Interpretable and flexible non-intrusive reduced-order models using reproducing kernel Hilbert spaces

Shane A. McQuarrie
Brigham Young University

Alejandro N. Diaz
John T. Tencer
Patrick J. Blonigan
Sandia National Laboratories

Workshop on Reduced Order and Surrogate Modeling for Digital Twins
IMSI ◇ Chicago ◇ 11 November 2025

BYU Mathematics

Sandia National Laboratories

Projection-based intrusive model reduction

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}\big(\mathbf{q}(t)\big) \qquad \mathbf{q}(t) \in \mathbb{R}^{n_q}$$ LARGE

$\mathbf{q}(t) \approx \mathbf{g}\big(\tilde{\mathbf{q}}(t)\big)$ approximation

$\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^{n_q}$ decompression

$\mathbf{h} : \mathbb{R}^{n_q} \to \mathbb{R}^r$ compression

$\mathbf{g} \circ \mathbf{h} : \mathbb{R}^{n_q} \to \mathbb{R}^{n_q}$ projection

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{q}}(t) = \mathbf{h}'(\mathbf{g}(\tilde{\mathbf{q}}(t)))\mathbf{f}(\mathbf{g}(\tilde{\mathbf{q}}(t))) \qquad \tilde{\mathbf{q}}(t) \in \mathbb{R}^{r}$$ small

Requirements

1) **Data** to choose the projection map
2) **Direct access** to original system dynamics

Projection-based intrusive model reduction

$$\boxed{\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}}$$

POD Basis

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}\big(\mathbf{q}(t)\big) = \mathbf{A}\mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)] \qquad \mathbf{q}(t) \in \mathbb{R}^{n_q}$$

LARGE

$$\mathbf{q}(t) \approx \mathbf{g}(\tilde{\mathbf{q}}(t)) = \mathbf{V}\tilde{\mathbf{q}}(t) \qquad \mathbf{h}(\mathbf{q}(t)) = \mathbf{V}^\mathsf{T}\mathbf{q}(t) \qquad (\mathbf{g} \circ \mathbf{h})(\mathbf{q}(t)) = \mathbf{V}\mathbf{V}^\mathsf{T}\mathbf{q}(t)$$

approximation decompression compression projection

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{q}}(t) &= \mathbf{V}^\mathsf{T}\mathbf{f}\big(\mathbf{V}\tilde{\mathbf{q}}(t)\big) = \mathbf{V}^\mathsf{T}\mathbf{A}\mathbf{V}\tilde{\mathbf{q}}(t) + \mathbf{V}^\mathsf{T}\mathbf{H}(\mathbf{V} \otimes \mathbf{V})[\tilde{\mathbf{q}}(t) \otimes \tilde{\mathbf{q}}(t)] \\ &= \tilde{\mathbf{A}}\tilde{\mathbf{q}}(t) + \tilde{\mathbf{H}}[\tilde{\mathbf{q}}(t) \otimes \tilde{\mathbf{q}}(t)]\end{aligned} \qquad \tilde{\mathbf{q}}(t) \in \mathbb{R}^{r}$$

small

projection preserves polynomial structure

<u>Requirements</u>

1) **Data** to choose the matrix $\mathbf{V}$
2) **Direct access** to matrices $\mathbf{A},\ \mathbf{H}$

Regression-based non-intrusive model reduction

no direct access

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}\big(\mathbf{q}(t)\big)$$

$$\mathbf{q}(t) \approx \mathbf{g}\big(\hat{\mathbf{q}}(t)\big)$$
approximation

$$\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^{n_q}$$
decompression

$$\mathbf{h} : \mathbb{R}^{n_q} \to \mathbb{R}^r$$
compression

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}\big(\hat{\mathbf{q}}(t)\big)$$

regression

$$\hat{\mathbf{f}} = \operatorname*{arg\,min}_{\mathbf{s}\in S} \left\{ \sum_{j=1}^{n_t} \left\| \dot{\hat{\mathbf{q}}}_j - \mathbf{s}\big(\hat{\mathbf{q}}_j\big) \right\|_2^2 + \mathcal{R}(\mathbf{s}) \right\}$$

Function space

$$\hat{\mathbf{q}}_j = \mathbf{h}(\mathbf{q}(t_j))$$
state data

$$\dot{\hat{\mathbf{q}}}_j = \frac{\hat{\mathbf{q}}_j - \hat{\mathbf{q}}_{j-1}}{t_j - t_{j-1}} \approx \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{h}(\mathbf{q}(t))\Big|_{t=t_j}$$
time derivative data

Requirements

1) **Data** to choose the projection **and** drive the regression
2) **Structure specification** for the reduced-order model

