

Model Reduction for Advection dominated (Hyperbolic) problems



Davide Torlo, Monica Nonino

MathLab, Mathematics Area, SISSA International
School for Advanced Studies, Trieste, Italy
davidetorlo.it

Vienna - 10th May 2023

Model Order Reduction for Advection Dominated (Hyperbolic) Problems

First part

- Calibration of solutions
- ALE formulation
- PODEIM-Greedy algorithm
- One parameter transformation maps
- Regression of calibration

Torlo, Davide. "Model reduction for advection dominated hyperbolic problems in an ALE framework: Offline and online phases." arXiv preprint arXiv:2003.13735 (2020).

Second part

- Multiple shocks moving
- Calibration optimization
- Sod shock tube test problem

Work in progress with Monica Nonino

Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Motivation: parametric hyperbolic systems

$$\begin{cases} \partial_t u(x, t, \mu) + \nabla \cdot F(u, x, t, \mu) = 0, & x \in D, t \in \mathbb{R}^+, \mu \in \mathcal{P} \subset \mathbb{R}^P \\ \mathbf{B}(u, \mu) = g(t, \mu) \\ u(x, t = 0, \mu) = u_0(x, \mu) \end{cases} \quad (1)$$

- F non linear dependence on μ !
- μ can influence boundaries, flux, initial conditions
- Why: many physical applications (fluid equations, kinetic models, etc.)
- Classical solvers: FV, FEM, FD, RD. (Slow for high-resolution)
- Many query task (UQ, optimization, etc.)

MOR: Ingredients

- Discretized solution $u_{\mathcal{N}}(\cdot, t, \mu) \in \mathbb{V}_{\mathcal{N}}$ for $t \in \mathbb{R}^+$, $\mu \in \mathcal{P}$
- Solution manifold: $\mathcal{S} := \{u_{\mathcal{N}}(\cdot, t, \mu) \in \mathbb{V}_{\mathcal{N}} : t \in \mathbb{R}^+, \mu \in \mathcal{P}\}$
- Ansatz:

$$\mathcal{S} \approx \mathbb{V}_{N_{RB}} \subset \mathbb{V}_{\mathcal{N}}, \quad N_{RB} \ll \mathcal{N} \quad (2)$$

- Example: diffusion equation $u_t + \mu u_{xx} = 0$ with $u_0 = \sin(x\pi)$

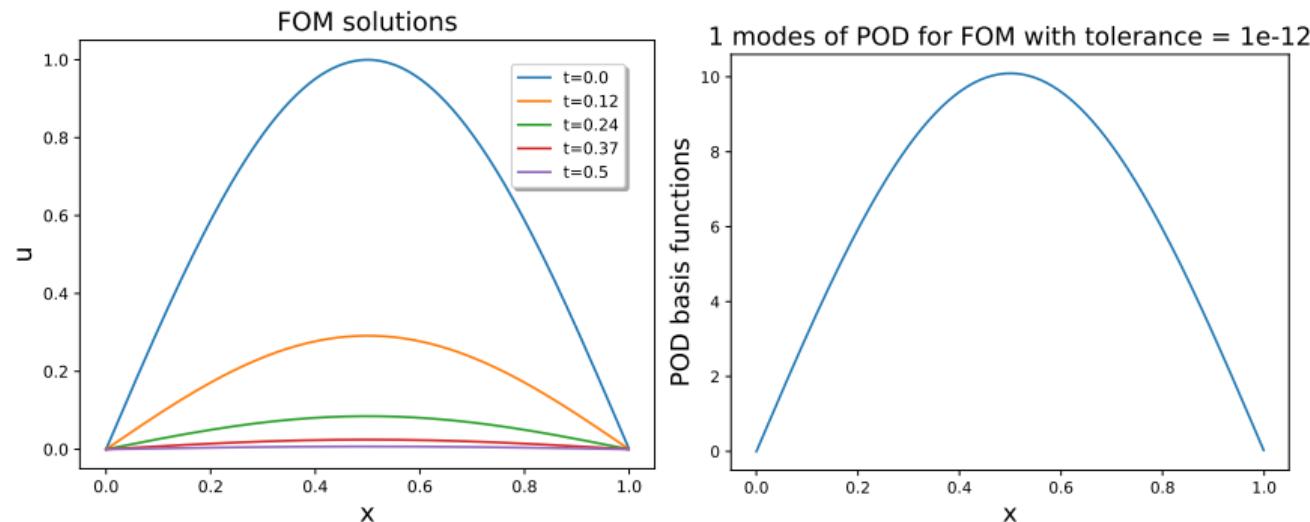


Figure: POD on a diffusion problem

MOR: Ingredients

Problem:

$$U^{n+1}(\mu) - U^n(\mu) + \mathcal{L}^n(U^n, \mu) = 0, \quad U^n, U^{n+1} \in \mathbb{V}_{\mathcal{N}} \quad (3)$$

Objective:

$$\sum_{i=1}^{N_{RB}} u_i^{n+1}(\mu) \psi_{RB}^i - u_i^n(\mu) \psi_{RB}^i + \sum_{i=1}^{N_{RB}} L^i(u^n, \mu) \psi_{RB}^i = 0, \quad (4)$$
$$\psi_{RB}^i \in \mathbb{V}_{\mathcal{N}}, u^n, u^{n+1} \in \mathbb{V}_{N_{RB}}$$

- EIM \Rightarrow non-linear fluxes and scheme $L^i(u^n, \mu)$
- POD \Rightarrow create the RB space and span the time evolution
- Greedy \Rightarrow span the parameter space

Proper orthogonal decomposition (POD)

POD

INPUT:

- Collection of functions $\{f_j\}_{j=1}^N$

OUTPUT:

- Reduced basis spaces $\mathbb{V}_{N_{RB}} = \arg \min_{U | \dim(U) = N_{POD}} \sum_{j=1}^N \|f_j - \mathcal{P}_U(f_j)\|_2$

ALGORITHM:

- Based on SVD of matrix $\{f_j\}_{j=1}^N$
- We obtain (ordered) singular values and vectors
- Retain most energetic (largest singular value)
- Prescribed tolerance to stop the algorithm or maximum number of basis
- Related singular vectors gives $\mathbb{V}_{N_{RB}}$
- Global optimizer of the problem

Greedy algorithm

Greedy algorithm

INPUT:

- Collection of functions $\{f_j\}_{j=1}^N$
- Cheap error estimator $\varepsilon_{RB}(f) \sim \|f - \mathcal{P}_{RB}(f)\|$

OUTPUT:

- Reduced basis space $\mathbb{V}_{N_{RB}}$

ALGORITHM:

- Iteratively choose the worst represented function $f^{worst} = \arg \max_f \varepsilon_{RB}(f)$
- Add f^{worst} to the $\mathbb{V}_{N_{RB}}$ space
- Stop up to a certain tolerance
- Not globally optimal, but locally optimal at each iteration (Greedy)

Empirical interpolation method (EIM)

Empirical interpolation method (EIM)

INPUT:

- $\mathcal{L}^n(U^n, \mu, t^n)$, for $\mu \in \mathcal{P}_h$, $n \leq N_t$

OUTPUT:

- $EIM = (\tau_k, f_k)_{k=1}^{N_{EIM}}$ where functions $f_k \in \mathbb{R}^{\mathcal{N}}$ and $\tau_k \in (\mathbb{R}^{\mathcal{N}})'$
 (τ_k, f_k) are often called magic points and functions, where τ_k are function evaluations

ALGORITHM:

- Greedy iterative procedure
- At each step chooses the worst approximated function via an error (estimator)
$$\mathcal{L}^{worst} = \arg \max_{\mathcal{L}} \|\mathcal{L} - \sum_{k=1}^{N_{EIM}} \tau_k(\mathcal{L}) f_k\|$$
- Maximise the functional τ on the function \mathcal{L}^{worst} $\tau^{chosen} = \arg \max_{\tau} |\tau(\mathcal{L}^{worst})|$
- $EIM = EIM \cup (\tau^{chosen}, \mathcal{L}^{worst})$
- Stop when error is smaller than a tolerance

PODEIM–Greedy

INITIALIZATION:

- EIM on $\mathcal{L}(U^n, \mu_0, t^n)$ for $n \leq N_t$
- $\mathbb{V}_{N_{RB}} = POD(\{U^n(\mu_0)\}_{n=0}^{N_t})$

ITERATION:

- Greedy algorithm spanning over the parameter space \mathcal{P}_h , with an error indicator $\varepsilon(\mathbf{U}(\mu))$ where $\mathbf{U}(\mu) \in \mathbb{R}^N \times \mathbb{R}^+$
- Choose worst parameter as $\mu^* = \arg \max_{\mu \in \mathcal{P}_h} \varepsilon(\mathbf{U}(\mu))$
- Apply POD on time evolution of selected solution $POD_{add} = POD(\{U^n(\mu^*)\}_{n=1}^{N_t})$
- Update the $\mathbb{V}_{N_{RB}}$ with $\mathbb{V}_{N_{RB}} = POD(\mathbb{V}_{N_{RB}} \cup POD_{add})$
- Update EIM basis function with $EIM_{space} = EIM_{space} \cup EIM(\{\mathcal{L}(U^n, \mu^*, t^n)\}_{n=0}^{N_t})$

¹B. Haasdonk and M. Ohlberger, in Hyperbolic problems: theory, numerics and applications, vol. 67, Amer. Math. Soc., 2009.

