mathbf{V}_\ell)$
- In practice, a set of solution snapshots $\{\mathbf{w}_s\}_{s=1}^{N_{\text{snap}}}$ – where in general, $\mathbf{w}_s = \mathbf{w}\left(t_i; \boldsymbol{\mu}^{(j)}\right)$ – is partitioned into L subsets using the k-means clustering algorithm
- This leads to a Voronoi tessellation of $\mathbb{R}^N$
- The k-means clustering algorithm is distance-dependent
- After clustering, each snapshot subset can be compressed into a local ROB, for example, using POD

- Local ROBs construction procedure

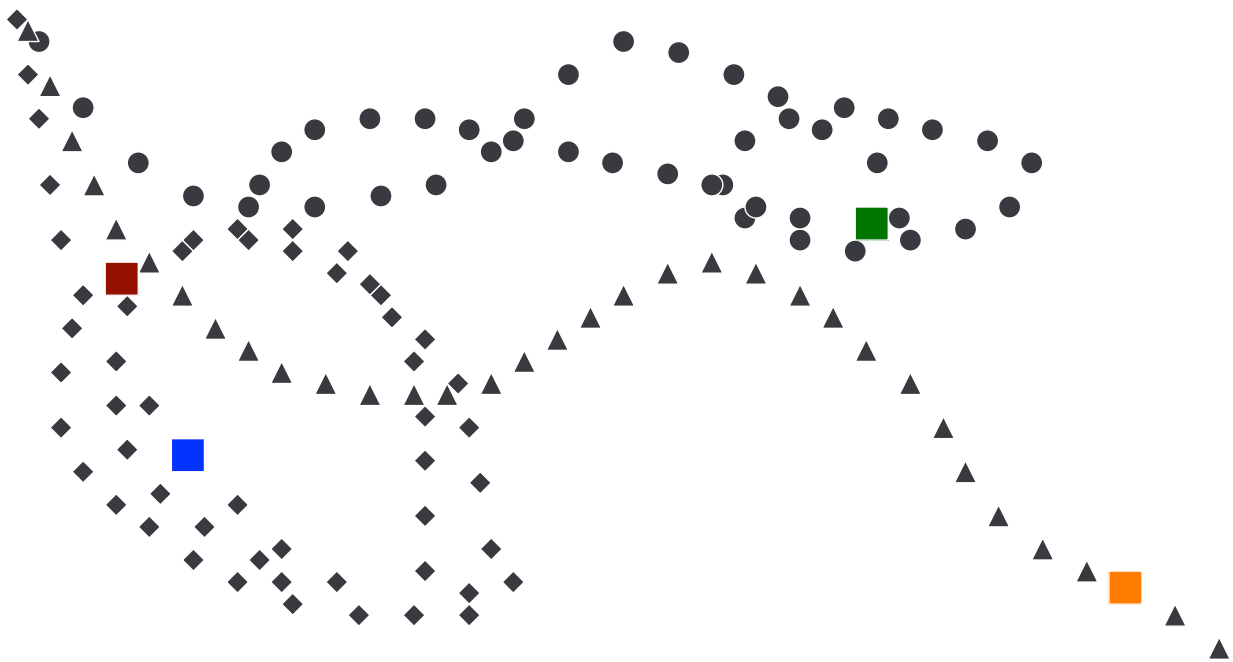


- Local ROBs construction procedure

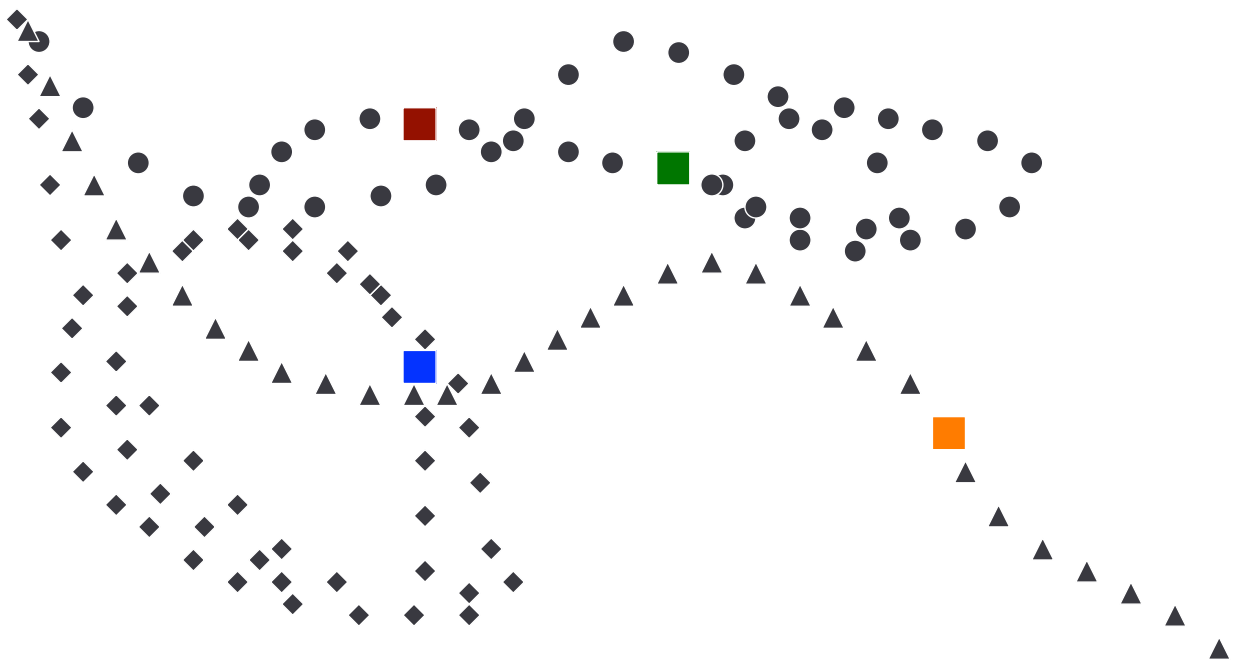


- Local ROBs construction procedure

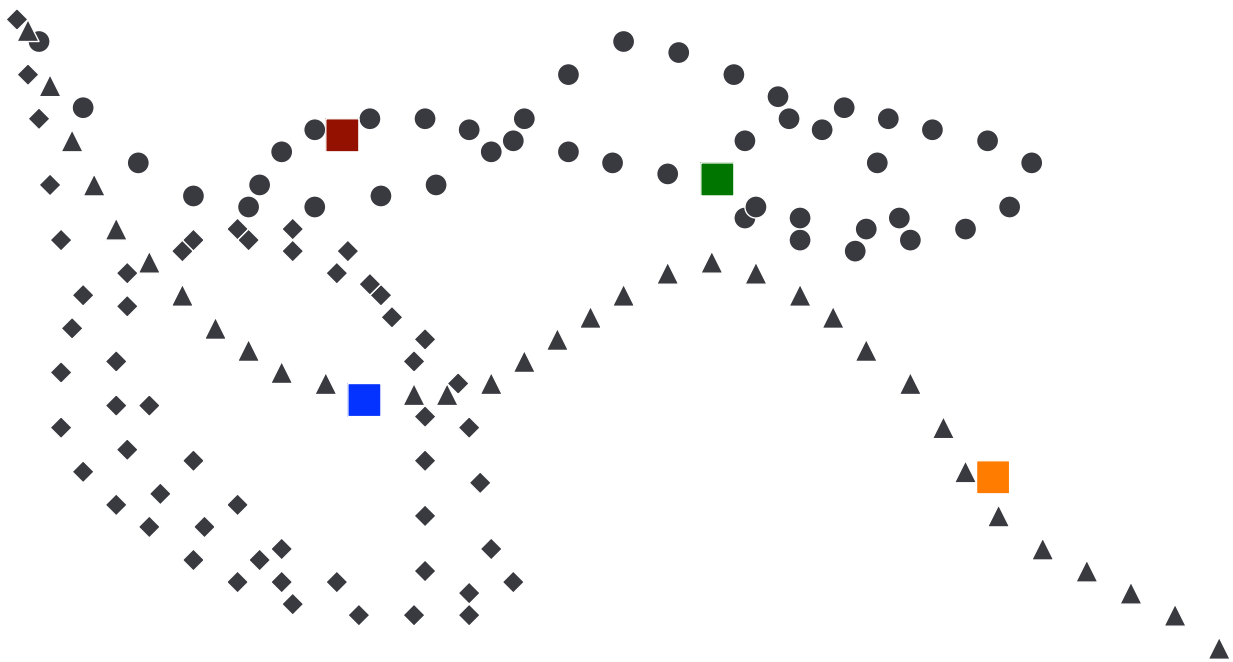
- Local ROBs construction procedure



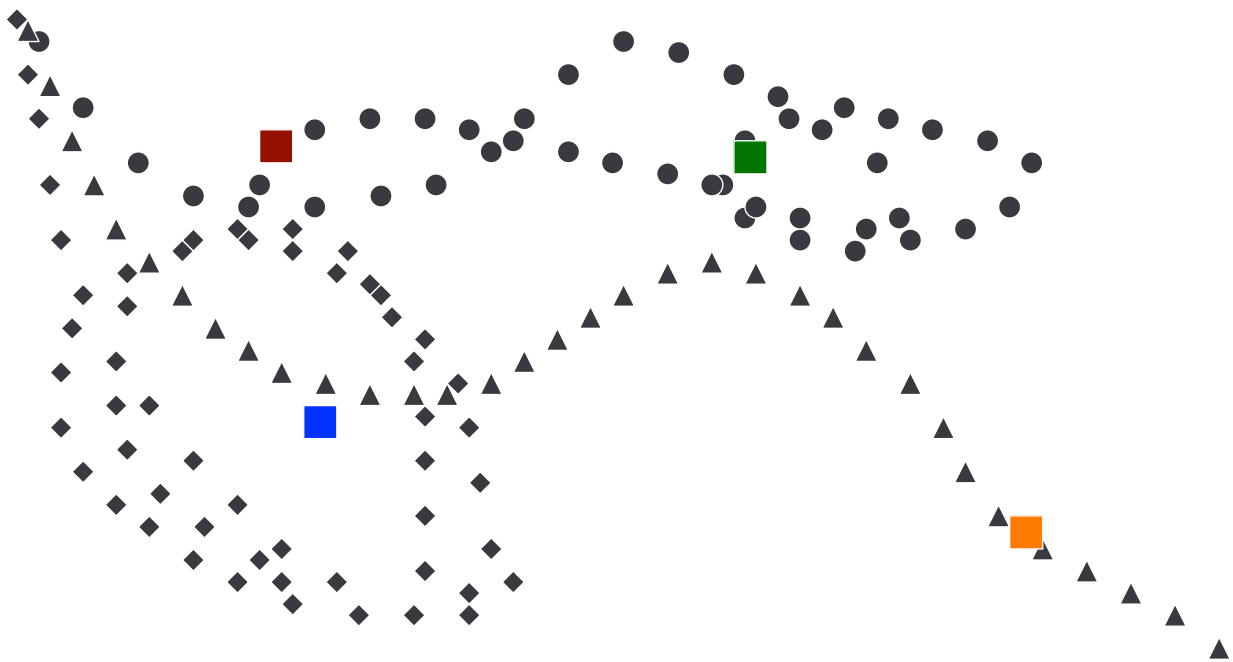