Regression-based non-intrusive model reduction

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}\big(\mathbf{q}(t)\big) = \mathbf{A}\mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t) \otimes \mathbf{q}(t)]$$

$$\mathbf{q}(t) \approx \mathbf{g}(\hat{\mathbf{q}}(t)) = \mathbf{V}\hat{\mathbf{q}}(t)$$

approximation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}\big(\hat{\mathbf{q}}(t)\big) = \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)]$$

Function space: $S(\hat{\mathbf{A}}, \hat{\mathbf{H}}) = \left\{\hat{\mathbf{q}} \mapsto \hat{\mathbf{A}}\hat{\mathbf{q}} + \hat{\mathbf{H}}[\hat{\mathbf{q}} \otimes \hat{\mathbf{q}}]\right\}$

$$\hat{\mathbf{A}}, \hat{\mathbf{H}} = \operatorname*{arg\,min}_{\hat{\mathbf{A}},\hat{\mathbf{H}}} \left\{ \sum_{j=1}^{n_t} \left\| \dot{\hat{\mathbf{q}}}_j - \left(\hat{\mathbf{A}}\hat{\mathbf{q}}_j + \hat{\mathbf{H}}[\hat{\mathbf{q}}_j \otimes \hat{\mathbf{q}}_j]\right) \right\|_2^2 + \lambda_1^2 \left\|\hat{\mathbf{A}}\right\|_F^2 + \lambda_2^2 \left\|\hat{\mathbf{H}}\right\|_F^2 \right\}$$

operator inference regression

Regression-based non-intrusive model reduction

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{f}\big(\mathbf{q}(t)\big)$$

$$\mathbf{q}(t) \approx \mathbf{g}(\hat{\mathbf{q}}(t))$$

approximation

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{f}}\big(\hat{\mathbf{q}}(t)\big) = \boldsymbol{\Omega}^{\mathsf{T}} K\big(\hat{\mathbf{Q}}, \hat{\mathbf{q}}(t)\big)$$

Reproducing kernel Hilbert space (RKHS): $S = \mathcal{H}_K^r = \mathcal{H}_K \times \mathcal{H}_K \times \cdots \times \mathcal{H}_K$

$$\hat{\mathbf{f}} = \operatorname*{arg\,min}_{\mathbf{s}\in\mathcal{H}_K^r} \left\{ \sum_{j=1}^{n_t} \left\| \dot{\hat{\mathbf{q}}}_j - \mathbf{s}\big(\hat{\mathbf{q}}_j\big) \right\|_2^2 + \lambda \left\|\mathbf{s}\right\|_{H_K^r}^2 \right\}$$

regularized kernel interpolation

Non-intrusive model reduction: comparison

	Interpretable ROM structure mirrors FOM	**Flexible** approximate arbitrary structure	**Error Estimates** certified ROM solutions
Neural networks	No	Yes	No*
Operator inference	Yes	No	No*
Kernel ROMs	Yes	Yes	Yes*

Regularized kernel interpolation

Kernel function

- $K : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}$
- $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$
- if $\mathbf{X} = [\, \mathbf{x}_1 \quad \cdots \quad \mathbf{x}_m \,]$, then $K(\mathbf{X}, \mathbf{X}) = [K(\mathbf{x}_i, \mathbf{x}_j)]$ positive semi-definite

Reproducing kernel Hilbert space (RKHS)

$$\mathcal{H}_K = \overline{\operatorname{span}\{K(\mathbf{x}, \cdot\,) \,:\, \mathbf{x} \in \mathbb{R}^{n_x}\}}$$

Representer Theorem

$$\underbrace{\mathbf{X} = [\, \mathbf{x}_1 \quad \cdots \quad \mathbf{x}_m \,]}_{\text{input data}} \in \mathbb{R}^{n_x \times m}, \qquad \underbrace{\mathbf{Y} = [\, \mathbf{y}_1 \quad \cdots \quad \mathbf{y}_m \,]}_{\text{output data}} \in \mathbb{R}^{n_y \times m}, \qquad \underbrace{\mathbf{x}_j \mapsto \mathbf{y}_j}_{\text{data pairs}}$$

$$\underbrace{\min_{\mathbf{s} \in \mathcal{H}_K^{n_y}} \left\{ \sum_{j=1}^{m} \|\mathbf{y}_j - \mathbf{s}(\mathbf{x}_j)\|_2^2 + \gamma \|\mathbf{s}\|_{\mathcal{H}_K^{n_y}}^2 \right\}}_{\text{regularized kernel interpolation}} \longrightarrow \boxed{\begin{gathered} \mathbf{s}(\mathbf{x}) = \mathbf{\Omega}^{\mathsf{T}} K(\mathbf{X}, \mathbf{x}) \\ \text{where} \\ (K(\mathbf{X}, \mathbf{X}) + \gamma \mathbf{I})\mathbf{\Omega} = \mathbf{Y}^{\mathsf{T}} \end{gathered}}$$