Reduced Order Model system

Solve the smaller system:

$$\sum_{i=1}^{N_{RB}} (u_i^{n+1}(\mu) - u_i^n(\mu)) \psi_{RB}^i + \sum_{i=1}^{N_{RB}} \sum_{j=1}^{N_{EIM}} \tau_j(\mathcal{L}(U^n, \mu)) \Pi_{RB,i}(f_j) \psi_{RB}^i = 0$$

- $\Pi_{RB,i}(f_j)$ are the projection on $\mathbb{V}_{N_{RB}}$ of the EIM functions: offline
- $\tau_j(\mathcal{L}(U^n, \mu))$ are inexpensive to compute, but depend on the method (for RD $\approx \mathcal{O}(d)$)
- MOR cost $\mathcal{O}(N_t N_{RB} N_{EIM})$ vs FOM cost $\mathcal{O}(N_t \mathcal{N})$
- Gain if $N_{RB}, N_{EIM} \ll \mathcal{N}$
- Error estimator

Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Travelling wave, time evolution solution

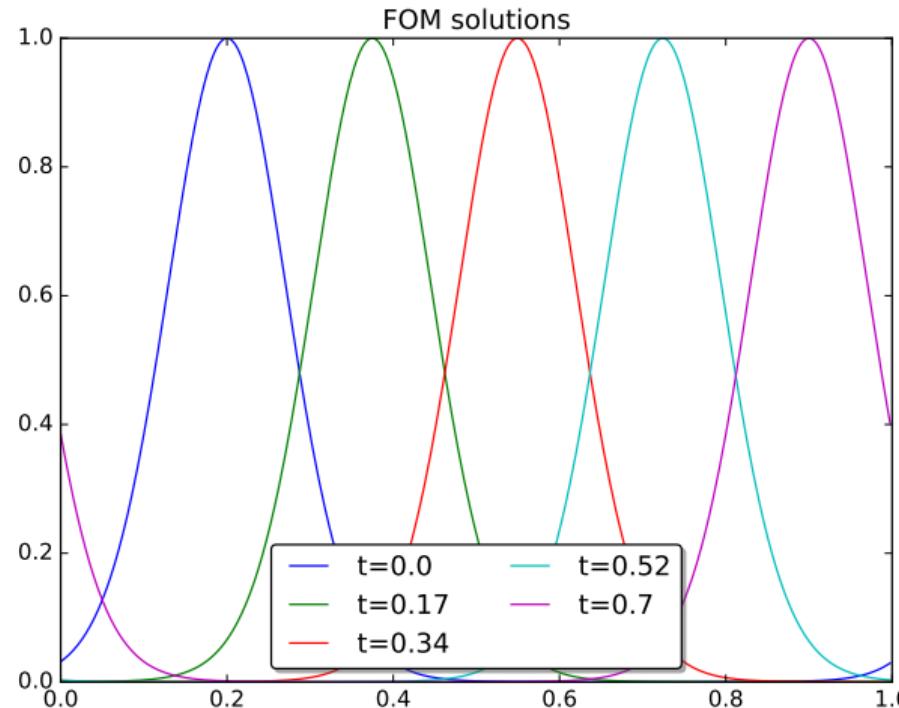


Figure: Solution of advection equation $\partial_t u + \partial_x u = 0$ with gaussian IC