- Local ROBs construction procedure



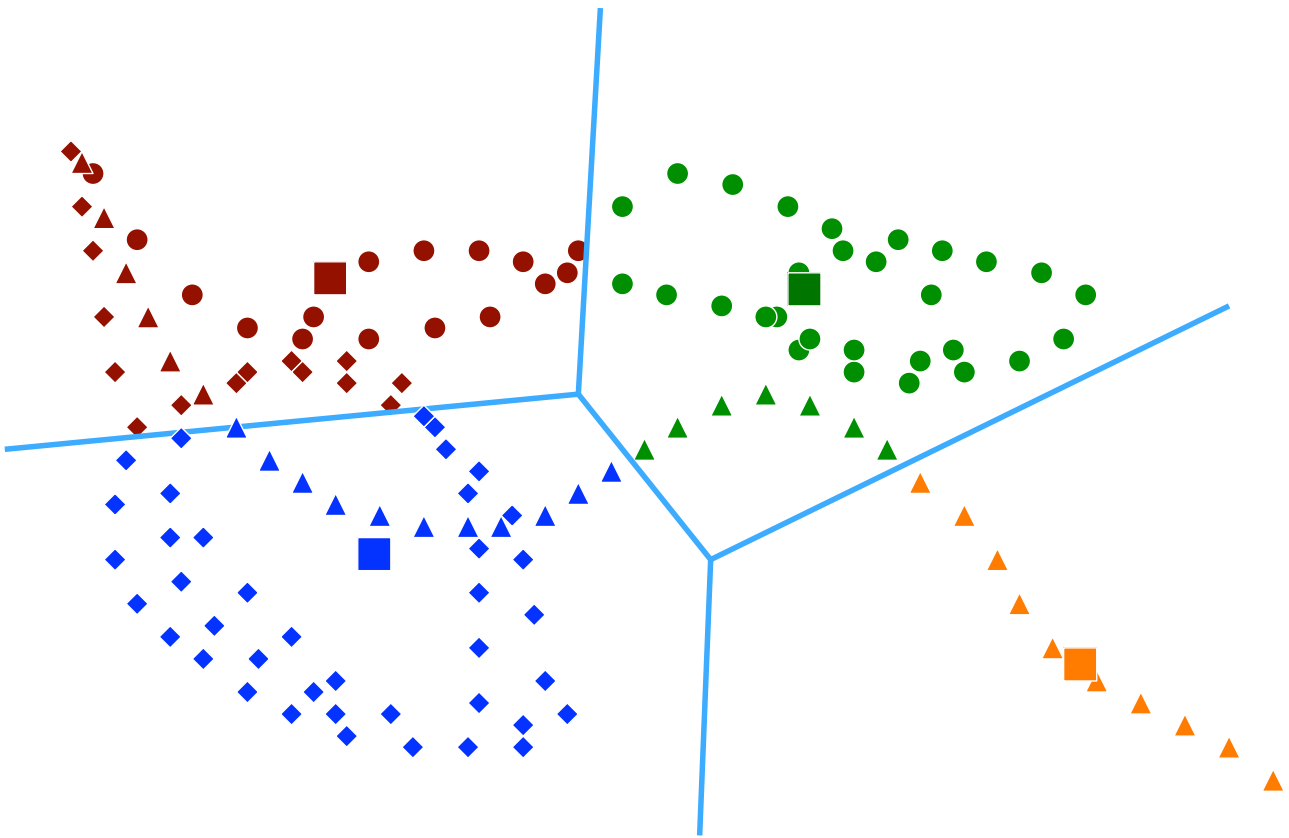
- Local ROBs construction procedure



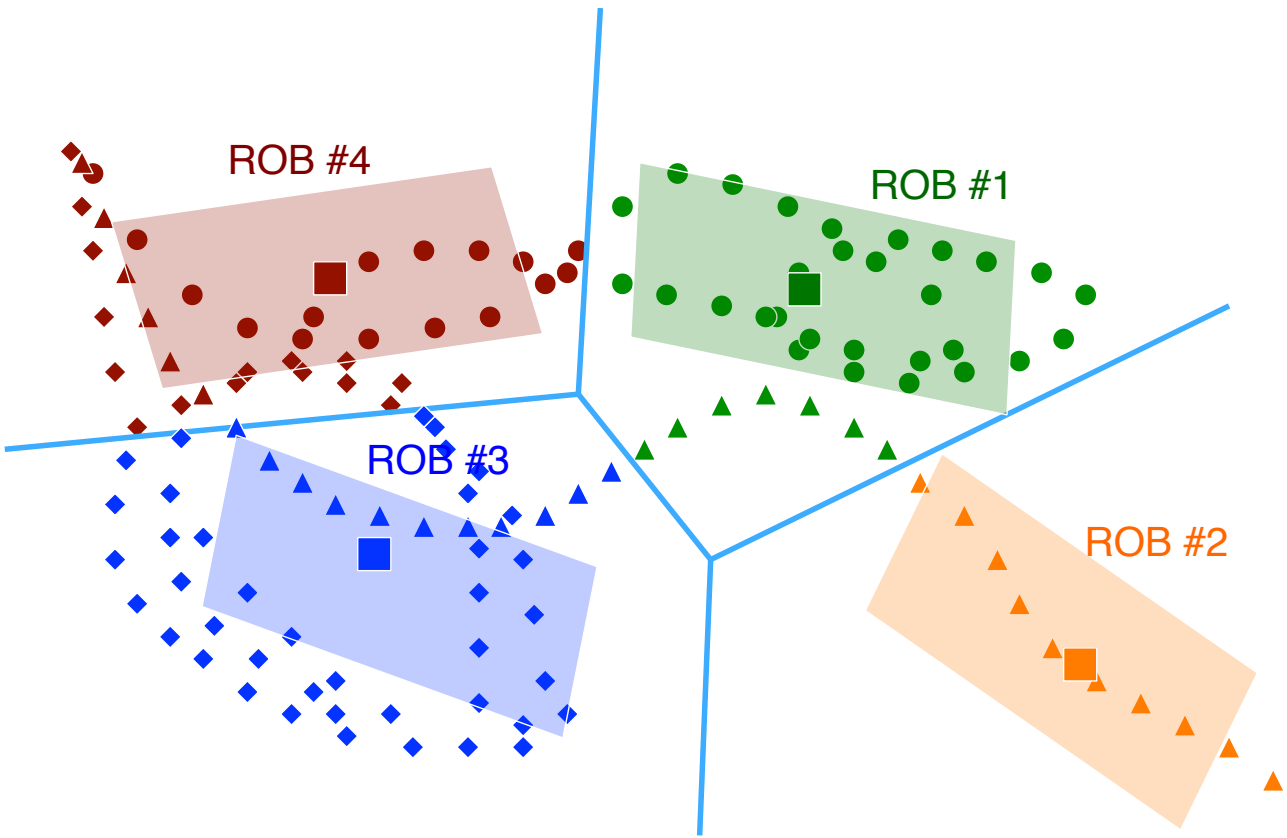
- Local ROBs construction procedure



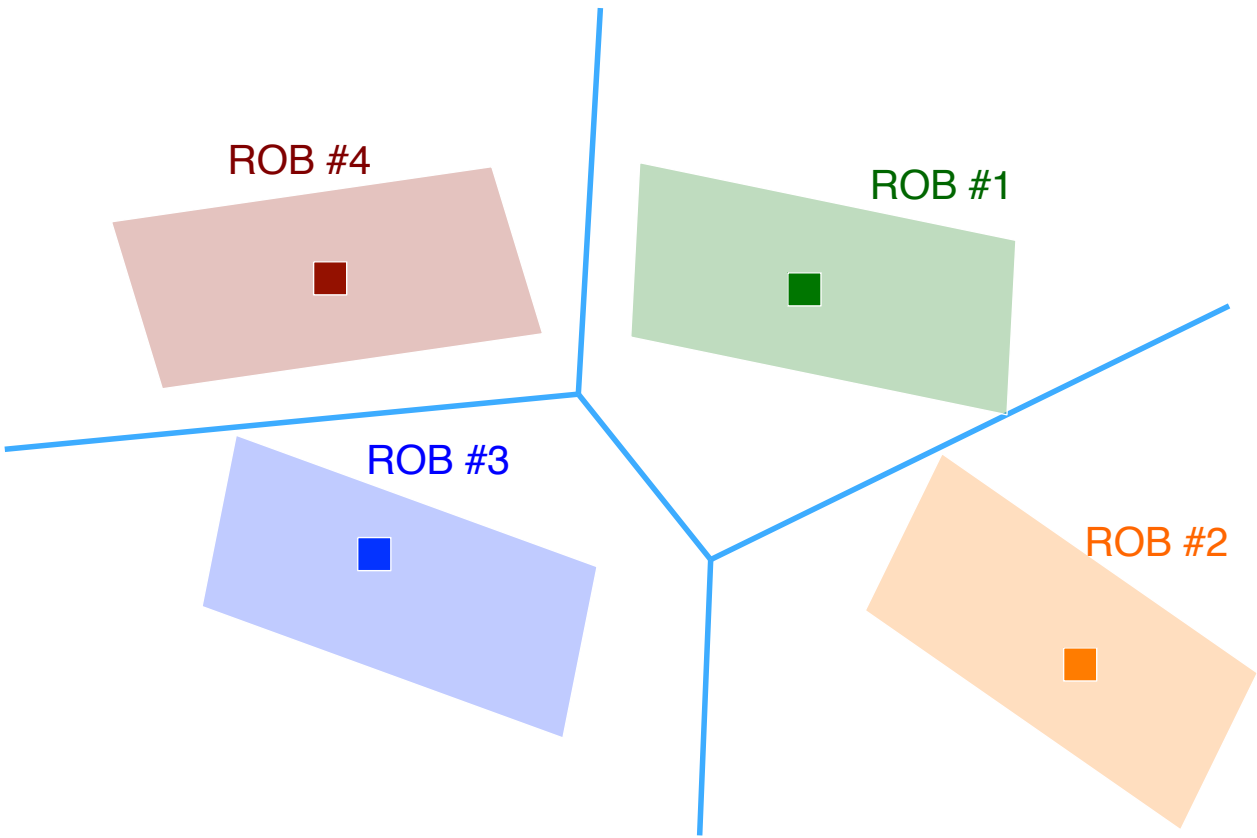
- Local ROBs construction procedure



- Local ROBs construction procedure



- Local ROBs construction procedure



- Online, at time step n, a pre-computed local ROB $\mathbf{V}_{\ell(n)}$ must be chosen
- The selection is based on the current location of $\mathbf{w}^{n-1}$ on the solution manifold $\mathcal{M}$
- The local approximation subspace is selected as that associated with the cluster whose center is the closest to $\mathbf{w}^{n-1}$

$$\ell(n) = \underset{\ell \in \{1,\cdots,L\}}{\arg\min}\ d\left(\mathbf{w}^{n-1}, \mathbf{w}_\ell^c\right)$$