Regularized kernel interpolation

$$\mathbf{s}(\mathbf{x}) = \boldsymbol{\Omega}^\mathsf{T} K(\mathbf{X}, \mathbf{x})$$

where

$$\underbrace{\big(K(\mathbf{X}, \mathbf{X}) + \gamma\mathbf{I}\big)}_{m \times m}\boldsymbol{\Omega} = \mathbf{Y}^\mathsf{T}$$

$m \times m$ ($m =$ # of training points)

Kernel selection to design interpolant structure

General purpose: radial basis function (RBF) kernel

$$K(\mathbf{x}, \mathbf{x}') = \underbrace{\psi(\epsilon\|\mathbf{x} - \mathbf{x}'\|_2)}_{\text{RBF}}$$

$$\mathbf{s}(\mathbf{x}) = \boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\psi}_\epsilon(\mathbf{x}) = \boldsymbol{\Omega}^\mathsf{T}\begin{bmatrix} \psi(\epsilon\|\mathbf{x}_1 - \mathbf{x}\|_2) \\ \vdots \\ \psi(\epsilon\|\mathbf{x}_m - \mathbf{x}\|_2) \end{bmatrix}$$

Known structure: feature map kernel

$$K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\mathsf{T}\mathbf{G}\boldsymbol{\phi}(\mathbf{x}')$$

$$\underbrace{\boldsymbol{\phi} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_\phi}}_{\text{feature map}}, \qquad \underbrace{\mathbf{G} \in \mathbb{R}^{n_\phi \times n_\phi}}_{\text{weight matrix}}$$

$$\mathbf{s}(\mathbf{x}) = \boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\phi}(\mathbf{X})^\mathsf{T}\mathbf{G}\boldsymbol{\phi}(\mathbf{x}) = \mathbf{C}\boldsymbol{\phi}(\mathbf{x})$$

(precomputable)

Hybrid: feature map + RBF

$$K(\mathbf{x}, \mathbf{x}') = \underbrace{c_\phi\boldsymbol{\phi}(\mathbf{x})^\mathsf{T}\mathbf{G}\boldsymbol{\phi}(\mathbf{x}')}_{\text{feature map part}} + \underbrace{c_\psi\psi(\epsilon\|\mathbf{x} - \mathbf{x}'\|_2)}_{\text{RBF part}}$$

$$\mathbf{s}(\mathbf{x}) = \underbrace{\mathbf{C}\boldsymbol{\phi}(\mathbf{x})}_{\text{feature map}} + \underbrace{c_\psi\boldsymbol{\Omega}^\mathsf{T}\boldsymbol{\psi}_\epsilon(\mathbf{x})}_{\text{RBF}}$$

Regularized kernel interpolation for **Kernel ROMs**

$$\hat{\mathbf{Q}} = [\ \hat{\mathbf{q}}_1 \quad \cdots \quad \hat{\mathbf{q}}_{n_t}\ ] \in \mathbb{R}^{r \times n_t},$$
input data (states)

$$\dot{\hat{\mathbf{Q}}} = [\ \dot{\hat{\mathbf{q}}}_1 \quad \cdots \quad \dot{\hat{\mathbf{q}}}_{n_t}\ ] \in \mathbb{R}^{r \times n_t},$$
output data (ddts)

$$\hat{\mathbf{q}}_j \mapsto \dot{\hat{\mathbf{q}}}_j$$
data pairs

Quadratic feature map $\phi(\hat{\mathbf{q}}) = \begin{bmatrix} \hat{\mathbf{q}} \\ \hat{\mathbf{q}} \otimes \hat{\mathbf{q}} \end{bmatrix}$, Gaussian RBF $\psi(x) = \exp(-x^2)$

for example

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) &= \hat{\mathbf{C}}\phi(\hat{\mathbf{q}}(t)) + c_\psi \boldsymbol{\Omega}^{\mathsf{T}} \psi_\epsilon(\hat{\mathbf{q}}(t)) \\ &= \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] + c_\psi \boldsymbol{\Omega}^{\mathsf{T}} \begin{bmatrix} \psi(\epsilon \|\hat{\mathbf{q}}_1 - \hat{\mathbf{q}}(t)\|_2) \\ \vdots \\ \psi(\epsilon \|\hat{\mathbf{q}}_{n_t} - \hat{\mathbf{q}}(t)\|_2) \end{bmatrix}\end{aligned}$$