Travelling wave, POD

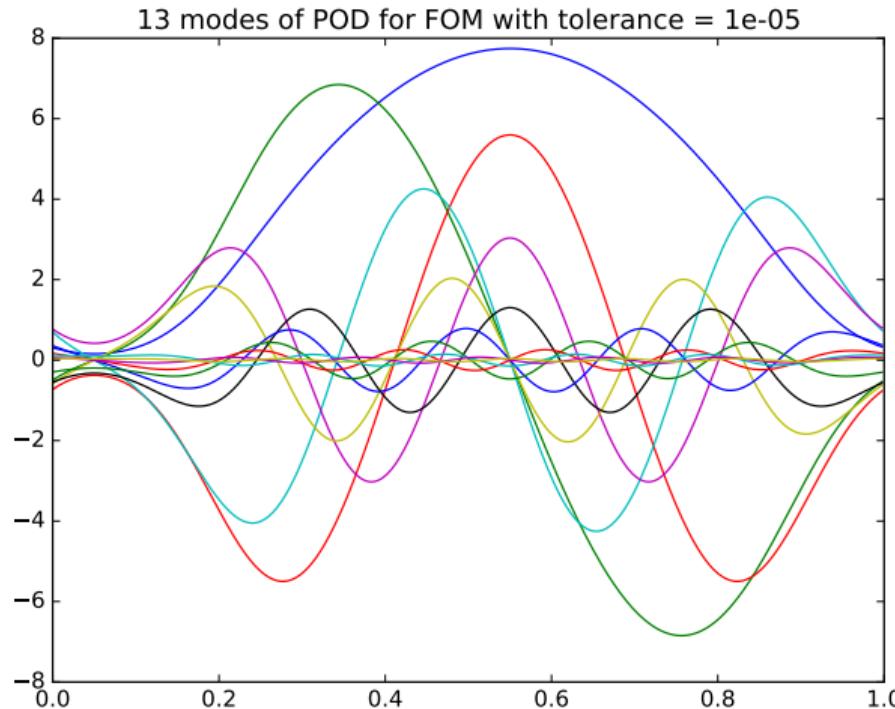


Figure: Solution of advection equation with wave IC

Travelling shock, time evolution solution, little diffusion

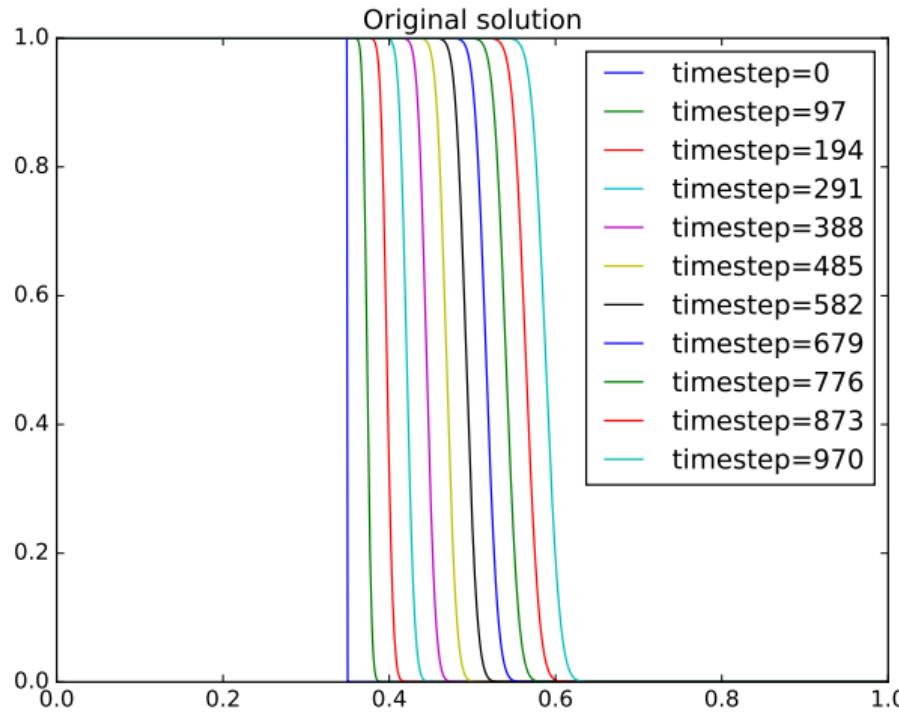


Figure: Solution of advection equation with shock IC

Travelling shock, POD, little diffusion

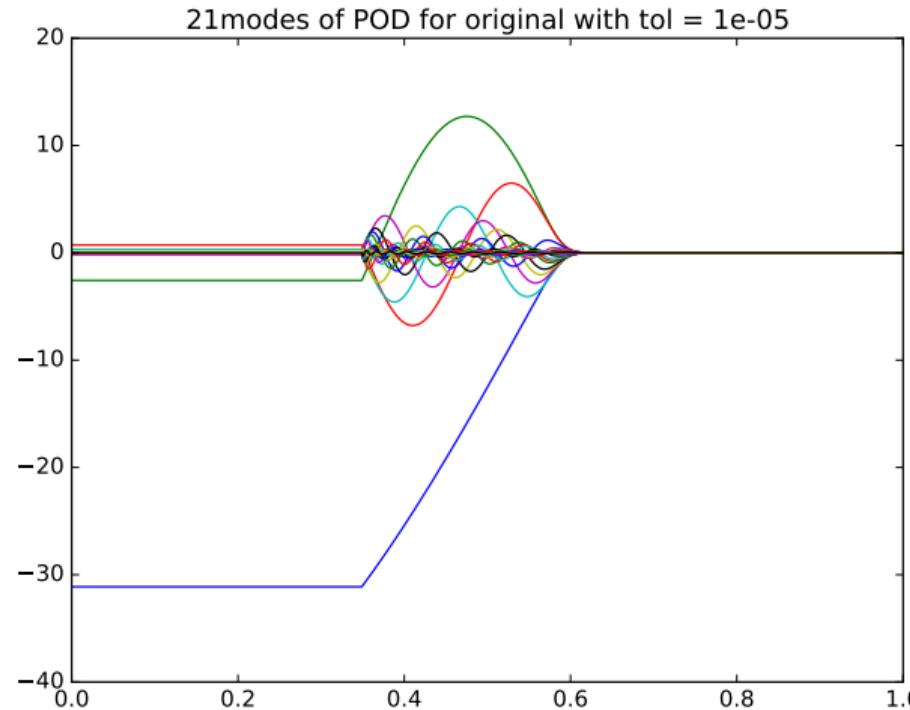


Figure: POD of time evolution of advection equation with shock IC

Travelling shock, time evolution solution, no diffusion

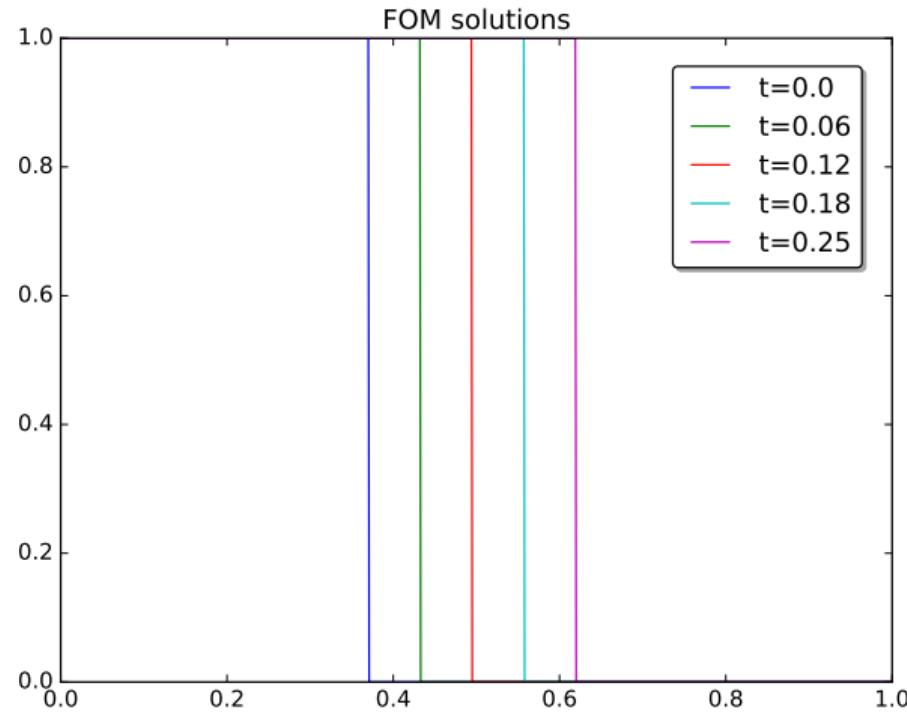


Figure: Solution of advection equation with shock IC

Travelling shock, POD, no diffusion

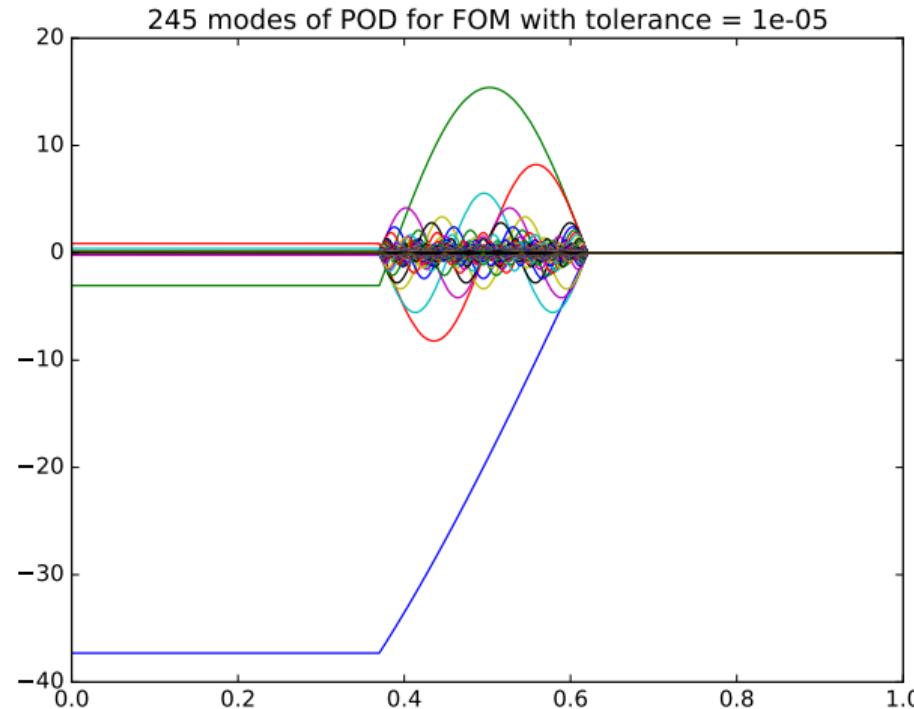


Figure: POD of time evolution of advection equation with shock IC

Common problems and properties

- As many basis functions as positions of the shock
- Slow decay of Kolmogorov N -width

$$d_N(\mathcal{S}, \mathbb{V}) := \inf_{\mathbb{V}_N \subset \mathbb{V}} \sup_{f \in \mathcal{S}} \inf_{g \in \mathbb{V}_N} \|f - g\|$$

- Non linear dependency leads to big EIM and RB space
- 1/2 parameters problem (highly non linear dependence on parameters)

Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Solutions or partial solutions

Some possibilities to incorporate the advection into RB framework

- Freezing **Ohlberger, M. and Rave, S.**
- Shifted POD **Reiss, J., Schulze, P., Sesterhenn, J. and Mehrmann, V.**
- Lagrangian basis method **Mojgani, R. and Balajewicz, M.**
- Advection modes by optimal mass transport **Iollo, A. and Lombardi, D.**
- Calibration (also 2D non-periodic boundaries) **Cagniart, N., Stamm, B. and Maday, Y., Crisovan, R. and Abgrall, R.**
- Online adaptive bases and samplings **Peherstorfer, B.**
- Transport Reversal **Rim, D., Moe, S. and LeVeque R. J.**
- Registration method **Taddei, T.**
- Preprocessing reduced basis **Karatzas, E., Nonino, M., Ballarin, F., Rozza, G. and Maday, Y.**
- Manifold learning via Neural Network **Carlberg, K. and Lee, K.; Lye, K., Mishra S. and Ray, D.; Fresca, S., Dedè, L. and Manzoni, A.**
- Dynamic Modes **Lu, H. and Tartakovsky, D. M.**

Solutions or partial solutions

Some possibilities to incorporate the advection into RB framework

- Freezing **Ohlberger, M. and Rave, S.**
- Shifted POD **Reiss, J., Schulze, P., Sesterhenn, J. and Mehrmann, V.**
- Lagrangian basis method **Mojgani, R. and Balajewicz, M.**
- Advection modes by optimal mass transport **Iollo, A. and Lombardi, D.**
- Calibration (also 2D non-periodic boundaries) **Cagniart, N., Stamm, B. and Maday, Y., Crisovan, R. and Abgrall, R.**
- Online adaptive bases and samplings **Peherstorfer, B.**
- Transport Reversal **Rim, D., Moe, S. and LeVeque R. J.**
- Registration method **Taddei, T.**
- Preprocessing reduced basis **Karatzas, E., Nonino, M., Ballarin, F., Rozza, G. and Maday, Y.**
- Manifold learning via Neural Network **Carlberg, K. and Lee, K.; Lye, K., Mishra S. and Ray, D.; Fresca, S., Dedè, L. and Manzoni, A.**
- Dynamic Modes **Lu, H. and Tartakovsky, D. M.**

Solutions or partial solutions

Some possibilities to incorporate the advection into RB framework

- Freezing **Ohlberger, M.** and **Rave, S.**
- Shifted POD **Reiss, J., Schulze, P., Sesterhenn, J.** and **Mehrmann, V.**
- Lagrangian basis method **Mojgani, R.** and **Balajewicz, M.**
- Advection modes by optimal mass transport **Iollo, A.** and **Lombardi, D.**
- Calibration (also 2D non-periodic boundaries) **Cagniart, N., Stamm, B.** and **Maday, Y., Crisovan, R.** and **Abgrall, R.**
- Online adaptive bases and samplings **Peherstorfer, B.**
- Transport Reversal **Rim, D., Moe, S.** and **LeVeque R. J.**
- Registration method **Taddei, T.**
- Preprocessing reduced basis **Karatzas, E., Nonino, M., Ballarin, F., Rozza, G.** and **Maday, Y.**
- Manifold learning via Neural Network **Carlberg, K.** and **Lee, K.; Lye, K., Mishra S.** and **Ray, D.; Fresca, S., Dedè, L.** and **Manzoni, A.**
- Dynamic Modes **Lu, H.** and **Tartakovsky, D. M.**