- Consider the case of the distance based on a weighted Euclidian norm

$$d(\mathbf{w}, \mathbf{z}) = \|\mathbf{w} - \mathbf{z}\|_{\mathbf{H}} = \sqrt{(\mathbf{w} - \mathbf{z})^T \mathbf{H} (\mathbf{w} - \mathbf{z})}$$

where $\mathbf{H} \in \mathbb{R}^{N \times N}$ is a symmetric positive definite matrix

- Choice of the local approximation subspace at time step n

$$\ell(n) = \underset{\ell \in \{1,\cdots,L\}}{\arg\min}\ d\left(\mathbf{w}^{n-1}, \mathbf{w}_\ell^c\right)$$

- For a distance based on a weighted Euclidian norm, the solution of the above problem can be computed efficiently at a cost that does not depend on the large dimension N

- To show this, consider the special form of the solution

$$\mathbf{w}^{n-1} = \mathbf{w}^0 + \sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1}$$

- Then, one needs to compare the distances $d\left(\mathbf{w}^{n-1}, \mathbf{w}_\ell^c\right)$ and $d\left(\mathbf{w}^{n-1}, \mathbf{w}_m^c\right)$ for $1 \leq \ell \neq m \leq L$

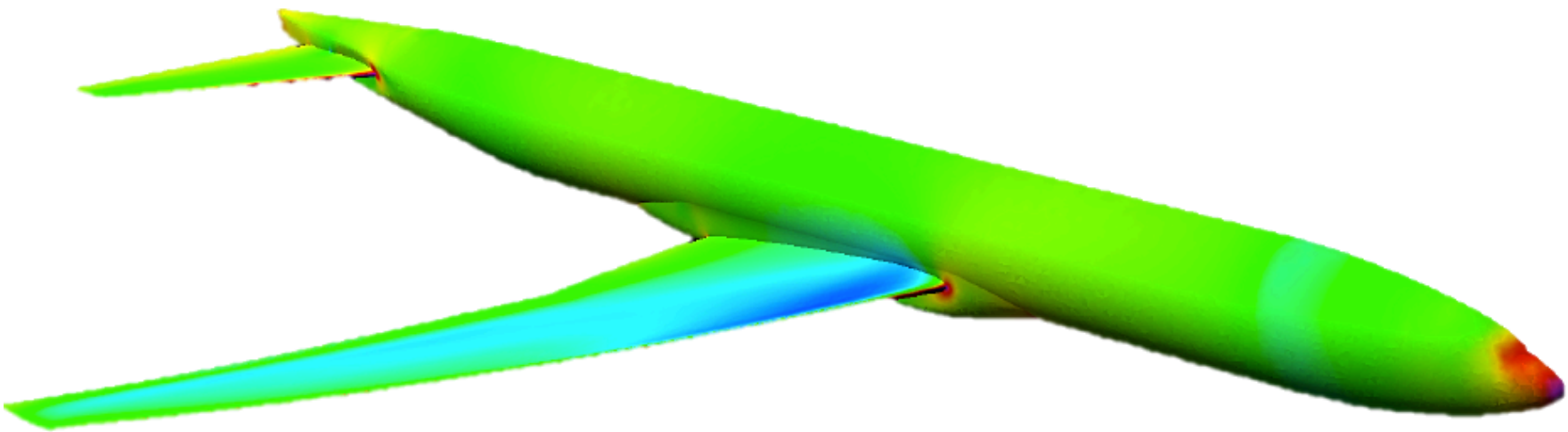
- The two distances $d\left(\mathbf{w}^{n-1}, \mathbf{w}_\ell^c\right)$ and $d\left(\mathbf{w}^{n-1}, \mathbf{w}_m^c\right)$ can be compared as follows

$$
\begin{aligned}
\Delta_{\ell,m} &= d\left(\mathbf{w}^{n-1}, \mathbf{w}_\ell^c\right)^2 - d\left(\mathbf{w}^{n-1}, \mathbf{w}_m^c\right)^2 \\
&= \left\|\mathbf{w}^{n-1} - \mathbf{w}_\ell^c\right\|_{\mathbf{H}}^2 - \left\|\mathbf{w}^{n-1} - \mathbf{w}_m^c\right\|_{\mathbf{H}}^2 \\
&= \left\|\sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1}\right\|_{\mathbf{H}}^2 + \left\|\mathbf{w}^0 - \mathbf{w}_\ell^c\right\|_{\mathbf{H}}^2 + 2\left(\mathbf{w}^0 - \mathbf{w}_\ell^c\right)^T \sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1} \\
&\quad - \left\|\sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1}\right\|_{\mathbf{H}}^2 - \left\|\mathbf{w}^0 - \mathbf{w}_m^c\right\|_{\mathbf{H}}^2 - 2\left(\mathbf{w}^0 - \mathbf{w}_m^c\right)^T \sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1} \\
&= \left\|\mathbf{w}^0 - \mathbf{w}_\ell^c\right\|_{\mathbf{H}}^2 - \left\|\mathbf{w}^0 - \mathbf{w}_m^c\right\|_{\mathbf{H}}^2 - 2\left(\mathbf{w}_\ell^c - \mathbf{w}_m^c\right)^T \sum_{i=1}^{L} \mathbf{V}_i \mathbf{q}_i^{n-1}
\end{aligned}
$$

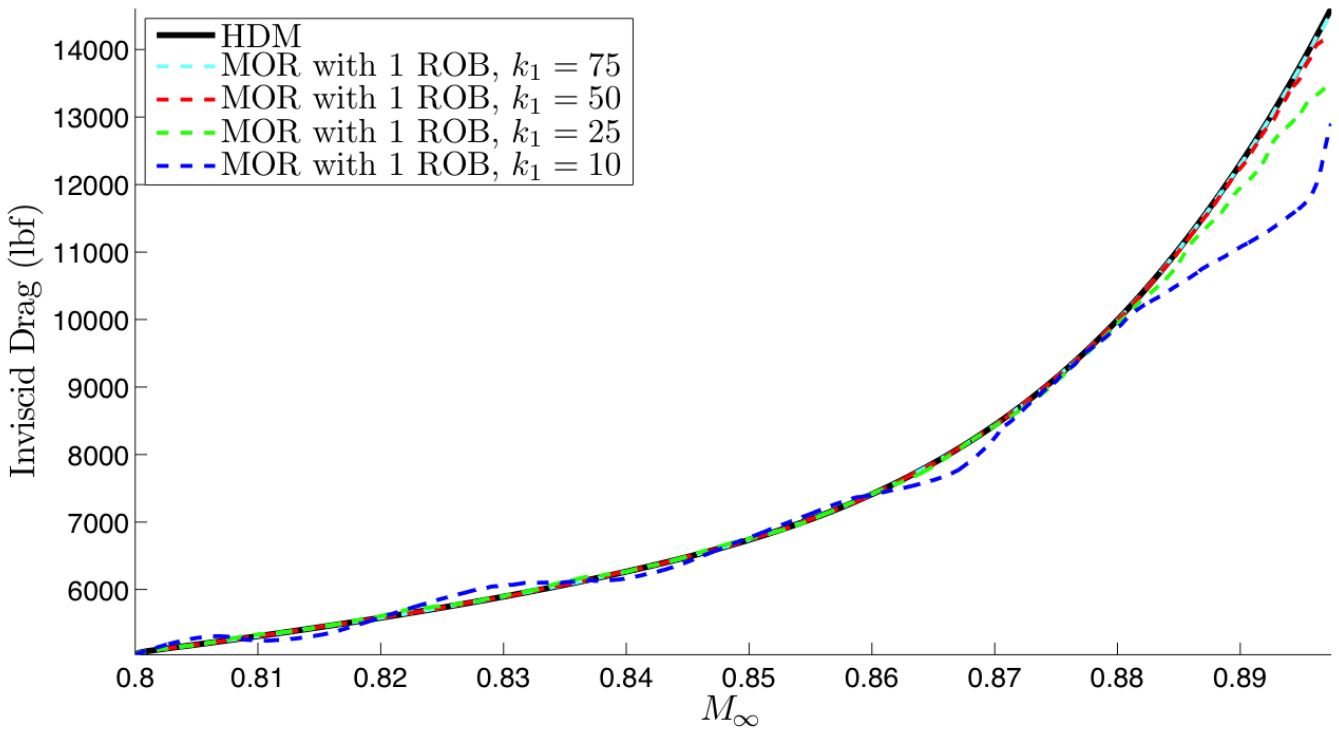
- The following low-dimensional quantities can be pre-computed offline and exploited online to compute rapidly $\Delta_{\ell,m},\ 1 \leq \ell \neq m \leq L$
  $a_{\ell,m} = \left\|\mathbf{w}^0 - \mathbf{w}_\ell^c\right\|_{\mathbf{H}}^2 - \left\|\mathbf{w}^0 - \mathbf{w}_m^c\right\|_{\mathbf{H}}^2 \in \mathbb{R},\ \mathbf{g}_{\ell,m} = \left(\mathbf{w}_\ell^c - \mathbf{w}_m^c\right)^T \mathbf{V}_i \in \mathbb{R}^{k_i}$

- The nonlinear PMOR method based on a piecewise linear or affine approximation has been equipped (relatively easily) with the hyperreduction methods DEIM and ECSW

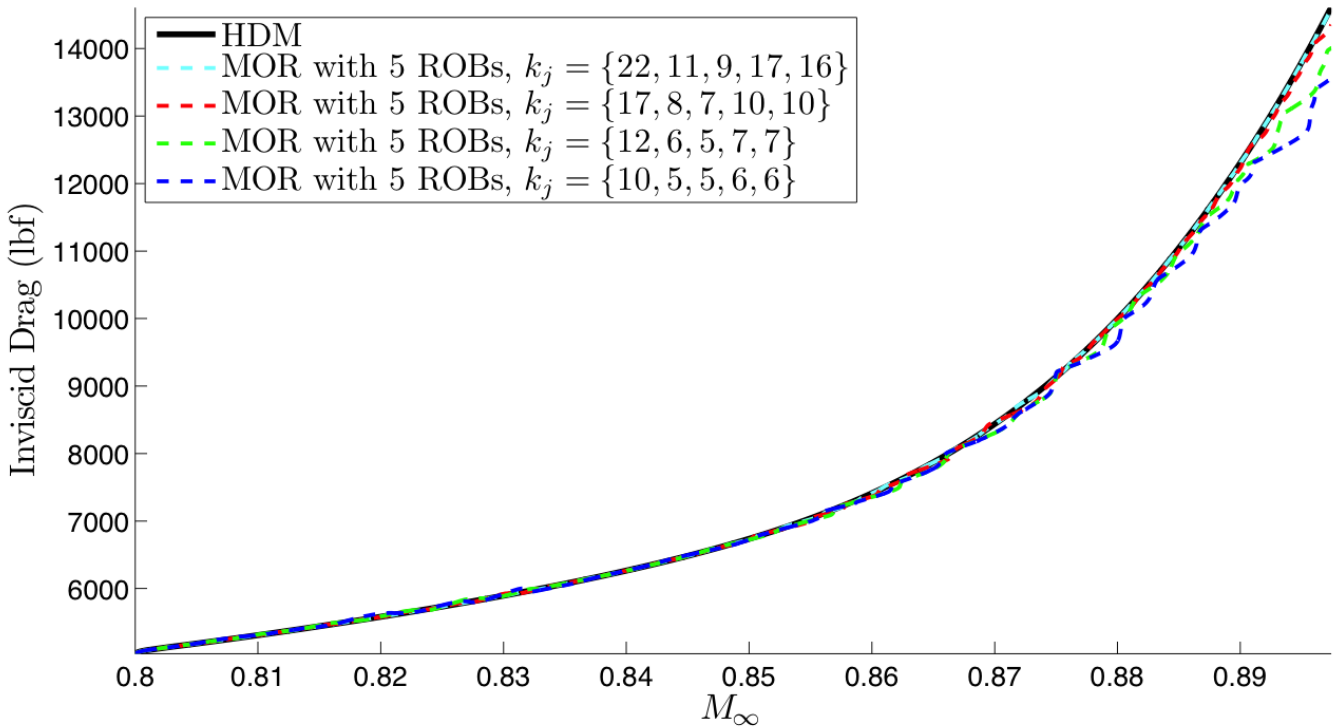
- Flow past the NASA Common Research Model (CRM) – (CFD benchmark in the aeronautical industry)
- 3D compressible Euler equations
- $N = 3.1 \times 10^6$
- Constant acceleration of 2.5 $\mathrm{m/s^2}$, from $M_\infty = 0.8$ to $M_\infty = 0.9$