Design based on **FOM structure** and the **approximation / projection**

Implementation details

Scaling: scale inputs, select weight matrix $\mathbf{G}$

$$K_{\boldsymbol{\nu}}(\mathbf{x}, \mathbf{x}') = K(\boldsymbol{\nu}(\mathbf{x}), \boldsymbol{\nu}(\mathbf{x}'))$$

$$\mathbf{G} = \begin{bmatrix} \frac{1}{\|\hat{\mathbf{Q}}\|_F}\mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \frac{1}{\|\hat{\mathbf{Q}}\|_F^2}\mathbf{I}_{r^2} \end{bmatrix}$$

Regularization: select λ

$$\hat{\mathbf{f}} = \operatorname*{arg\,min}_{\mathbf{s}\in\mathcal{H}_K^r} \left\{ \sum_{j=1}^{n_t} \left\| \dot{\hat{\mathbf{q}}}_j - \mathbf{s}(\hat{\mathbf{q}}_j) \right\|_2^2 + \lambda \|\mathbf{s}\|_{H_K^r}^2 \right\}$$

Kernel tuning: select ϵ and c_ψ

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{q}}(t) = \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}[\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)] + c_\psi \boldsymbol{\Omega}^{\mathsf{T}} \begin{bmatrix} \psi(\epsilon\|\hat{\mathbf{q}}_1 - \hat{\mathbf{q}}(t)\|_2) \\ \vdots \\ \psi(\epsilon\|\hat{\mathbf{q}}_{n_t} - \hat{\mathbf{q}}(t)\|_2) \end{bmatrix}$$

A posteriori error bounds for Kernel ROMs

Theorem. If $\hat{\mathbf{f}}$ is an unregularized kernel interpolant of $\tilde{\mathbf{f}} + \boldsymbol{\delta} \in \mathcal{H}_K^r$ where $\|\boldsymbol{\delta}(\hat{\mathbf{q}}(s))\|_{\mathbf{M}} < \delta(s)$, then

$$\|\mathbf{q}(t) - \mathbf{g}(\hat{\mathbf{q}}(t))\|_{\mathbf{M}} \leq \int_0^t (\alpha_p(s) + \alpha_K(s)) e^{\int_s^t \beta(\tau) d\tau} ds + e^{\int_0^t \beta(\tau) d\tau} \|\mathbf{q}(t) - \mathbf{g}(\mathbf{h}(\mathbf{q}(0)))\|_{\mathbf{M}}$$