Solutions or partial solutions

Some possibilities to incorporate the advection into RB framework

- Freezing **Ohlberger, M.** and **Rave, S.**
- Shifted POD **Reiss, J., Schulze, P., Sesterhenn, J.** and **Mehrmann, V.**
- Lagrangian basis method **Mojgani, R.** and **Balajewicz, M.**
- Advection modes by optimal mass transport **Iollo, A.** and **Lombardi, D.**
- Calibration (also 2D non-periodic boundaries) **Cagniart, N., Stamm, B.** and **Maday, Y., Crisovan, R.** and **Abgrall, R.**
- Online adaptive bases and samplings **Peherstorfer, B.**
- Transport Reversal **Rim, D., Moe, S.** and **LeVeque R. J.**
- Registration method **Taddei, T.**
- Preprocessing reduced basis **Karatzas, E., Nonino, M., Ballarin, F., Rozza, G.** and **Maday, Y.**
- Manifold learning via Neural Network **Carlberg, K.** and **Lee, K.; Lye, K., Mishra S.** and **Ray, D.; Fresca, S., Dedè, L.** and **Manzoni, A.**
- Dynamic Modes **Lu, H.** and **Tartakovsky, D. M.**

Transfomation of the domain

Geometry map T

$$T : \Theta \times \mathcal{R} \rightarrow \Omega \quad (5)$$

- $T(\cdot, \cdot) \in \mathcal{C}^1(\Theta \times \mathcal{R}, \Omega)$,
- $\exists T^{-1} : \Theta \times \Omega \rightarrow \mathcal{R}$ such that $T^{-1}(\theta, T(\theta, y)) = y$ for $y \in \mathcal{R}$ and $T(\theta, T^{-1}(\theta, x)) = x$ for $x \in \Omega$,
- $T^{-1}(\cdot, \cdot) \in \mathcal{C}^1(\Theta \times \Omega, \mathcal{R})$.

Calibration map θ

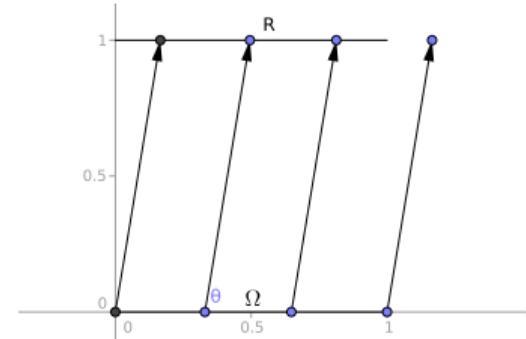
$$\theta : \mathcal{P} \times [0, t_f] \rightarrow \Theta$$

- $\theta(\cdot, \mu) \in \mathcal{C}^1([0, t_f], \Theta)$ for all $\mu \in \mathcal{P}$,
- $u_N(T(\theta(t, \mu), y), t, \mu) \approx \bar{v}(y)$, $\forall \mu \in \mathcal{P}, t \in [0, t_f], y \in \mathcal{R}$

Transformation map for MOR

Examples: θ is the point of maximum height or of steepest solution.

- Translation: $T(\theta, y) = y + \theta - 0.5$
 $T^{-1}(\theta, x) = x - \theta + 0.5$



Transformation map for MOR

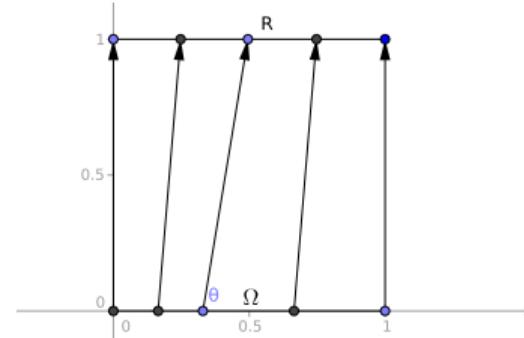
Examples: θ is the point of maximum height or of steepest solution.

- Translation: $T(\theta, y) = y + \theta - 0.5$

$$T^{-1}(\theta, x) = x - \theta + 0.5$$

- Dilatation:

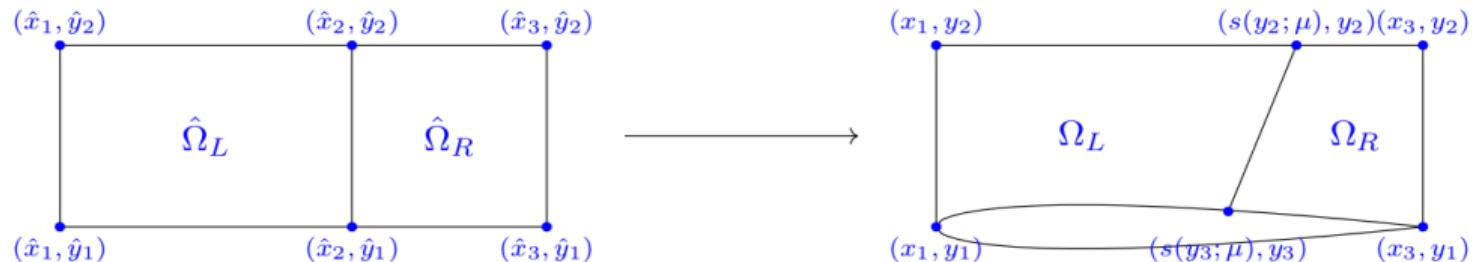
$$T(\theta, y) = \frac{y\theta}{(2\theta-1)y+1-\theta} \quad T^{-1}(\theta, x) = \frac{x(\theta-1)}{(2\theta-1)x-\theta}$$



Transformation map for MOR

Examples: θ is the point of maximum height or of steepest solution.

- Translation:
- Dilatation:
- Higher degree polynomials
- Gordon-Hall



Transformation examples

Translation (for periodic BC): $T^{-1}(\mu, x) = x - \theta(\mu, t) + 0.5$

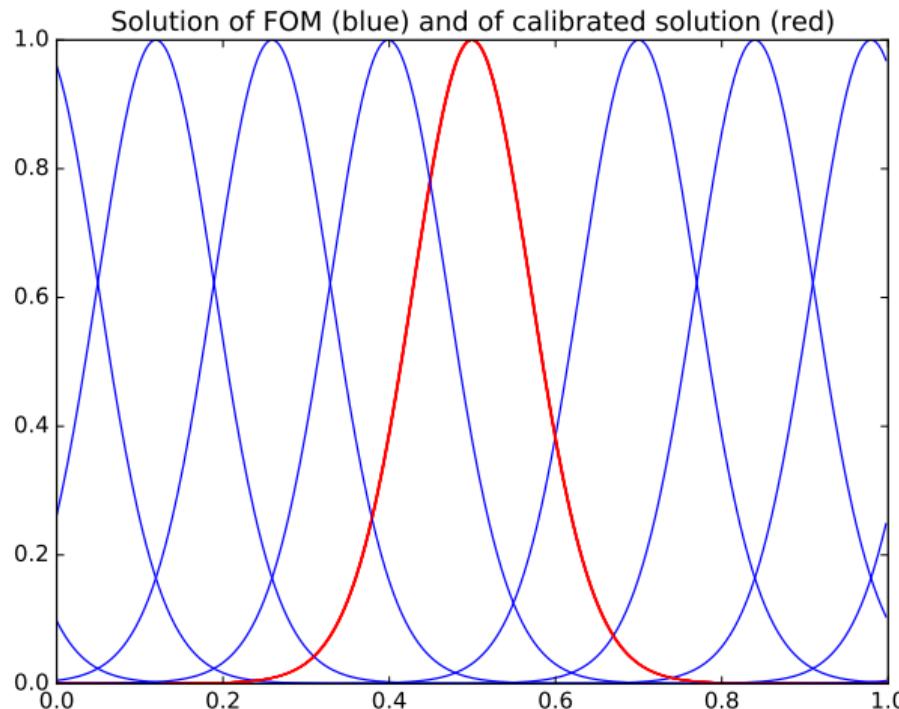


Figure: Calibrated and original solutions for traveling wave

POD of calibrated solutions

Translation (for periodic BC): $T^{-1}(\mu, x) = x - \theta(\mu, t) + 0.5$

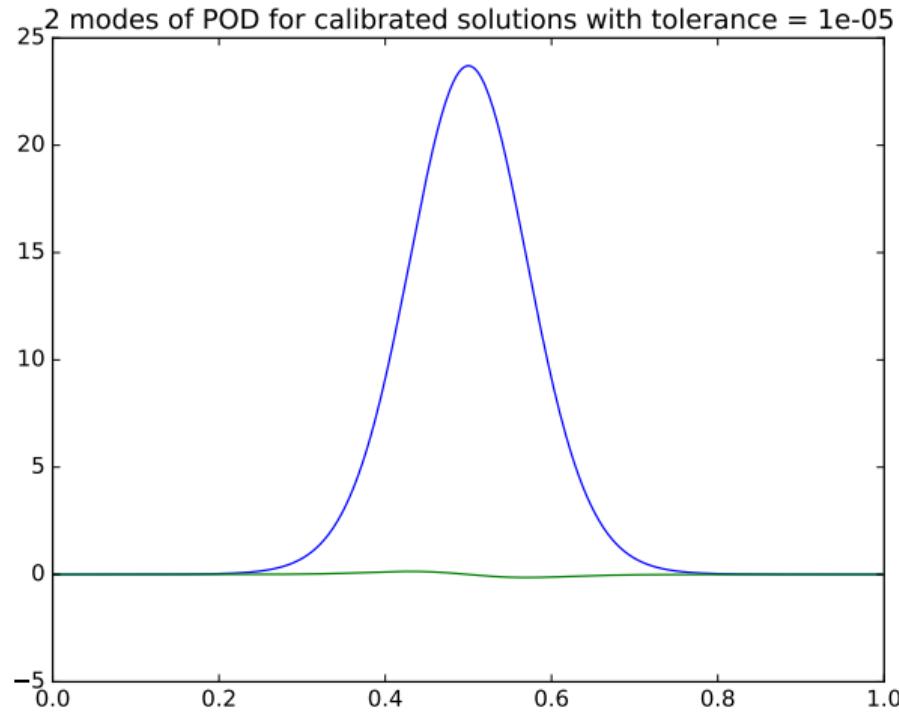


Figure: POD of calibrated solutions for traveling wave

Transformation examples

Dilatation (for other BCs): $T^{-1}(\theta, x) = x \frac{\theta-1}{(2\theta-1)x-\theta}$