- PMOR using a global ROB



- PMOR using 5 local ROBs



- Very good accuracy can be obtained with $k_\ell \leq 17$ as opposed to $k = 50$ (global ROB) – and therefore much faster than with a global approximation method

- General approach

$$\tilde{\mathbf{w}} = \sum_{i=1}^{p} \mathbf{G}_{p-i+1}\, \mathbf{q}^{\otimes p-i+1} \; + \; \mathbf{w}_{\text{ref}}$$

  where p represents the degree of the polynomial approximation, $\mathbf{G}_j \in \mathbb{R}^{N \times k^j}$, $\otimes j$ designates the j-fold Kronecker product, and therefore $\mathbf{q}^{\otimes j} \in \mathbb{R}^{k^j}$

- The case $p = 1$ recovers the affine approximation, where $\mathbf{G}_1 = \mathbf{V}$

- The following quadratic approximation is framed in a data-driven setting, to obtain a comprehensive, nonlinear approximation approach that is computationally tractable for large-scale problems and effective in *delaying* the effect of the Kolmogorov k-width

$$\boxed{\tilde{\mathbf{w}} = \mathbf{H}\,\mathbf{q}^{\otimes 2} + \mathbf{V}\,\mathbf{q} + \mathbf{w}_{\text{ref}}, \quad \mathbf{H} \in \mathbb{R}^{N \times k^2}}$$

  - $\mathbf{q} \in \mathbb{R}^k$ is the traditional reduced-order vector of generalized coordinates associated with the ROB $\mathbf{V} \in \mathbb{R}^{N \times k}$
  - $\mathbf{q}^{\otimes 2} \in \mathbb{R}^{k^2}$ is the vectorized Kronecker product given by

$$\mathbf{q}^{\otimes 2} = \begin{bmatrix} q_1^2 \; q_1 q_2 \; \cdots \; q_1 q_k & q_2 q_1 \; q_2^2 \; \cdots \; q_2 q_k & q_k q_1 \; q_k q_2 \; \cdots \; q_k^2 \end{bmatrix}^T$$

$$\tilde{\mathbf{w}} = \mathbf{H}\,\mathbf{q}^{\otimes 2} + \mathbf{V}\,\mathbf{q} + \mathbf{w}_{\text{ref}}, \quad \mathbf{H} \in \mathbb{R}^{N \times k^2}$$

- Train $\mathbf{H}$ in a three-step process
  - first, construct $\mathbf{V}$ by compressing a series of snapshots $\mathbf{w}_s$, $s = 1, \cdots, N_{\text{snap}}$
  - next, compute the set of generalized coordinates $\mathbf{q}_s = \mathbf{V}^T(\mathbf{w}_s - \mathbf{w}_{\text{ref}})$, $s = 1, \cdots, N_{\text{snap}}$
  - then, determine $\mathbf{H}$ from the resulting set of reduced coordinates $\mathbf{q}_s$ to reduce further the error vectors $\mathbf{e}_s = \mathbf{w}_s - \mathbf{V}\,\mathbf{q}_s$

- Once the error vectors associated with the affine approximation are computed, determine $\mathbf{H}$ by minimizing a global loss function as follows

$$\arg \min_{\mathbf{H} \in \mathbb{R}^{N \times k^2}} \left\| \begin{pmatrix} \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_{N_{\text{snap}}}^T \end{pmatrix}^T - \mathbf{H} \begin{pmatrix} \mathbf{q}_1^{\otimes 2^T} \\ \vdots \\ \mathbf{q}_{N_{\text{snap}}}^{\otimes 2^T} \end{pmatrix}^T \right\|_F, \ \mathbf{e}_i \in \mathbb{R}^N, \ \mathbf{q}_i^{\otimes 2} \in \mathbb{R}^{k^2}$$

- Let $\mathbf{E} = \begin{bmatrix} \mathbf{e}_1 & \dots & \mathbf{e}_{N_{\text{snap}}} \end{bmatrix} \in \mathbb{R}^{N \times N_{\text{snap}}}$, $\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1^{\otimes 2} & \dots & \mathbf{q}_{N_{\text{snap}}}^{\otimes 2} \end{bmatrix} \in \mathbb{R}^{k^2 \times N_{\text{snap}}}$: Then, (2) can be re-written as

$$\mathbf{H} = \arg \min_{\mathbf{H}' \in \mathbb{R}^{N \times k^2}} \|\mathbf{E} - \mathbf{H}'\mathbf{Q}\|_F = \arg \min_{\mathbf{H}' \in \mathbb{R}^{N \times k^2}} \sum_{i=1}^{N} \left\|\mathbf{E}_{i,:} - \mathbf{H}'_{i,:}\,\mathbf{Q}\right\|_2^2$$

- Hence, (2) is equivalent to the following N independent minimization problems

$$\mathbf{h}_i = \mathbf{H}_{i,:} = \arg \min_{\mathbf{h}'_i \in \mathbb{R}^{1 \times k^2}} \|\mathbf{E}_{i,:} - \mathbf{h}'_i\mathbf{Q}\|_2^2, \qquad i = 1, \dots, N$$

- It follows that $\mathbf{H}$ can be determined by solving the following N independent minimization problems in an embarrassingly parallel fashion

$$\mathbf{h}_i = \mathbf{H}_{i,:} = \arg \min_{\mathbf{h}_i' \in \mathbb{R}^{1 \times k^2}} \|\mathbf{E}_{i,:} - \mathbf{h}_i' \mathbf{Q}\|_2^2, \qquad i = 1, \ldots, N$$

- The solution of each of the above optimization problems is vulnerable to overfitting $\Rightarrow$ Tikhonov regularization using $\mathbf{\Gamma} = \alpha \mathbf{I}$, $\alpha > 0$

$$\mathbf{h}_i = \mathbf{H}_{i,:} = \arg \min_{\mathbf{h}_i' \in \mathbb{R}^{1 \times k^2}} \|\mathbf{E}_{i,:} - \mathbf{h}_i' \mathbf{Q}\|_2^2 + \alpha \, ||\mathbf{h}_i'||_2^2, \quad i = 1, \ldots, N$$

- Due to the symmetry of the Kronecker product $\mathbf{q}_i^{\otimes 2} = \mathbf{q}_i \otimes \mathbf{q}_i$, there are only $(k+1)k/2$ linearly independent elements in every row of $\mathbf{H}$
  - the $k(k-1)/2$ redundant terms in every row of $\mathbf{H}$ should be eliminated to avoid ill-conditioning
  - $\mathbf{Q} = \left[\mathbf{q}_1^{\otimes 2} \ \ldots \ \mathbf{q}_{N_{\text{snap}}}^{\otimes 2}\right]$ is transformed into a redundancy-free matrix $\overline{\mathbf{Q}} \in \mathbb{R}^{k(k+1)/2 \times N_{\text{snap}}}$ and the corresponding entries in each row vector $\mathbf{h}_i$ are excluded $\Rightarrow \mathbf{h}_i \in \mathbb{R}^{1 \times k(k+1)/2}$

$$\mathbf{h}_i = \mathbf{H}_{i,:} = \arg \min_{\mathbf{h}_i' \in \mathbb{R}^{1 \times k^2}} \|\mathbf{E}_{i,:} - \mathbf{h}_i' \mathbf{Q}\|_2^2 + \alpha \, ||\mathbf{h}_i'||_2^2, \quad i = 1, \ldots, N$$

- It is preferrable that N_{snap} and k verify $N_{\text{snap}} > k(k+1)/2$, so that each of the above regularized least-squares problem is overdetermined and can be solved using the non-truncated thin SVD of $\overline{\mathbf{Q}}$ as follows

$$\overline{\mathbf{Q}} = \mathbf{U}_{\overline{\mathbf{Q}}} \Sigma_{\overline{\mathbf{Q}}} \mathbf{Y}_{\overline{\mathbf{Q}}}^T \Rightarrow \bar{\mathbf{h}}_i^T = \sum_{\ell=1}^{k_{\overline{\mathbf{Q}}}} \left(\frac{\sigma_{\overline{\mathbf{Q}},\ell}^2}{\sigma_{\overline{\mathbf{Q}},\ell}^2 + \alpha^2} \right) \frac{\mathbf{y}_{\overline{\mathbf{Q}},\ell}^T \left(\mathbf{E}^T\right)_i}{\sigma_{\overline{\mathbf{Q}},\ell}} \mathbf{u}_{\overline{\mathbf{Q}},\ell}, \quad i = 1, \ldots, N$$

- A reasonable value of α can be found using GCV
- Occasional overregularization may occur

- $\mathbf{V}_{\text{qua}} \in \mathbb{R}^{N \times k_{\text{qua}}}$ is built as in the traditional subspace approximation, where k_{qua} is determined using the singular value energy criterion
- For the same reference solution $\mathbf{w}_{\text{ref}} \in \mathbb{R}^N$
  - traditional subspace approximation depends on Nk_{tra} control variables defining $\mathbf{V}_{\text{tra}}$
  - quadratic approximation depends on $Nk_{\text{qua}}(k_{\text{qua}}+1)/2 + Nk_{\text{qua}}$ control variables defining the rows $\bar{\mathbf{h}}_i \in R^{1 \times k_{\text{qua}}(k_{\text{qua}}+1)/2}$, $i = 1, \ldots, N$, and $\mathbf{V}_{\text{qua}} \in \mathbb{R}^{N \times k_{\text{qua}}}$
- Hence, equating the two different numbers of control variables suggests that a quadratic PROM of dimension $k_{\text{qua}} = \left(\sqrt{9 + 8k_{\text{tra}}} - 3\right)/2 \ll k_{\text{tra}}$ should deliver the same accuracy as that of a traditional counterpart of dimension k_{tra}
- GCV is vulnerable to overregularization $\Rightarrow$ potential loss of some of the capacity of the $Nk_{\text{qua}}(k_{\text{qua}}+1)/2$ control variables defining $\bar{\mathbf{h}}_i \in \mathbb{R}^{1 \times k_{\text{qua}}(k_{\text{qua}}+1)/2}$, $i = 1, \ldots, N$
- The above result can be interpreted as working effectively with a number of control variables $Nk'_{\text{qua}}(k'_{\text{qua}}+1)/2 < Nk_{\text{qua}}(k_{\text{qua}}+1)/2$ – or equivalently, with a dimension $k'_{\text{qua}} < k_{\text{qua}}$