= error from low-dimensional **state approximation**

+ error between **intrusive ROM** and kernel interpolant

+ error in **time derivative approximation**

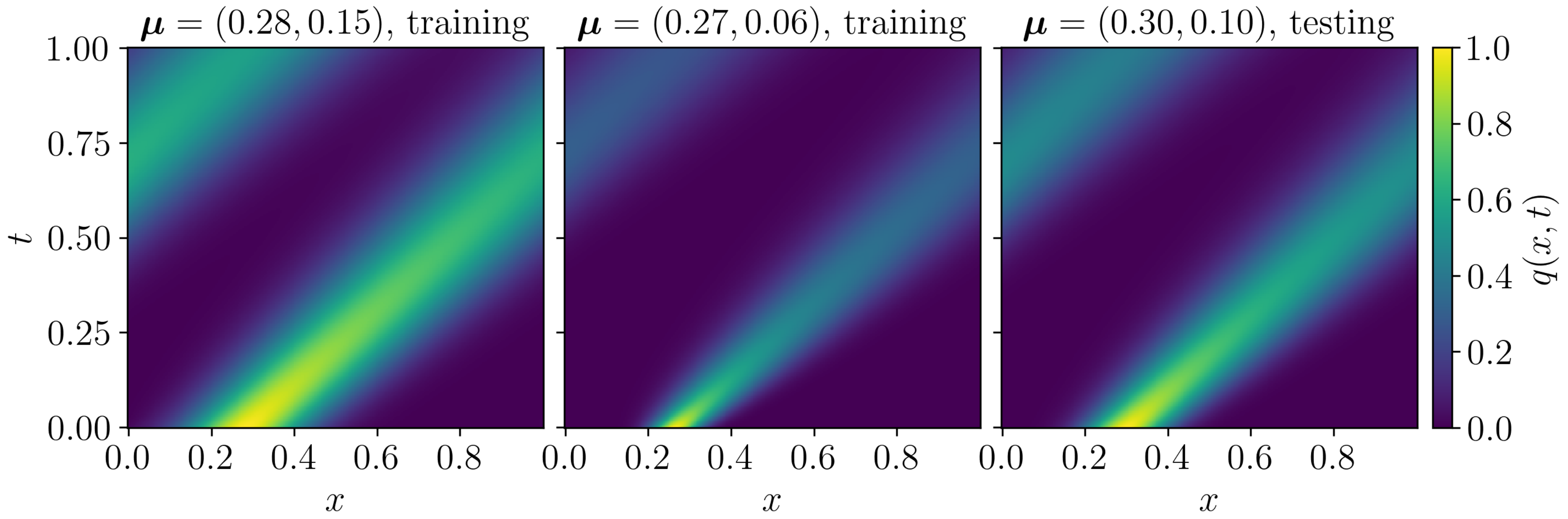
+ **initial condition** error

Error bound **computable** but requires **intrusive evaluations**

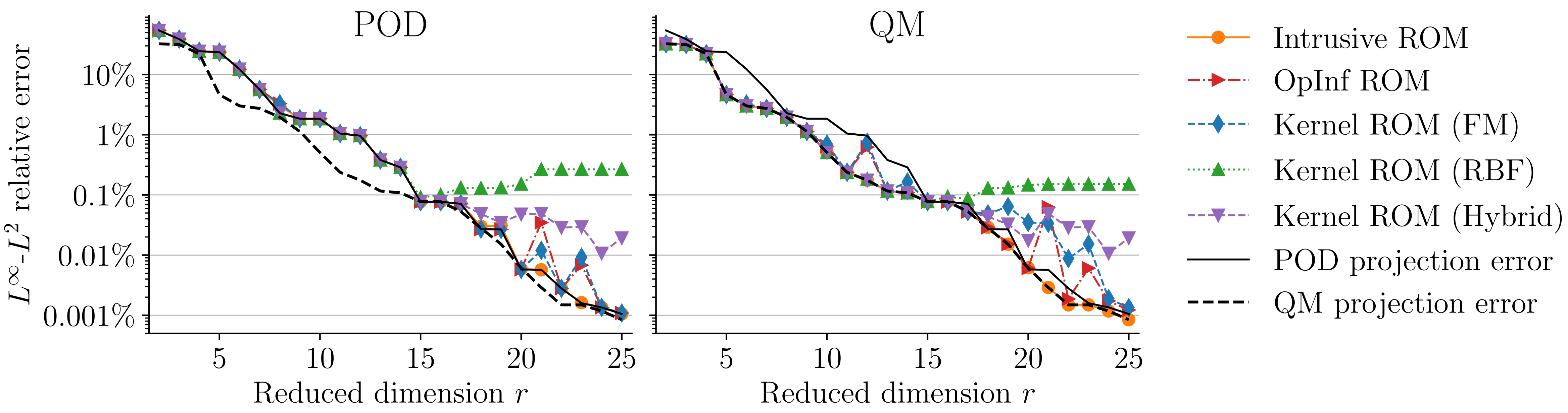
Linear example: Advection-diffusion

$$
\begin{aligned}
\frac{\partial}{\partial t}q(x,t) - \kappa\frac{\partial^2}{\partial x^2}q(x,t) + \beta\frac{\partial}{\partial x}q(x,t) &= 0, && x \in (0,1), \quad t \in (0,T), \\
q(0,t) = q(1,t), \quad \frac{\partial}{\partial x}q(0,t) &= \frac{\partial}{\partial x}q(1,t), && t \in (0,T), \\
q(x,0) = q_0(x;\boldsymbol{\mu}) &= e^{-(x-\mu_1)^2/\mu_2^2}, && x \in (0,1).
\end{aligned}
$$