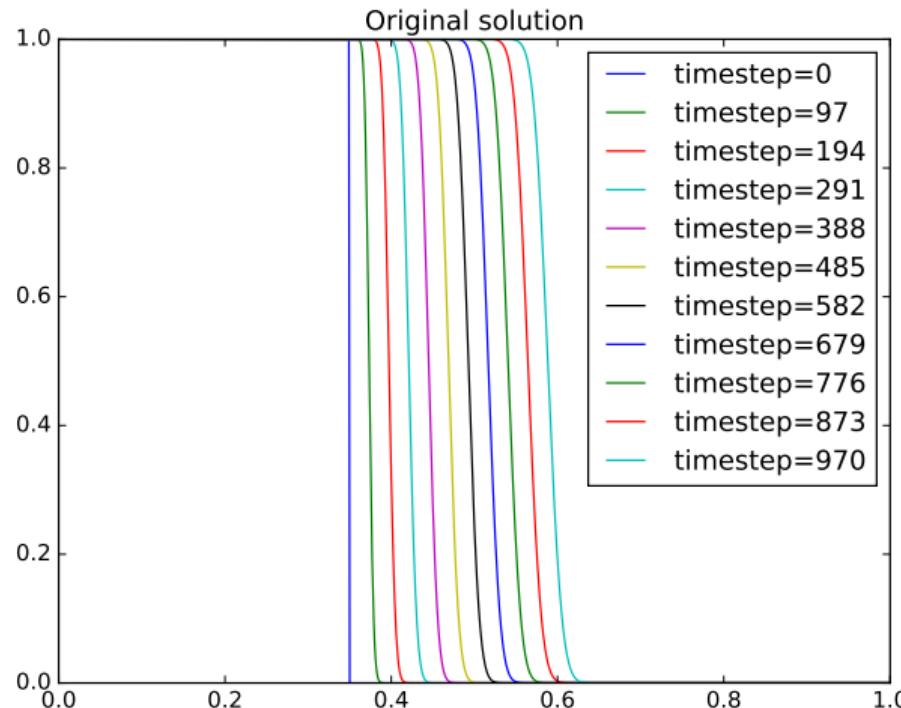


Figure: Original solutions for traveling shock

Transformation examples

Dilatation (for other BCs): $T^{-1}(\theta, x) = x \frac{\theta-1}{(2\theta-1)x-\theta}$

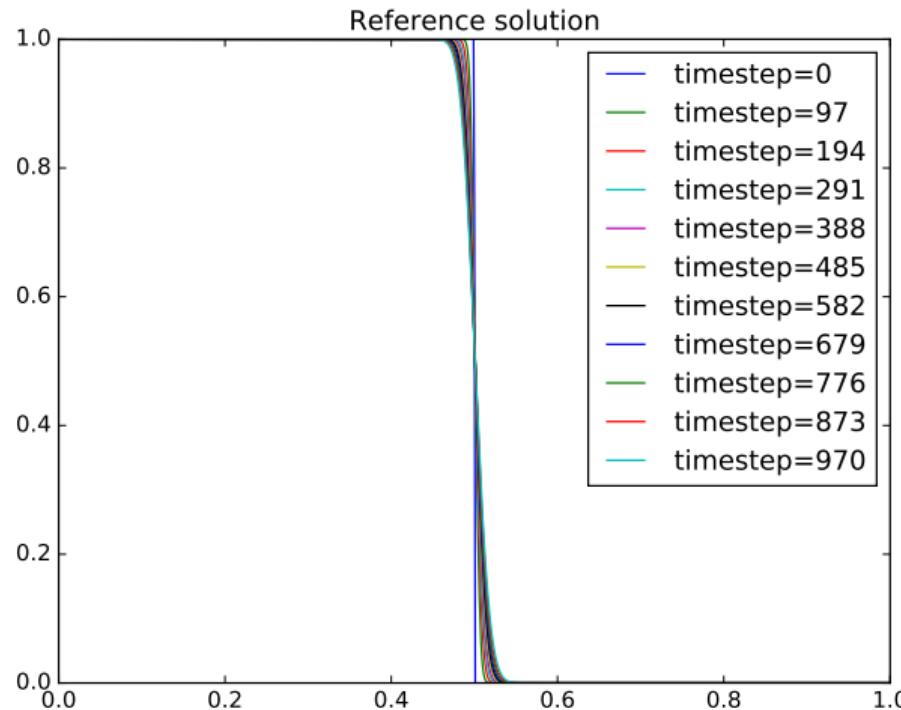


Figure: Calibrated solutions for traveling shock

POD of calibrated solutions

Dilatation (for other BCs): $T^{-1}(\theta, x) = x \frac{\theta-1}{(2\theta-1)x-\theta}$

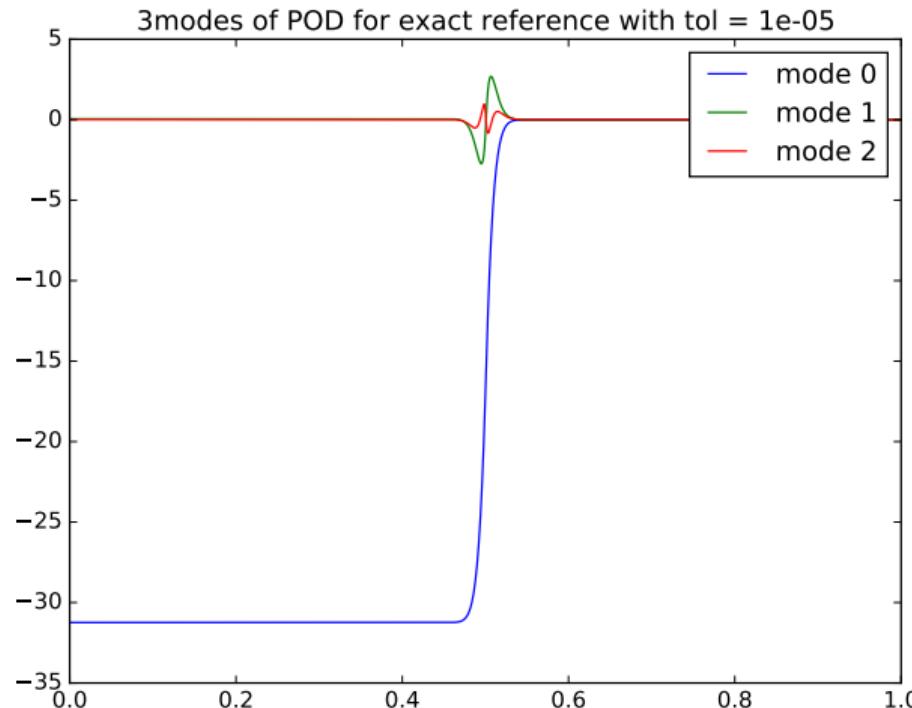


Figure: POD of calibrated solutions for travelling shock

Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

$$x := T(\theta(t, \mu), y), \quad v(y, t, \mu) := u(T(\theta(t, \mu), y), t, \mu) = u(x, t, \mu)$$

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

$$x := T(\theta(t, \mu), y), \quad v(y, t, \mu) := u(T(\theta(t, \mu), y), t, \mu) = u(x, t, \mu)$$

$$\frac{d}{dt} v(y, t, \mu) = \frac{d}{dt} u(T(\theta(t, \mu), y), t, \mu)$$

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

$$x := T(\theta(t, \mu), y), \quad v(y, t, \mu) := u(T(\theta(t, \mu), y), t, \mu) = u(x, t, \mu)$$

$$\begin{aligned}\frac{d}{dt} v(y, t, \mu) &= \frac{d}{dt} u(T(\theta(t, \mu), y), t, \mu) \\ &= \partial_t u(x, t, \mu) + \partial_x u(x, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt}\end{aligned}$$

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

$$x := T(\theta(t, \mu), y), \quad v(y, t, \mu) := u(T(\theta(t, \mu), y), t, \mu) = u(x, t, \mu)$$

$$\begin{aligned}\frac{d}{dt} v(y, t, \mu) &= \frac{d}{dt} u(T(\theta(t, \mu), y), t, \mu) \\&= \partial_t u(x, t, \mu) + \partial_x u(x, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt} \\&= -\frac{d}{dx} F(u(x, t, \mu), \mu) + \frac{d}{dx} u(x, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt}\end{aligned}$$