- Compute the dimension k_{tra} associated with the singular value energy criterion (and say $\varepsilon_S = 10^{-4} \Rightarrow 1 - \varepsilon_S = 99.99\%$)
- Compute $k'_{\text{qua}} = \left(\sqrt{9 + 8k_{\text{tra}}} - 3\right)/2$ based on matching the numbers of control variables of the quadratic and traditional PROMs, and assuming no overregularization of the least-squares problems defining $\mathbf{H}$
- Set $k_{\text{qua}} = (1 + \zeta)k'_{\text{qua}}$, where $0 < \zeta < 0.2$, to correct for any overregularization of the least-squares problems defining $\mathbf{H}$
- Finally, set

$$\boxed{k = \min\left(k_{\text{qua}}, \left(\sqrt{1 + 8N_{\text{snap}}} - 1\right)/2\right)}$$

  to satisfy the constraint $N_{\text{snap}} > k(k+1)/2$
- Note: interestingly, $k_{\text{qua}} \approx \sqrt{k_{\text{tra}}}$ (asymptotically)

- Residual minimization in the 2-norm

$$\mathbf{q}^{n+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^k} \left\| \mathbf{r}^{n+1}\left(\underbrace{\mathbf{H}\,\mathbf{x}^{n+1\otimes 2} + \mathbf{V}\,\mathbf{x}^{n+1} + \mathbf{w}_{\text{ref}}}_{\tilde{\mathbf{w}}^{n+1}(\mathbf{x}^{n+1})}, t^{n+1} \right) \right\|_2^2$$

$$\iff \mathbf{q}^{n+1} \text{ solution of } \mathbf{W}^{n+1^T}\mathbf{r}^{n+1}\left(\underbrace{\mathbf{H}\,\mathbf{q}^{n+1\otimes 2} + \mathbf{V}\,\mathbf{q}^{n+1} + \mathbf{w}_{\text{ref}}}_{\tilde{\mathbf{w}}^{n+1}(\mathbf{q}^{n+1})}, t^{n+1} \right) = \mathbf{0}$$

- Hence, for a quadratic LSPG PROM, the *left* ROB is given by

$$\mathbf{W}^{n+1} = \underbrace{\mathbf{J}^{n+1}\,\mathbf{H}\left[\mathbf{q}^{n+1}\otimes\mathbf{I} + \mathbf{I}\otimes\mathbf{q}^{n+1}\right]}_{\substack{\textit{additional term}\\ \textit{due to quadratic approximation}}} + \underbrace{\mathbf{J}^{n+1}\mathbf{V}}_{\substack{\textit{traditional}\\ \textit{left ROB}}}, \quad \mathbf{J}^{n+1} = \frac{\partial \mathbf{r}^{n+1}}{\partial \tilde{\mathbf{w}}}\left(\tilde{\mathbf{w}}^{n+1}\right) \in \mathbb{R}^{N\times N}$$

- Additional cost associated with constructing the left ROB

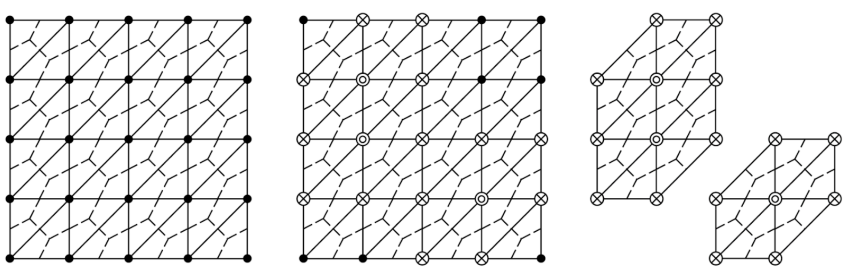
$$\mathbf{W}^{n+1} = \underbrace{\mathbf{J}^{n+1}\,\mathbf{H}\left[\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right]}_{\substack{\textit{additional term}\\ \textit{due to quadratic approximation}}} + \underbrace{\mathbf{J}^{n+1}\mathbf{V}}_{\substack{\textit{traditional}\\ \textit{left ROB}}} \quad , \mathbf{J}^{n+1} = \frac{\partial \mathbf{r}^{n+1}}{\partial \tilde{\mathbf{w}}}\left(\tilde{\mathbf{w}}^{n+1}\right) \in \mathbb{R}^{N\times N}$$

  - $\left(\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right) \in \mathbb{R}^{k(k+1)/2\times k}$ is a sparse matrix $\Rightarrow$ $\mathbf{H}\left[\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right]$ should be performed using dense-sparse matrix-matrix computations
  - the number of nonzero entries in $\left(\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right)$ grows as k^2
  - each evaluation of $\mathbf{H}\left[\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right]$ requires $\mathcal{O}\left(Nk^2\right)$ operations
  - the construction of $\mathbf{W}^{n+1}$ requires $\mathcal{O}(2N^3k)$ operations whereas in the case of the traditional PROM, it requires $\mathcal{O}(N^3k)$ operations
  - hyperreduction eliminates the dependence on N from both complexities
  - in the case of the quadratic PROM (QPROM), the construction of $\mathbf{W}$ requires the additional storage of the matrix $\mathbf{H}\left[\mathbf{q}^{n+1}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{q}^{n+1}\right] \in R^{N\times k}$

- Energy conserving sampling and weighting (ECSW)
  - project-then-approximate rather than approximate-then-project

$$\begin{aligned}\mathbf{r}_k^{n+1}\left(\mathbf{q}^{n+1}, t^{n+1}\right) &= \mathbf{W}^T \mathbf{r}^{n+1}\left(\mathbf{H}\,\mathbf{q}^{n+1\otimes 2} + \mathbf{V}\,\mathbf{q}^{n+1} + \mathbf{w}_{\text{ref}}, t^{n+1}\right) \\ &\approx \sum_{e_i \in \widetilde{\mathcal{E}}} \xi^{e_i} \left(\mathbf{L}^{e_i}\mathbf{W}\right)^T \mathbf{r}^{e_i\,n+1}\left(\mathbf{L}^{e_i+}\left[\mathbf{H}\,\mathbf{q}^{n+1\otimes 2} + \mathbf{V}\,\mathbf{q}^{n+1} + \mathbf{w}_{\text{ref}}\right], t^{n+1}\right)\end{aligned}$$

  - recall the interpretation of the ECSW method: cubature where the elements of the reduced mesh $\widetilde{\mathcal{E}} \subset \mathcal{E}$ are the points and $\left\{\xi_{e_1}, \ldots, \xi_{e_{\widetilde{N_e}}}\right\}$ are the corresponding weights
  - implementation (context of the finite volume method): augmented reduced mesh



Augmented reduced mesh: ◎ represents a selected node attached to a selected element; and ⊗ represents an added node to enable the full representation of the computational stencil at the selected node/element

- ECSW training for the QPROM predictions

$$\mathbf{r}_k^{n+1}\left(\mathbf{q}^{n+1}, t^{n+1}\right) \approx \sum_{e_i \in \widetilde{\mathcal{E}}} \xi^{e_i} \left(\mathbf{L}^{e_i}\mathbf{W}\right)^T \mathbf{r}^{e_i\,n+1} \left(\mathbf{L}^{e_i\,+} \left[\underbrace{\mathbf{H}\,\mathbf{q}^{n+1\otimes 2} + \mathbf{V}\,\mathbf{q}^{n+1} + \mathbf{w}_{\text{ref}}}_{\tilde{\mathbf{w}}^{n+1}\left(\mathbf{q}^{n+1}\right)} \right], t^{n+1} \right)$$

  - for QPROMs, $\mathbf{q}^{n+1}$ can no longer be identified via projection onto the right ROB: instead, it requires the solution of a nonlinear problem of the form

$$\delta^{n+1}(\mathbf{q}^{n+1}) = \mathbf{H}\,\mathbf{q}^{n+1^{\otimes 2}} + \mathbf{V}\,\mathbf{q}^{n+1} + \mathbf{w}_{\text{ref}} - \mathbf{w}^{n+1} = \mathbf{0} \Rightarrow \tilde{\mathbf{w}}^{\mathbf{n+1}}\left(\mathbf{q}^{\mathbf{n+1}}\right)$$

  - Yet another Gauss-Newton procedure

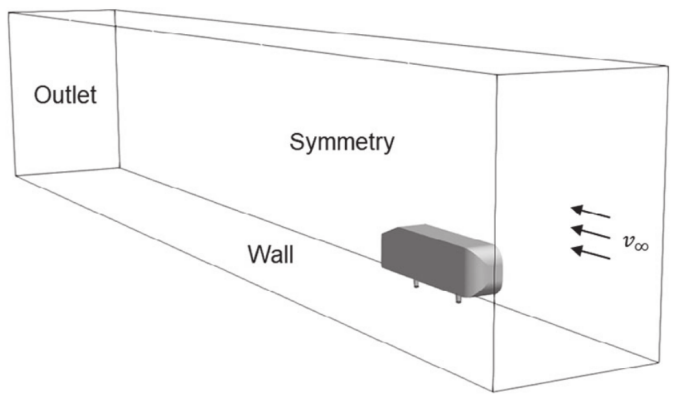
$$\mathbf{q}^{n+1,0} = \mathbf{V}^T\mathbf{w}^{n+1}$$

$$\mathbf{q}^{n+1,\ell+1} = \mathbf{q}^{n+1,\ell} - \left(\frac{\partial \delta^{n+1}}{\partial \mathbf{q}}\left(\mathbf{q}^{n+1,\ell}\right)\right)^{+} \delta^{n+1}\left(\mathbf{q}^{n+1,\ell}\right)$$

  where ℓ designates the ℓ-th iteration and $+$ designates the Moore-Penrose inverse

- Major advantage for hyperreduction
  - recall that ECSW is a cubature method
  - recall that the number of cubature points required for approximating a d-dimensional integral function with p cubature points along each dimension grows as p^d
  - recall that $k_{\text{qua}} \approx \sqrt{k_{\text{tra}}}$
  - it follows that for a fixed level of training accuracy, ECSW can be expected to deliver for a PROM a much smaller reduced mesh than for a traditional counterpart

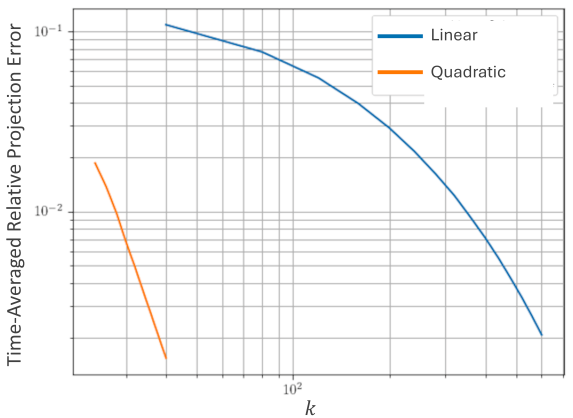
- Benchmark CFD problem in the automotive industry: slant angle = $20°$, $V_\infty = 60$ m/s, zero angle of attack, $\mathrm{Re} = 4.29 \times 10^6$
- DES (detached eddy simulation) model based on the Spalart-Allmaras turbulence model
- HDM: second-order in space and time with $N \approx 1.7 \times 10^7$; executed on **240 CPU cores** of a Linux cluster
- Data collection: $N_{\mathsf{snap}} = 1\,251$ solution snapshots uniformly collected in $[0, 2 \times 10^{-1}]$ s
- All HPROMs executed on **8 CPU cores** of same Linux cluster