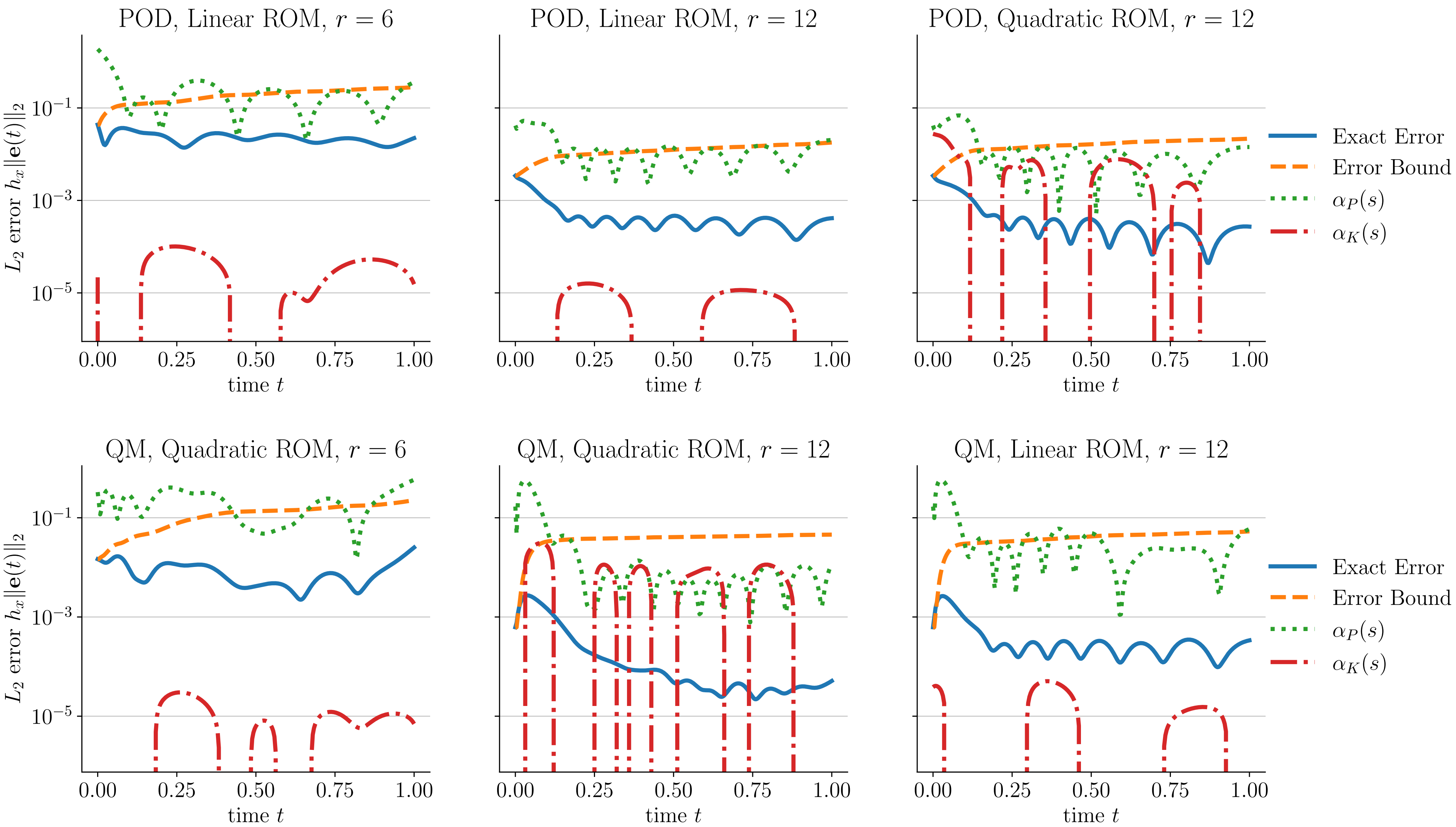
$$
\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{A}\mathbf{q}(t)
$$



Linear example: Advection-diffusion



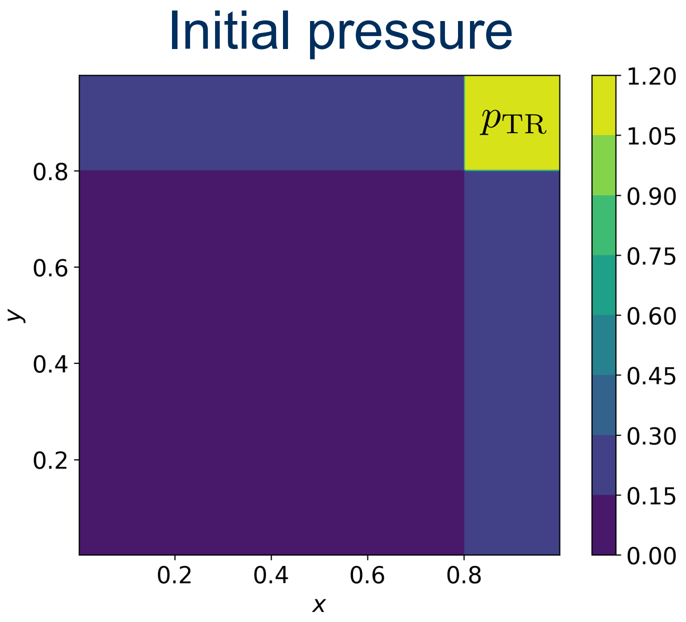
Linear example: Advection-diffusion



Challenging example: 2D Euler–Riemann problem

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho u \\ \rho v \\ \rho E\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho u \\ \rho u^2 + p \\ \rho u v \\ (E+p)u\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v \\ \rho u v \\ \rho v^2 + p \\ (E+p)v\end{bmatrix} = 0$$