ALE formulation of the hyperbolic equations

$$\frac{d}{dt} u(x, t, \mu) + \frac{d}{dx} F(u(x, t, \mu), \mu) = 0$$

$$x := T(\theta(t, \mu), y), \quad v(y, t, \mu) := u(T(\theta(t, \mu), y), t, \mu) = u(x, t, \mu)$$

$$\begin{aligned}\frac{d}{dt} v(y, t, \mu) &= \frac{d}{dt} u(T(\theta(t, \mu), y), t, \mu) \\&= \partial_t u(x, t, \mu) + \partial_x u(x, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt} \\&= -\frac{d}{dx} F(u(x, t, \mu), \mu) + \frac{d}{dx} u(x, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt} \\&= -\frac{dy}{dx} \frac{d}{dy} F(v(y, t, \mu), \mu) + \frac{dy}{dx} \frac{d}{dy} v(y, t, \mu) \frac{dT(\theta(t, \mu), y)}{dt}\end{aligned}$$

Arbitrary Lagrangian–Eulerian formulation

ALE formulation

$$\frac{\partial}{\partial t} v(y, \mu, t) + \frac{dy}{dx} \frac{d}{dy} F(v, \mu) - \frac{dy}{dx} \frac{dv}{dy} \frac{\partial T}{\partial t} = 0$$

With ALE formulation we can apply the EIM procedure with points on the reference domain \mathcal{R} .

What does it imply?

- We must know $T(\theta(t, \mu), y)$
 - Offline phase: detect some interesting points (maxima, steepest gradient) (second part)
 - Offline phase: optimize the transformation in some sense (T. Taddei, Ohlberger et al.) (second part)
 - Online phase: predict the value of the transformation (regression/ machine learning techniques (RNN) / projections)
- Compute the Jacobian of the transformation $\frac{dy}{dx}$ and the new flux $\frac{dv}{dy} \Rightarrow$ increasing computational costs also in online phase

Arbitrary Lagrangian–Eulerian formulation

ALE formulation

$$\frac{\partial}{\partial t} v(y, \mu, t) + \frac{dy}{dx} \frac{d}{dy} F(v, \mu) - \frac{dy}{dx} \frac{dv}{dy} \frac{\partial T}{\partial t} = 0$$

With ALE formulation we can apply the EIM procedure with points on the reference domain \mathcal{R} .

What does it imply?

- We must know $T(\theta(t, \mu), y)$
 - Offline phase: detect some interesting points (maxima, steepest gradient) (second part)
 - Offline phase: optimize the transformation in some sense (T. Taddei, Ohlberger et al.) (second part)
 - Online phase: predict the value of the transformation (regression/ machine learning techniques (RNN) / projections)
- Compute the Jacobian of the transformation $\frac{dy}{dx}$ and the new flux $\frac{dv}{dy} \Rightarrow$ increasing computational costs also in online phase

Calibration map

- $\theta(\mu)$ tells us where a feature is (maximum point, steepest gradient)
- How to choose it? (second part)
- How to learn the map? (Offline we want to know the map in advance)
- Offline: optimization process on a training sample
- Generation of a regression map

Piecewise linear regression for every timestep t^n

- If parameter domain is a grid \Rightarrow Easy, fast
- Non-structured parameter domain \Rightarrow Different algorithms, may be costly
- Precise if $|\mathcal{P}_h| \sim s^P$ with s big enough
- May not catch the nonlinear behavior and produce unreasonable results

Calibration map

- $\theta(\mu)$ tells us where a feature is (maximum point, steepest gradient)
- How to choose it? (second part)
- How to learn the map? (Offline we want to know the map in advance)
- Offline: optimization process on a training sample
- Generation of a regression map

Polynomial regression

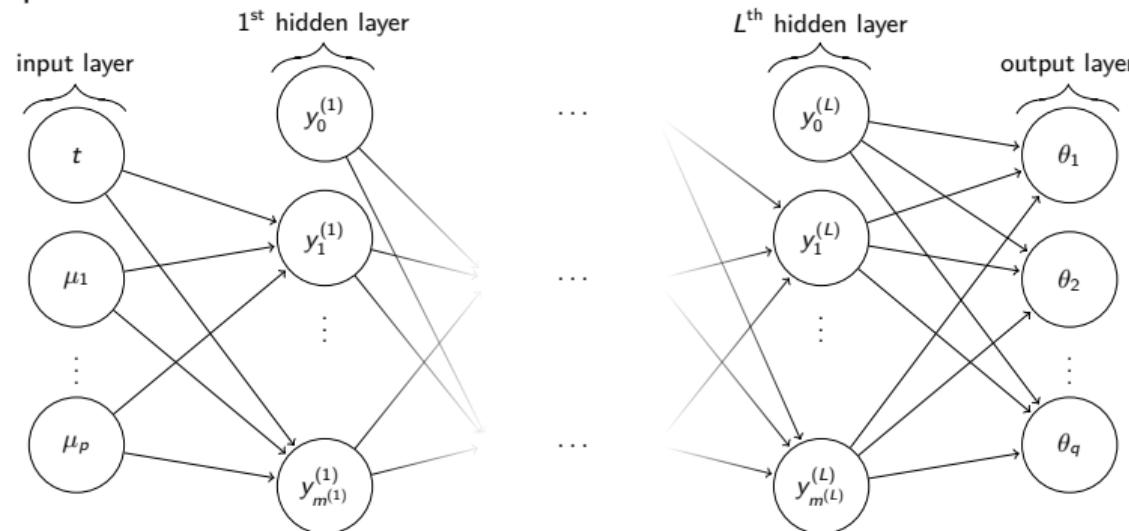
- Hyperparameter p
- Risk of overfitting
- Can easily catch the nonlinear (polynomial) behavior
- Number of coefficients grows exponentially with p

$$\theta(\mu, t) \approx \sum_{|\alpha| \leq p} \beta_\alpha t^{\gamma_0} \prod_{i=1}^p \mu_i^{\gamma_i} \quad (6)$$

Neural networks

- Why? Naturally nonlinear, we may not have a structured dictionary
- Which one? Multi-layer-perceptron, recursive neural network (RNN)

Multilayer perceptron



Neural networks

- Why? Naturally nonlinear, we may not have a structured dictionary
- Which one? Multi-layer-perceptron, recursive neural network (RNN)

Multilayer perceptron

- Many hyperparameters: N layers ($[4, 10]$), M_n nodes ($[6, 20]$)
- Not so precise, error ~ 20 cells

Review of the algorithm

INITIALIZATION:

- Compute or optimize $\theta(\mu_k, t^n)$ for some $\mu_k \in \mathcal{P}$ and $n \leq N_t$
- Build the regression map $\hat{\theta} : \mathcal{P} \times \mathbb{R}^+ \rightarrow \mathbb{R}^q$
- EIM on $\tilde{\mathcal{L}}(U^n, \mu_0, t^n, \hat{\theta}(\mu_0, t^n))$ for $n \leq N_t$
- $RB = POD(\{U^n(\mu_0)\}_{n=0}^{N_t})$

ITERATION:

- Greedy algorithm spanning over the parameter space \mathcal{P}_h , with an error indicator $\varepsilon(\mathbf{U}(\mu, t^n, \hat{\theta}(\mu, t^n)))$ where $\mathbf{U} \in \mathbb{R}^N$
- Choose worst parameter as $\mu^* = \arg \max_{\mu \in \mathcal{P}_h} \varepsilon(\mathbf{U}(\mu))$
- Apply POD on time evolution of selected solution $POD_{add} = POD(\{U^n(\mu^*)\}_{n=1}^{N_t})$
- Update the RB with $RB = POD(RB \cup POD_{add})$
- Update EIM basis function with $EIM_{space} = EIM_{space} \cup EIM(\{\tilde{\mathcal{L}}(U^n, \mu^*, t^n, \hat{\theta}(\mu^*, t^n))\}_{n=0}^{N_t})$

Table of contents

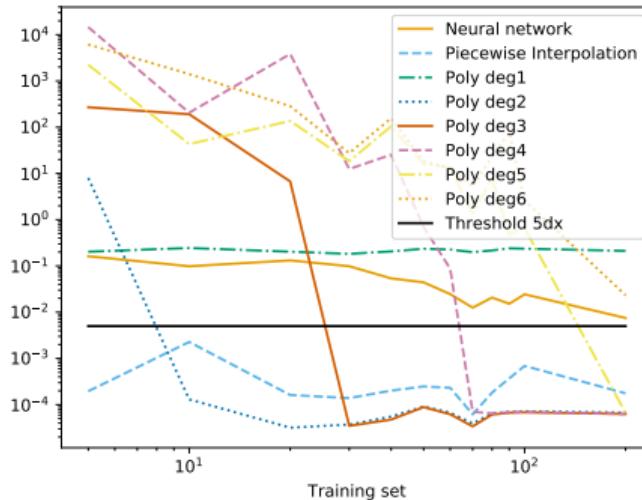
- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration

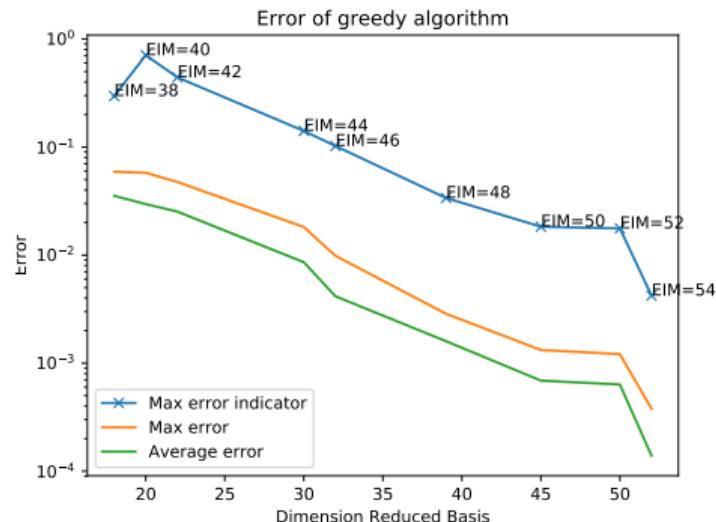
With calibration: Regressions



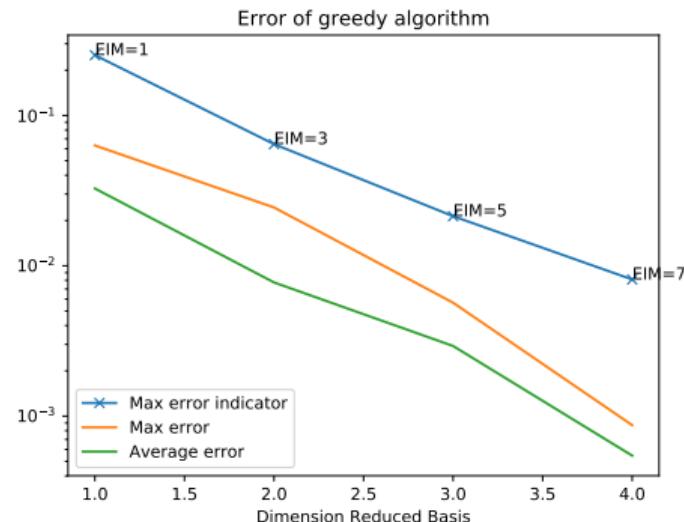
Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration



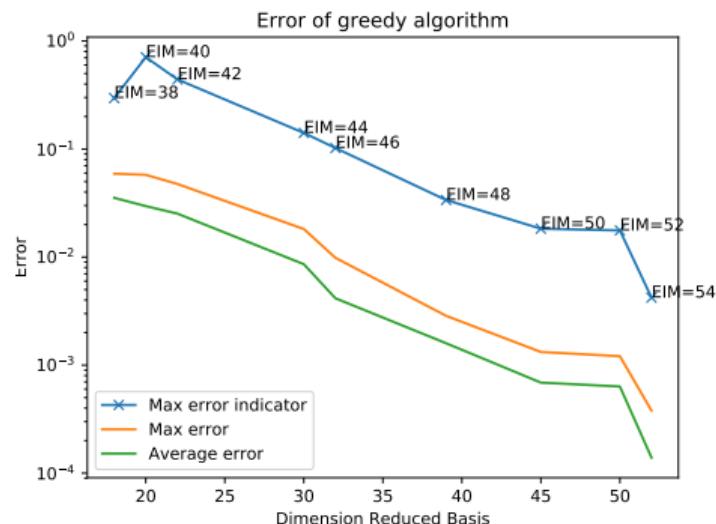
With calibration: Poly2



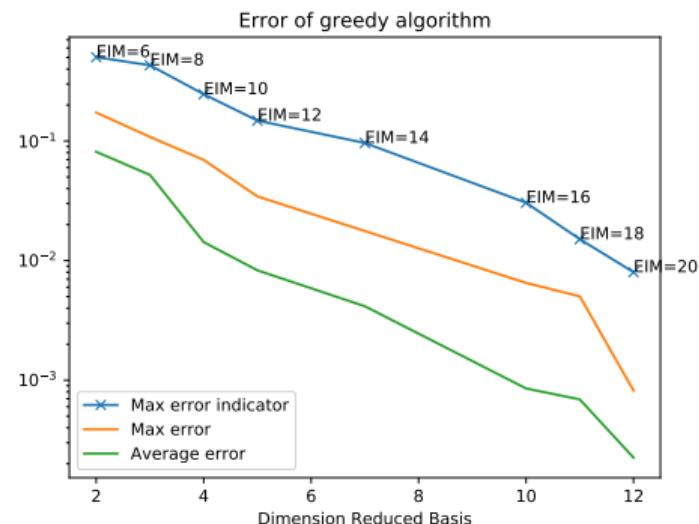
Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration



With calibration: ANN



Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration		With calibration: Poly2	
RB dim	52	RB dim	4
EIM dim	54	EIM dim	7
FOM time	191 s	FOM time	516 s
RB time	24 s	RB time	18 s
RB/FOM time	12%	RB/FOM time	3%

Advection: traveling wave

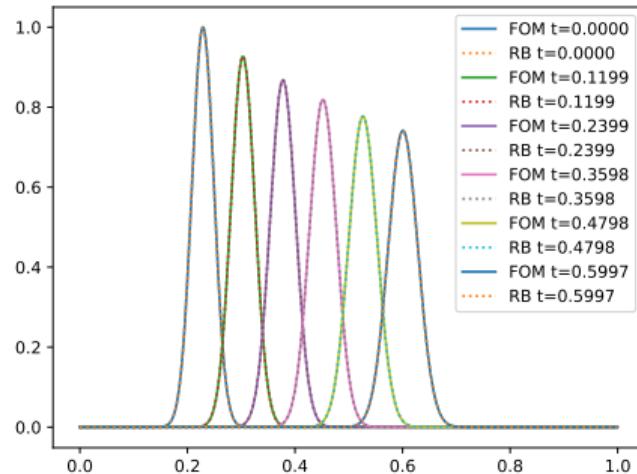
$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration		With calibration: ANN	
RB dim	52	RB dim	12
EIM dim	54	EIM dim	20
FOM time	191 s	FOM time	516 s
RB time	24 s	RB time	38 s
RB/FOM time	12%	RB/FOM time	7%

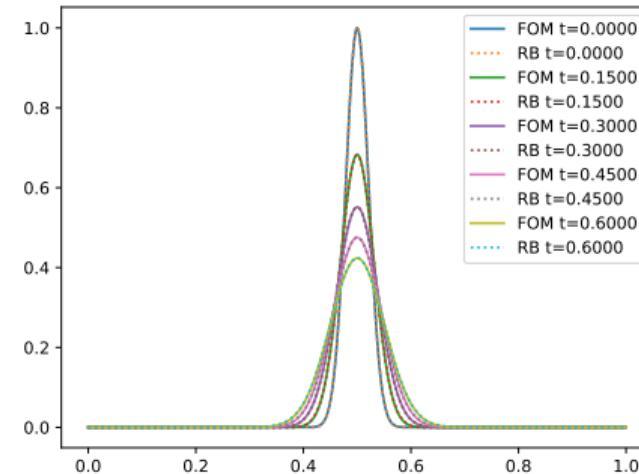
Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration



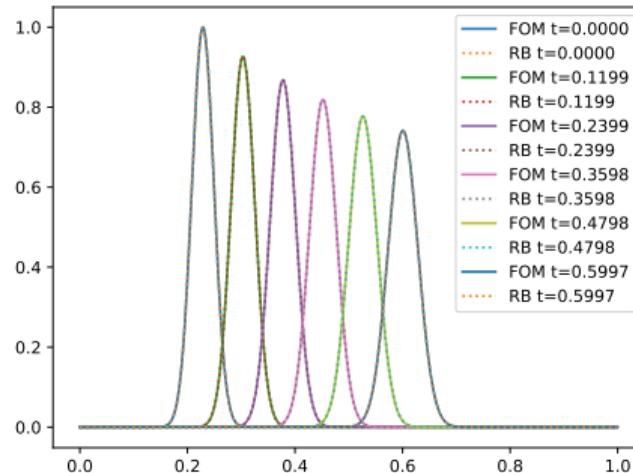
With calibration: Poly2



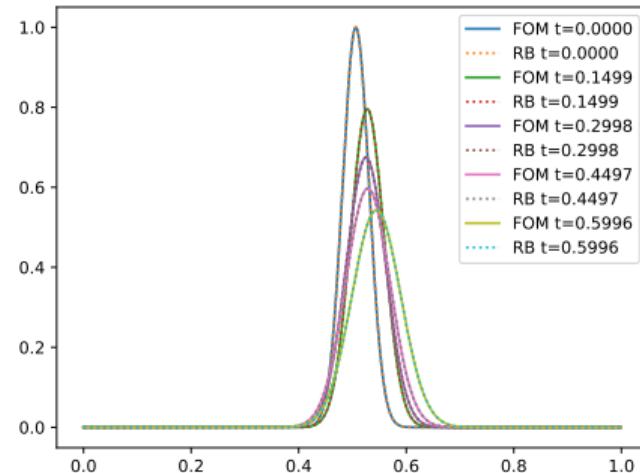
Advection: traveling wave

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 0.6, \text{ periodic BC} \\ u_0(x, \mu) = e^{-\mu_1(x - \mu_2)^2} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([500, 1500]), \mu_2 \sim \mathcal{U}([0.1, 0.3]) \end{cases}$$