- Projection errors
  - time-averaged relative projection error

$$\frac{1}{\sum_{s=1}^{N_{\text{snap}}} \Delta t_s} \sum_{s=1}^{N_{\text{snap}}} \frac{\Delta t_s \left|\left|\tilde{\mathbf{w}}\left(\mathbf{q}_s\right) - \mathbf{w}_s\right|\right|_2}{\left|\left|\mathbf{w}_s\right|\right|_2}$$

  - $\Delta t_s = 2\Delta t = 1.6 \times 10^{-4}$ s
  - hyperreduced QPROM (HQPROM) converges significantly faster than traditional (affine subspace approximation) HPROM



- Relative errors: focus is set on quantities of interest (QoIs)
  - integral QoIs (lift and drag coefficients)
  - pointwise (probed) QoIs (e.g., v_x and v_z)
- Relative errors of HPROM- and HQPROM-based predictions ($\widetilde{QoI}(t)$) are globally measured in $[0, 2 \times 10^{-1}]$ s, with respect to HDM-based counterparts $QoI(t)$, as follows

$$\mathbb{RE}_{\text{QoI}} = \frac{\sqrt{\sum_{t \in \mathcal{T}} \left(\widetilde{QoI}(t) - QoI(t) \right)^2}}{\sqrt{\sum_{t \in \mathcal{T}} QoI(t)^2}}$$

- $\mathcal{T} = \left\{ t \in \{0, \Delta s, 2\Delta s, \dots\} : t \leq 2 \times 10^{-1} \text{ s} \right\}$

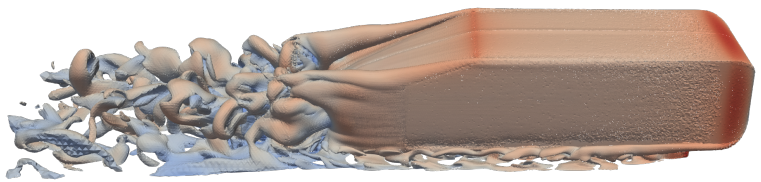
- Offline performance (excluding computation of solution snapshots)

Computational model	k	$\widetilde{N_e}$	$\widetilde{N_e}/N_e$ (%)	ECSW time
PROM $\rightarrow$ HPROM	627	7 389	0.26	7.9 h
QPROM $\rightarrow$ HQPROM	39	544	0.019	1.8 mn

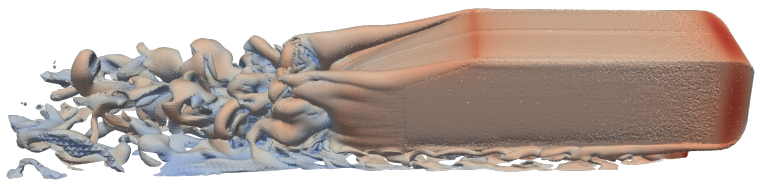
Computational model	k	Wall clock time (**V**)	Wall clock time (**H**)	Wall clock time (total offline)
HPROM	627	16.8 mn	–	9.0 h
HQPROM	39	17.0 mn	9.6 mn	1.7 h [1]

[1] dominated by the cost of GCV for regularization

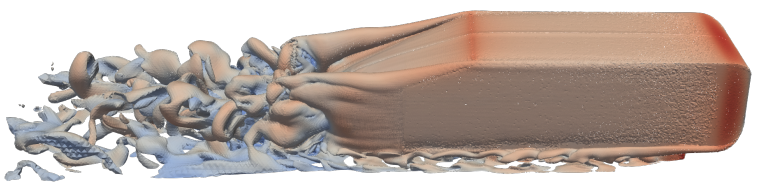
- Online performance (accuracy, qualitative): Visualization at $t = 2 \times 10^{-1}$ s of the predicted iso-vorticity contours colored by the Mach number



HDM (1.7×10^7)

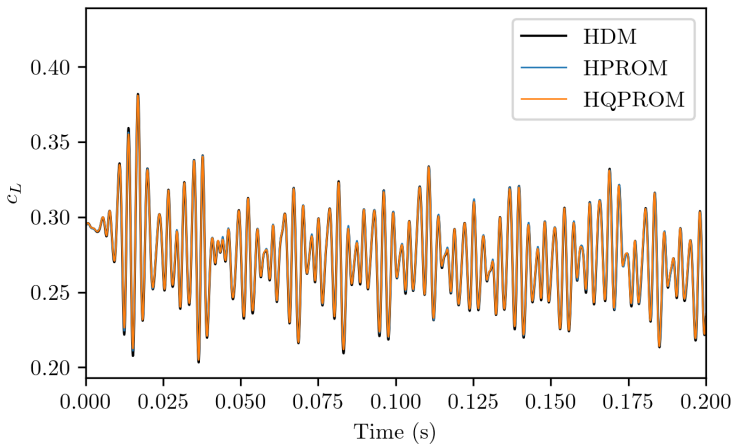


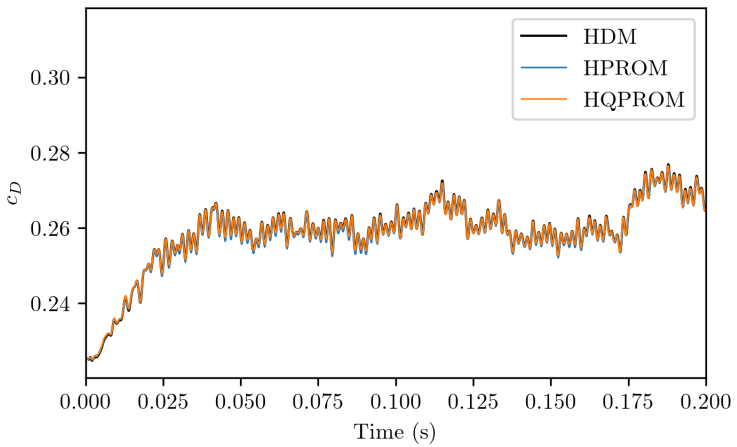
HPROM (627)

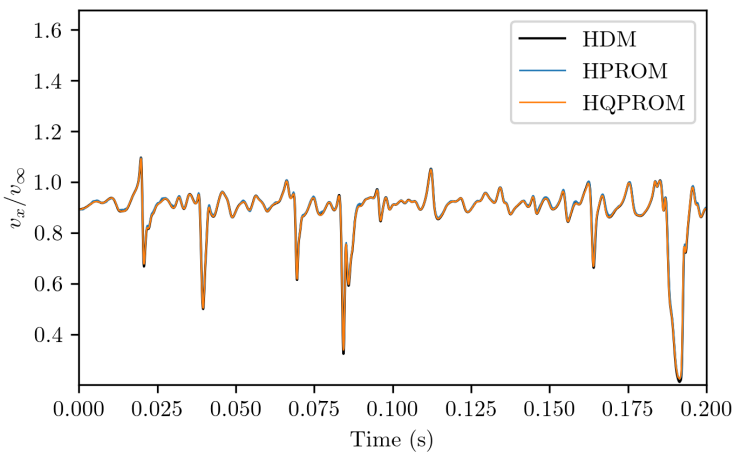


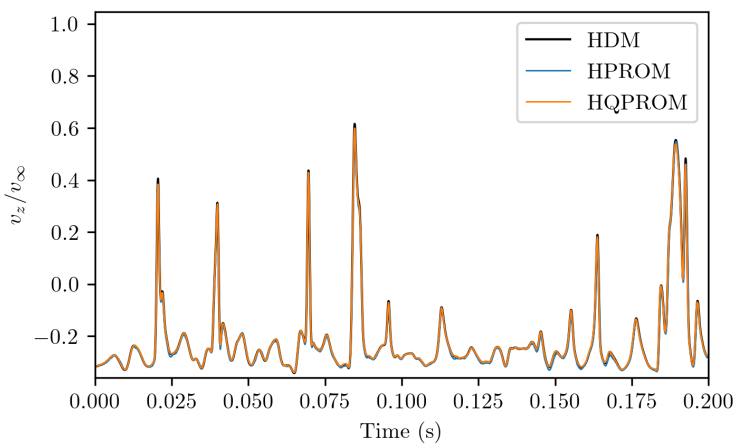
HQPROM (39)

- Online performance (accuracy, qualitative): Predicted time-histories of the lift (top, left) and drag (top, right) coefficients, v_x/v_∞ (bottom, left), and v_z/v_∞ (bottom, right)









- Online performance (accurady and wall-clock time, quantitative)

Computational model	k	$\mathbb{RE}_{c_D}$ (%)	$\mathbb{RE}_{c_L}$ (%)	$\mathbb{RE}_{v_x}$ (%)	$\mathbb{RE}_{v_z}$ (%)
HPROM	627	0.24	0.77	0.83	3.95
HQPROM	39	0.10	0.71	0.54	2.66

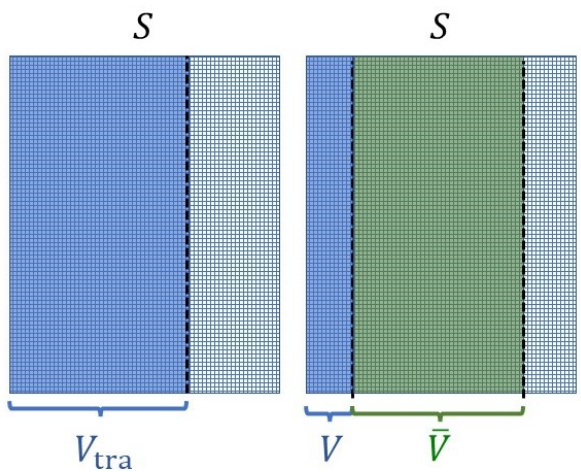
Computational model (number of cores)	k	Wall clock time	Speed-up factor (wall-clock time)	Speed-up factor (CPU time)
HDM (**240**)	–	15.1 h	–	–
HPROM (**8**)	627	3.75 h	4	121
HQPROM (**8**)	39	6.9 mn	131	3 940