$$p = (\gamma - 1)\big(\rho E - \tfrac{1}{2}\rho(u^2 + v^2)\big),$$



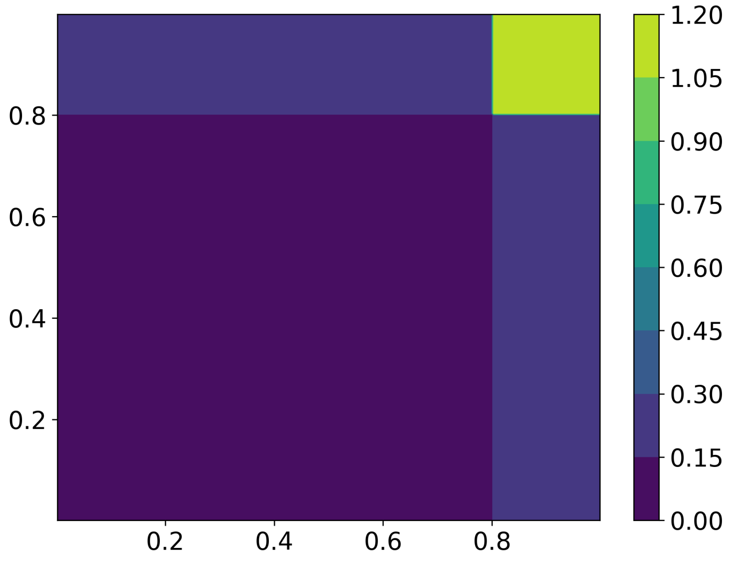
Experimental setup

- Parameterized by initial top-right pressure. Train: $p_{TR} \in \{0.5, 0.75, 1.0, 1.25, 1.5\}$, Test: $p_{TR} = 1.125$
- CCFV discretization, 256×256 uniform grid, SSP3 time stepping scheme, $\delta t = 10^{-3}$, $n_t = 800$
- Lift to specific volume form → quadratic model
  Kernel ROM with quadratic feature map

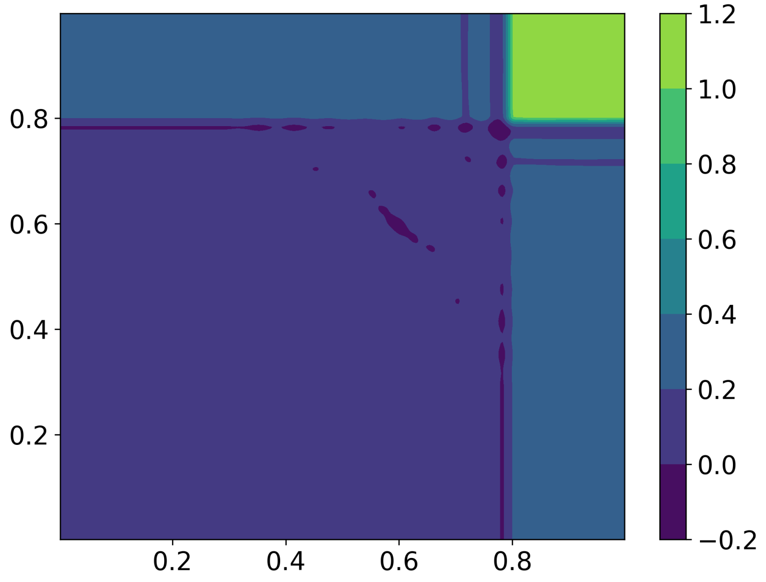
$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(t) = \mathbf{A}\mathbf{q}(t) + \mathbf{H}[\mathbf{q}(t)\otimes\mathbf{q}(t)], \quad \mathbf{q}(0) = \mathbf{q}_0(p_{\mathrm{TR}})$$