Nonlinear approximation manifold generated by a ROB and an artificial neural network (ANN)

$$\boxed{\tilde{\mathbf{w}} = \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q} + \overline{\mathbf{V}}\mathcal{N}(\mathbf{q})}$$

where

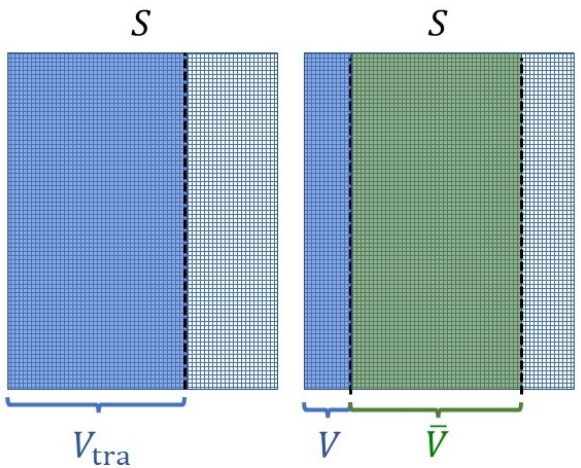
- $\mathbf{V} \in \mathbb{R}^{N \times k}$ is constructed using the first $k \ll N$ columns of $\mathbf{U}$ and $k \ll k_{\text{tra}}$ (looser tolerance ϵ)
- $\overline{\mathbf{V}} \in \mathbb{R}^{N \times \bar{k}}$ is constructed using a subset of the next $\bar{k} \ll N$ columns of $\mathbf{U}$ and $\bar{k} \gg k$
- $\mathbf{V}$ and $\overline{\mathbf{V}}$ satisfy $\mathbf{V}^T\mathbf{V} = \mathbf{I}_k$, $\overline{\mathbf{V}}^T\overline{\mathbf{V}} = \mathbf{I}_{\bar{k}}$, $\mathbf{V}^T\overline{\mathbf{V}} = \mathbf{0}_{k,\bar{k}}$, and $\overline{\mathbf{V}}^T\mathbf{V} = \mathbf{0}_{\bar{k},k}$
- $\mathcal{N}$ is an ANN representing a map $\mathbb{R}^k \to \mathbb{R}^{\bar{k}}$ whose size k_{ANN} scales with $\bar{k} \ll N$
- $\mathbf{q} \in \mathbb{R}^k$ is the vector of generalized coordinates



$$\boxed{\tilde{\mathbf{w}} = \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q} + \overline{\mathbf{V}}\mathcal{N}(\mathbf{q})} \quad (\star)$$

Let $k_{\text{ideal}} \geq k_{\text{tra}}$ denote the *ideal dimension* of $\mathbf{V}_{\text{tra}}$. If $\sum_{\ell > k} \sigma_\ell$ decays slowly, k_{ideal} is unaffordable, and then $(\star)$ expresses the following three-part idea

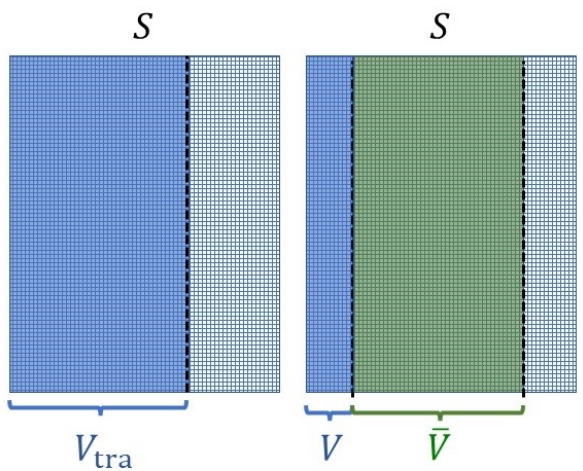
- **Part 1**: construct a ROB $\begin{bmatrix}\mathbf{V} & \overline{\mathbf{V}}\end{bmatrix}$ of dimension $(k + \bar{k}) < k_{\text{ideal}} \ll N$, where $k \ll \bar{k}$
- **Part 2**: split this ROB in two orthogonal ROBs $\mathbf{V} \in \mathbb{R}^{N \times k}$ and $\overline{\mathbf{V}} \in \mathbb{R}^{N \times \bar{k}}$, and construct the affine approximation $\tilde{\mathbf{w}} = \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q} + \overline{\mathbf{V}}\bar{\mathbf{q}}$
- **Part 3**: treat only $\mathbf{q} \in \mathbb{R}^k$ as a vector of generalized coordinates $\Rightarrow$ build a PROM of dimension $k \ll (k + \bar{k})$; and *learn* the dependence of $\bar{\mathbf{q}} \in \mathbb{R}^{\bar{k}}$ on $\mathbf{q} \in \mathbb{R}^k$ $\Rightarrow \bar{\mathbf{q}} = f(\mathbf{q})$, where $f(\mathbf{q}) : \mathbb{R}^k \to \mathbb{R}^{\bar{k}}$ is represented by a deep ANN $\mathcal{N}$



$$\boxed{\tilde{\mathbf{w}} = \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q} + \overline{\mathbf{V}}\underbrace{\mathcal{N}(\mathbf{q})}_{\bar{\mathbf{q}}}} \quad (\star)$$

Interpretations

- Data-driven model of the closure error: $\overline{\mathbf{V}}\bar{\mathbf{q}} = \overline{\mathbf{V}}\mathcal{N}(\mathbf{q}) = \tilde{\mathbf{w}} - (\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q})$
- Variational multi-scale approach
- Nonlinear compression of a ROB and associated PROM: $k \ll (k + \bar{k})$
- Most efficient use of a set of solution snapshots defining a solution manifold: $(\mathbf{q};\ \bar{\mathbf{q}} = f(\mathbf{q}))$



- Let $\mathcal{N}(\mathbf{q};\gamma)$ be the ANN representing the map $f(\mathbf{q}) : \mathbb{R}^k \to \mathbb{R}^{\bar{k}}$, where the vector-valued hyperparameter $\gamma \in \mathbb{R}^{k_{\text{ANN}}}$ and $k_{\text{ANN}} \ll N$
- Construct $\mathcal{N}(\mathbf{q};\gamma)$ such that ideally

$$\mathbf{w}_i = \mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q}_i + \overline{\mathbf{V}}\mathcal{N}\left(\mathbf{q}_i;\gamma\right), \quad i = 1, \cdots, N_{\text{snap}}$$

- From the above and the orthogonality properties of $\mathbf{V}$ and $\overline{\mathbf{V}}$, it follows that

$$\mathbf{q}_i = \mathbf{V}^T\left(\mathbf{w}_i - \mathbf{w}_{\text{ref}}\right) \text{ and } \mathcal{N}\left(\mathbf{q}_i;\gamma\right) = \overline{\mathbf{V}}^T\left(\mathbf{w}_i - \mathbf{w}_{\text{ref}}\right) \equiv \bar{\mathbf{q}}_i,\ i = 1, \cdots, N_{\text{snap}}$$

- Hence

$$\mathbf{q}\,(\text{input}) \to \boxed{\mathcal{N}(\mathbf{q};\gamma)} \to \bar{\mathbf{q}}\,(\text{output})$$

and

$$\gamma = \arg\min_{\gamma'} \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \left(\bar{\mathbf{q}}_i - \mathcal{N}\left(\mathbf{q}_i;\gamma'\right)\right)^2$$

- Residual minimization in the 2-norm

$$\mathbf{q}^{n+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^k} \left\| \mathbf{r}^{n+1}\left(\underbrace{\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{x}^{n+1} + \overline{\mathbf{V}}\mathcal{N}\left(\mathbf{x}^{n+1}\right)}_{\tilde{\mathbf{w}}^{n+1}(\mathbf{x}^{n+1})}, t^{n+1} \right) \right\|_2^2$$

$$\iff \mathbf{q}^{n+1} \text{ sol. of } \mathbf{W}^{n+1^T}\mathbf{r}^{n+1}\left(\underbrace{\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q}^{n+1} + \overline{\mathbf{V}}\mathcal{N}\left(\mathbf{q}^{n+1}\right)}_{\tilde{\mathbf{w}}^{n+1}(\mathbf{q}^{n+1})}, t^{n+1} \right) = \mathbf{0}$$

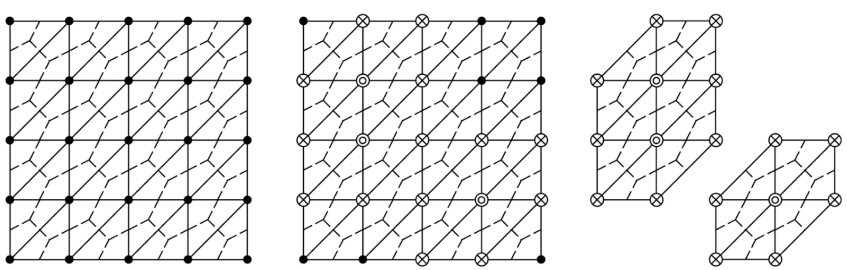
- Hence, for an ANN-LSPG PROM, the *left* ROB is given by

$$\mathbf{W}^{n+1} = \underbrace{\mathbf{J}^{n+1}\mathbf{V}}_{\substack{\textit{traditional left ROB}}} + \underbrace{\mathbf{J}^{n+1}\,\overline{\mathbf{V}}\frac{\partial\mathcal{N}}{\partial\mathbf{q}}\left(\mathbf{q}^{n+1}\right)}_{\substack{\textit{additional term}\\ \textit{due to ANN approximation}}}, \quad \mathbf{J}^{n+1} = \frac{\partial\mathbf{r}}{\partial\tilde{\mathbf{w}}}\left(\tilde{\mathbf{w}}^{n+1}\right) \in \mathbb{R}^{N\times N}$$

- Energy conserving sampling and weighting (ECSW)
  - project-then-approximate rather than approximate-then-project

$$\begin{aligned}\mathbf{r}_k^{n+1}\left(\mathbf{q}^{n+1}, t^{n+1}\right) &= \mathbf{W}^T\mathbf{r}^{n+1}\left(\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q}^{n+1} + \overline{\mathbf{V}}\mathcal{N}\left(\mathbf{q}^{n+1}\right), t^{n+1}\right)\\ &\approx \sum_{e_i\in\widetilde{\mathcal{E}}} \xi^{e_i}\left(\mathbf{L}^{e_i}\mathbf{W}\right)^T \mathbf{r}^{e_i\,n+1}\left(\mathbf{L}^{e_i+}\left[\mathbf{w}_{\text{ref}} + \mathbf{V}\mathbf{q}^{n+1} + \overline{\mathbf{V}}\mathcal{N}\left(\mathbf{q}^{n+1}\right)\right], t^{n+1}\right)\end{aligned}$$

  - recall the interpretation of the ECSW method: cubature where the elements of the reduced mesh $\widetilde{\mathcal{E}} \subset \mathcal{E}$ are the points and $\left\{\xi_{e_1}, \ldots, \xi_{e_{\widetilde{N_e}}}\right\}$ are the corresponding weights
  - implementation (context of the finite volume method): augmented reduced mesh



Augmented reduced mesh: ◎ represents a selected node attached to a selected element; and ⊗ represents an added node to enable the full representation of the computational stencil at the selected node/element

- 2D, parametric, inviscid Bugers' problem

$$\frac{\partial u_x}{\partial t} + \frac{1}{2}\left(\frac{\partial u_x^2}{\partial x} + \frac{\partial\left(u_x u_y\right)}{\partial y}\right) = 0.02\exp(\mu_2 x)$$

$$\frac{\partial u_y}{\partial t} + \frac{1}{2}\left(\frac{\partial\left(u_y u_x\right)}{\partial x} + \frac{\partial u_y^2}{\partial y}\right) = 0$$

$$u_x(x=0,y,t;\boldsymbol{\mu}) = \mu_1$$

$$u_x(x,y,t=0) = u_y(x,y,t=0) = 1$$

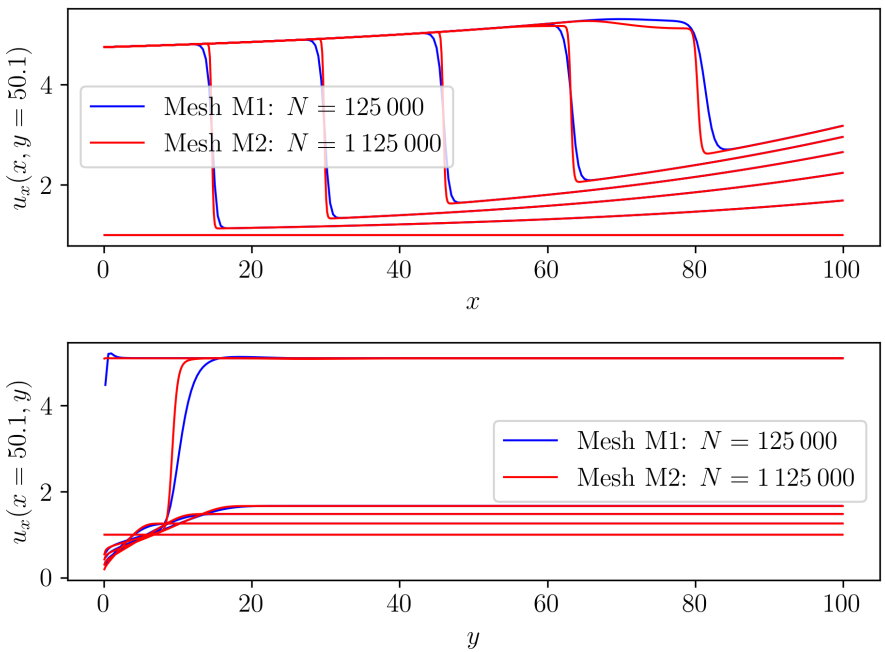
  - computational domain: $(x,y) \in [0,100] \times [0,100]$
  - time-interval: $t \in [0,25]$
  - parameter domain: $\boldsymbol{\mu} = (\mu_1,\mu_2) \in \mathcal{D} = [4.25,5.50] \times [0.015,0.03]$
  - computing system: Linux cluster where 1 node is configured with 2 Intel Xeon Gold 5118 processors clocked at 2.3 GHz and 192 GB of memory

HDMs

- Godunov-type scheme and two uniform meshes
  M1: $N_e = 250 \times 250 \Rightarrow N = 125\,000$
  M2: $N_e = 750 \times 750 \Rightarrow N = 1\,125\,000$
- Temporal discretization: trapezoidal method and constant $\Delta t = 0.05$ ($N_t = 500$ time-steps)
- $u_{\text{ref}} = 0$ (in all cases)
- Measure of the relative error

$$\mathbb{RE}(\boldsymbol{\mu}) = \frac{\sum_{m=0}^{N_t} ||u^m(\boldsymbol{\mu}) - \tilde{u}^m(\boldsymbol{\mu})||_2}{\sum_{m=0}^{N_t} ||u^m(\boldsymbol{\mu})||_2}$$

Far-traveling shock problem

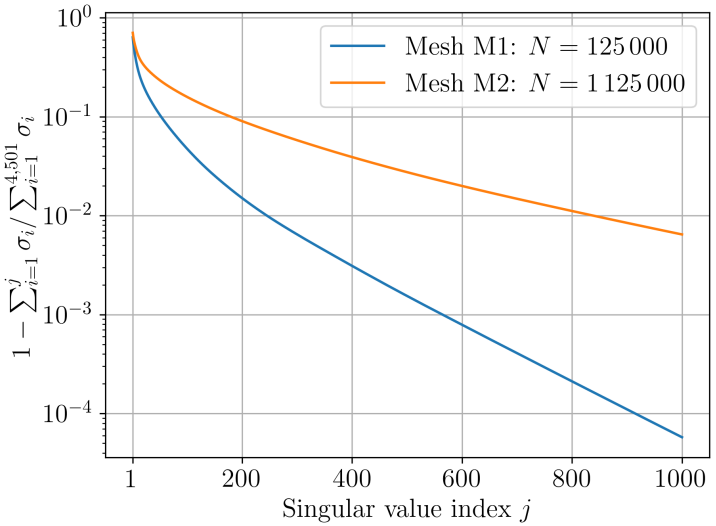


($\boldsymbol{\mu}^\star = (4.75,\ 0.02)$)

Training

- Uniform sampling of $\mathcal{D} = [4.25, 5.50] \times [0.015, 0.03]$ by a 3×3 grid characterized by $\Delta\mu_1 = 0.625$ and $\Delta\mu_2 = 0.0075 \Rightarrow$ 9 training parameter points
- Above sampling leads to $N_s = 4\,501$ solution snapshots, including the initial condition which is nonparametric and therefore shared by all sampled parameter points
- LSPG PROM with $\boldsymbol{n}_{\text{tra}} = \mathbf{95}$

Kolmogorov k_{tra}-width



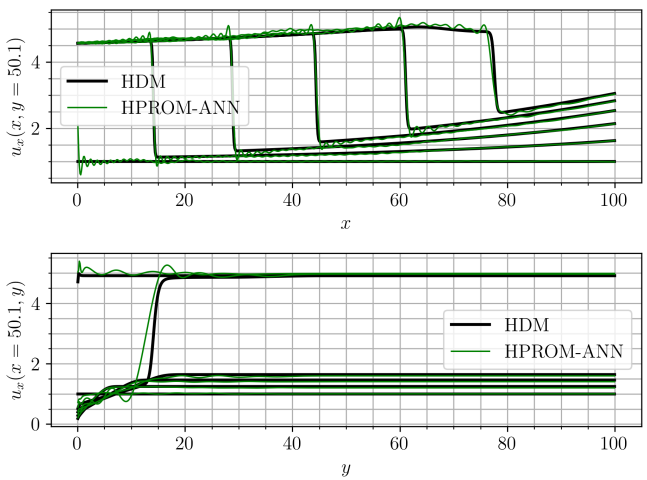
- `PyTorch` for constructing the ANN $\mathcal{N}$ for the map $f(q) : \mathbb{R}^k \to \mathbb{R}^{\bar{k}}$

  $$(n, 32) \underset{\text{ELU}}{\to} (32, 64) \underset{\text{ELU}}{\to} (64, 128) \underset{\text{ELU}}{\to} (128, 256) \underset{\text{ELU}}{\to} (256, 256) \underset{\text{ELU}}{\to} (256, \bar{k})$$

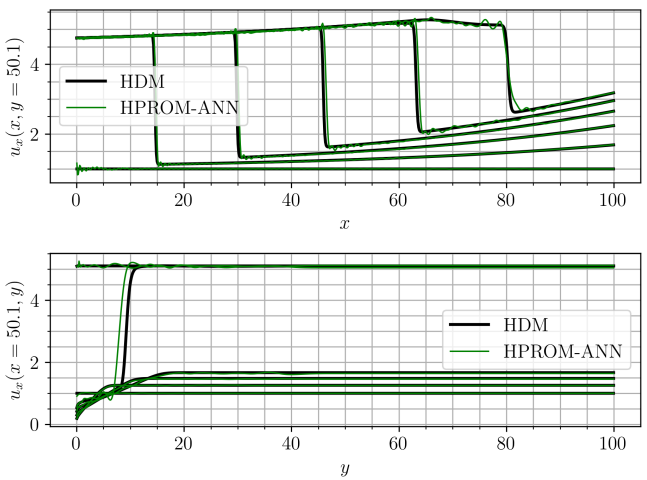
- Exponential linear unit (ELU) activation functions
- 90%-10% training-testing random split of the set of generalized coordinates associated with the $N_s = 4\,501$ collected solution snapshots
- `PyTorch` for computing the gradient $\partial\mathcal{N}/\partial\mathbf{q}$ (forward mode automatic differentiation)
- Mesh M1: $(\boldsymbol{k}, \bar{\boldsymbol{k}}) = (\mathbf{10}, \mathbf{140})$
- Mesh M2: $(\boldsymbol{k}, \bar{\boldsymbol{k}}) = (\mathbf{10}, \mathbf{140})$ and $(\boldsymbol{k}, \bar{\boldsymbol{k}}) = (\mathbf{20}, \mathbf{280})$

- All tests are performed for the following queried but unsampled parameter points
  $\boldsymbol{\mu}^{(1)} = (5.19, 0.026)$
  $\boldsymbol{\mu}^{(2)} = (4.56, 0.019)$
  $\boldsymbol{\mu}^{(3)} = (4.75, 0.020)$
- Maximum relative error $\mathbb{RE}_{\text{max}} = \max_{\boldsymbol{\mu}^{(1)}, \boldsymbol{\mu}^{(2)}, \boldsymbol{\mu}^{(3)}} \mathbb{RE}(\boldsymbol{\mu})$ is reported

- Accuracy results: Case M2, queried but unsampled parameter point $\boldsymbol{\mu}^{(2)} = (4.56, 0.019)$ corresponding to $\mathbb{RE}_{\text{max}}$



$(\boldsymbol{k}, \bar{\boldsymbol{k}}) = (\mathbf{10}, \mathbf{140})$



$(\boldsymbol{k}, \bar{\boldsymbol{k}}) = (\mathbf{20}, \mathbf{280})$

- Offline performance results (wall-clock time): Case M2, queried but unsampled parameter point $\boldsymbol{\mu}^{(2)} = (4.56, 0.019)$ corresponding to $\mathbb{RE}_{\text{max}}$, 1 core except when otherwise specified

Computational model	k (N for HDM)	$\widetilde{N_e}$ (N_e for HDM)	Time (mins)
HDM ($\Delta t = 0.05$)	**1 125 000**	562 500	407.47
HPROM (ECSW, offline, 24 cores)	**95**	63 106	45.25
HPROM-ANN (ANN, offline)	**10**	–	23.40
HPROM-ANN (ECSW, offline)	**10**	3 496	26.40
HPROM-ANN (ANN, offline)	**20**	–	20.80
HPROM-ANN (ECSW, offline, 24 cores)	**20**	22 984	5.21

note: configuration $(\boldsymbol{n}, \bar{\boldsymbol{n}}) = (\mathbf{20}, \mathbf{280})$ uses a single-layer-ANN

- Online performance results (wall-clock time): Case M2, queried but unsampled parameter point $\boldsymbol{\mu}^{(2)} = (4.56, 0.019)$ corresponding to $\mathbb{RE}_{\text{max}}$, 1 core

Computational model	k (N for HDM)	$\widetilde{N}_e$ (N_e for HDM)	$\mathbb{RE}_{\text{max}}$	Time (mins)	Speedup factor
HDM	**1 125 000**	562 500	–	407.47	–
HPROM	**95**	63 106	8.05%	35.08	**11.61**
PROM-ANN	**10**	–	5.50%	23.90	**17.05**
HPROM-ANN	**10**	3 496	3.43%	0.500	**814.94**
PROM-ANN	**20**	–	4.60%	65.13	**6.26**
HPROM-ANN	**20**	22 984	4.72%	8.74	**46.62**

note: configuration $(\boldsymbol{n}, \bar{\boldsymbol{n}}) = (\mathbf{20}, \mathbf{280})$ uses a single-layer-ANN