Challenging example: 2D Euler–Riemann problem

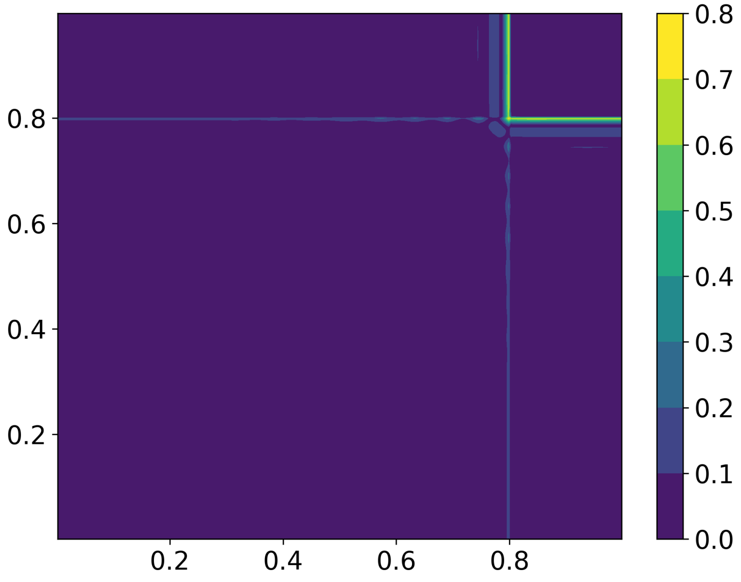
FOM pressure
$n_x = 262{,}144$



ROM pressure
$r = 20$



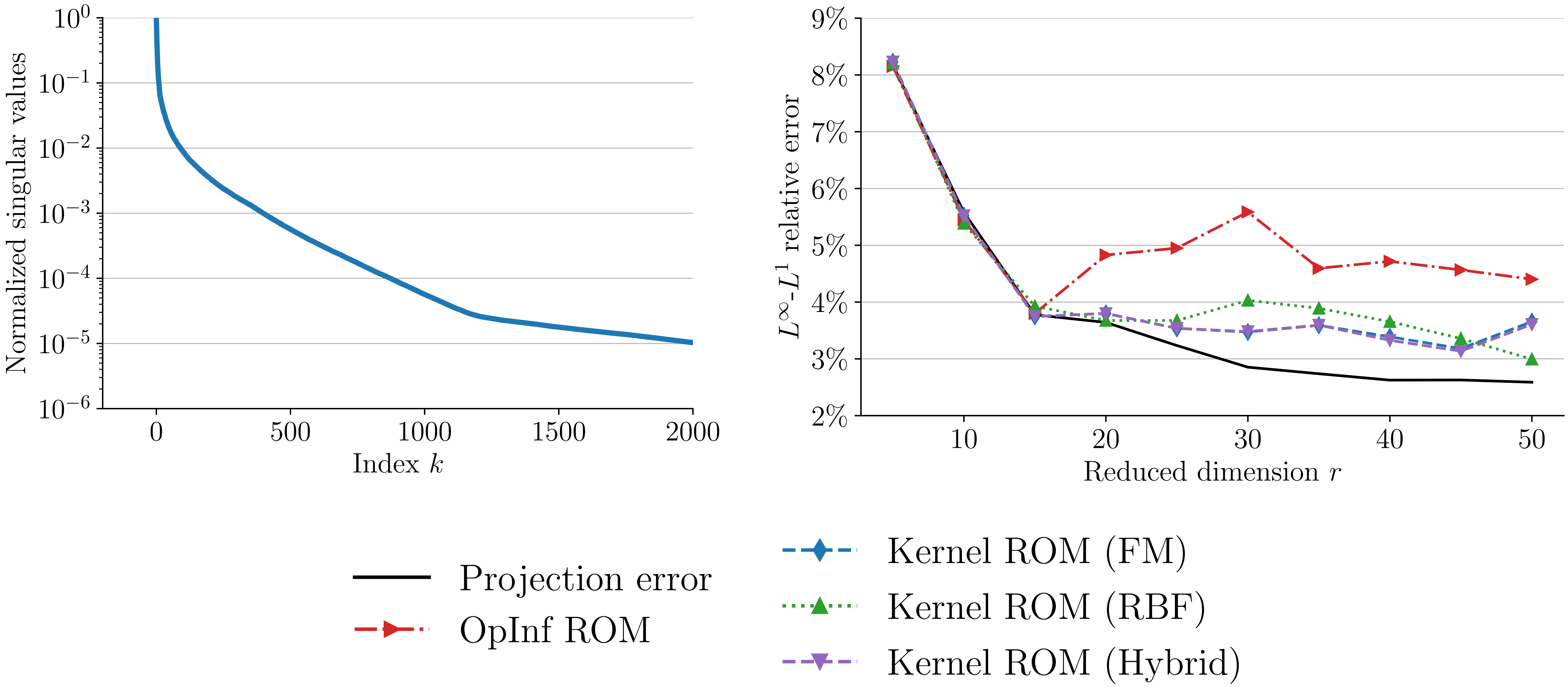
Pressure error
$r = 20$



Relative L_∞-L_1 error $= 3.65\%$

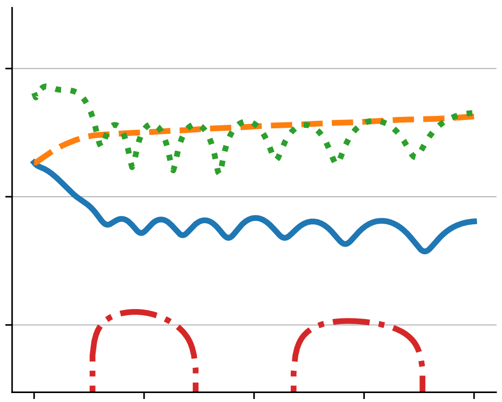
Relative speed up $= 344.8\times$

Challenging example: 2D Euler–Riemann problem



Summary

- **Non-intrusive** ROM construction using regularized kernel interpolation with RKHS
- Kernel ROMs are **flexible** but can be designed with **interpretable** structure
- RKHS theory gives *a posteriori* **error bounds** based on estimates from intrusive projection



arXiv

Future Work

- **Parametric** kernels for parametric problems
- Kernel tuning / selection, feature preservation
- Intelligent selection of **training** data
- Fully non-intrusive **error bounds**