Without calibration



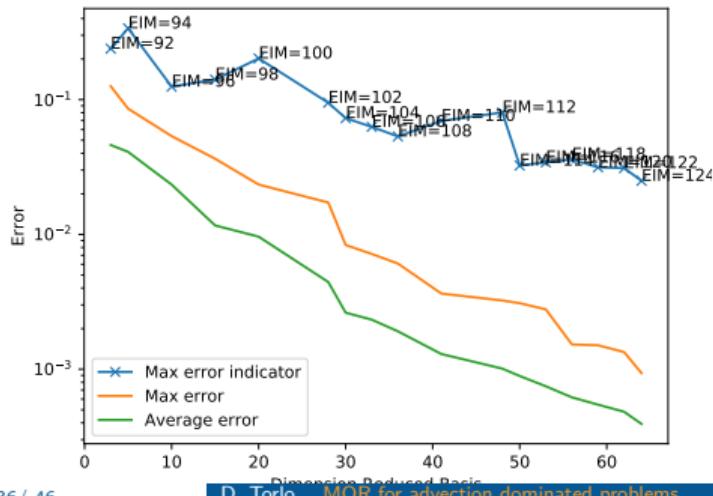
With calibration: ANN



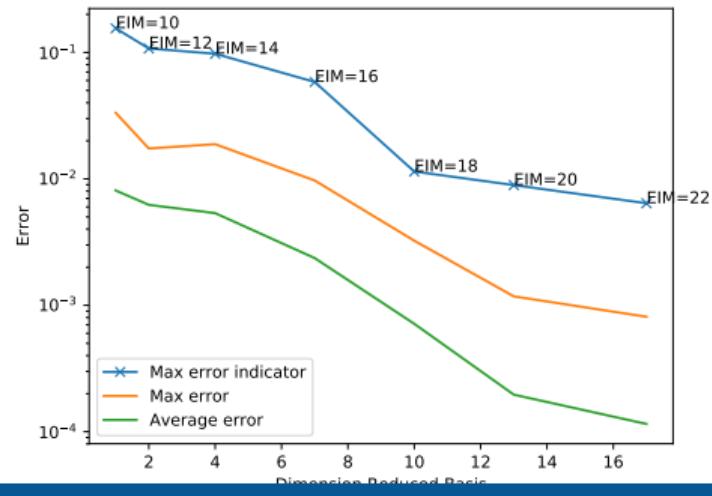
Advection: traveling shock

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 1.5, \text{ Dirichlet BC} \\ u_0(x, \mu) = \begin{cases} \mu_1 & \text{if } x < 0.35 + 0.05\mu_2 \\ 0 & \text{else} \end{cases} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1, \mu_2 \sim \mathcal{U}([-1, 1]) \end{cases}$$

Without calibration



With calibration: Poly2



Advection: traveling shock

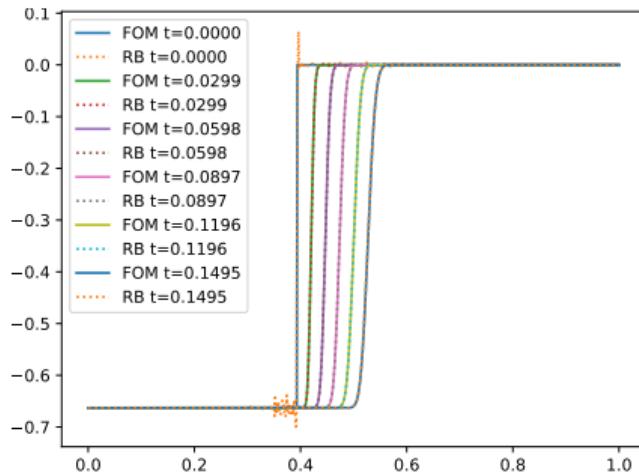
$$\begin{cases} u_t + \mu_0 u_x = 0, \quad D = [0, 1], \quad T_{max} = 1.5, \text{ Dirichlet BC} \\ u_0(x, \mu) = \begin{cases} \mu_1 & \text{if } x < 0.35 + 0.05\mu_2 \\ 0 & \text{else} \end{cases} \\ \mu_0 \sim \mathcal{U}([0, 2]), \quad \mu_1, \mu_2 \sim \mathcal{U}([-1, 1]) \end{cases}$$

Without calibration		With calibration: Poly2	
RB dim	64	RB dim	17
EIM dim	124	EIM dim	22
FOM time	49 s	FOM time	125 s
RB time	9 s	RB time	6 s
RB/FOM time	18%	RB/FOM time	5%

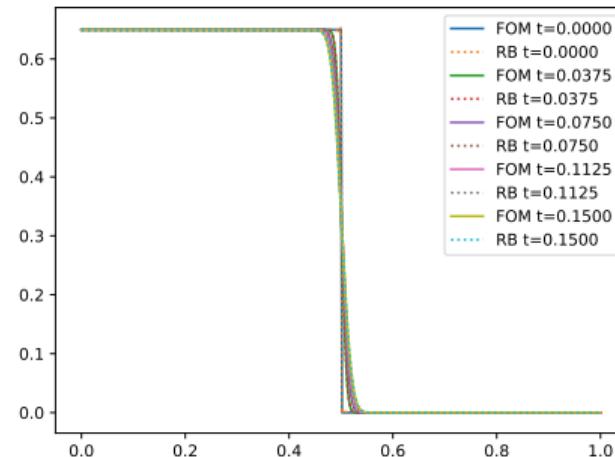
Advection: traveling shock

$$\begin{cases} u_t + \mu_0 u_x = 0, D = [0, 1], T_{max} = 1.5, \text{ Dirichlet BC} \\ u_0(x, \mu) = \begin{cases} \mu_1 & \text{if } x < 0.35 + 0.05\mu_2 \\ 0 & \text{else} \end{cases} \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1, \mu_2 \sim \mathcal{U}([-1, 1]) \end{cases}$$

Without calibration



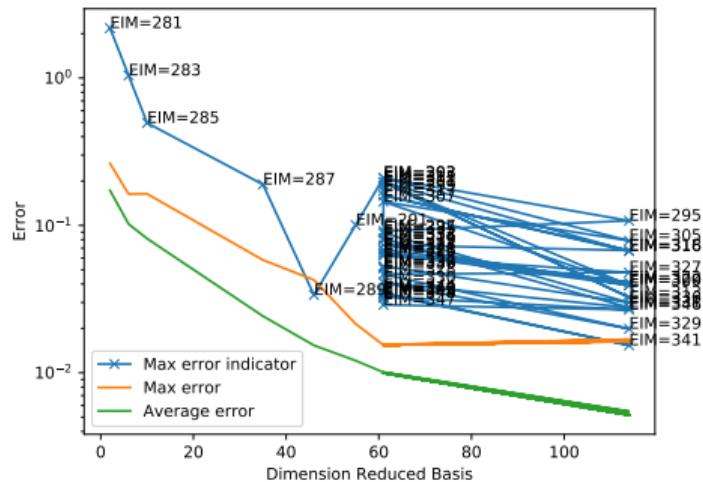
With calibration: Poly2



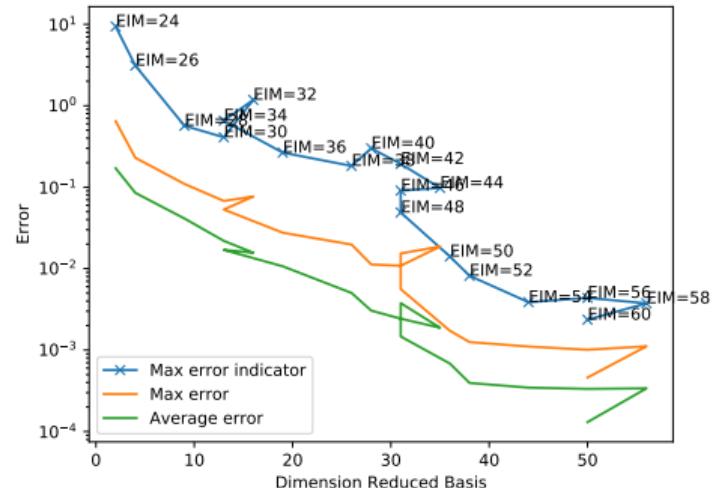
Burgers oscillation

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, D = [0, 1], T_{max} = 0.6, \text{ Dirichlet BC} \\ u_0(x, \mu) = \sin(2\pi(x + 0.1\mu_1))e^{-(60+20\mu_2)(x-0.5)^2}(1 + 0.5\mu_3x) \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([0, 1]), \mu_2, \mu_3 \sim \mathcal{U}([-1, 1]) \end{cases}$$

Without calibration



With calibration: Poly3



Burgers oscillation

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, \quad D = [0, 1], \quad T_{max} = 0.6, \text{ Dirichlet BC} \\ u_0(x, \mu) = \sin(2\pi(x + 0.1\mu_1))e^{-(60+20\mu_2)(x-0.5)^2}(1 + 0.5\mu_3x) \\ \mu_0 \sim \mathcal{U}([0, 2]), \quad \mu_1 \sim \mathcal{U}([0, 1]), \quad \mu_2, \mu_3 \sim \mathcal{U}([-1, 1]) \end{cases}$$

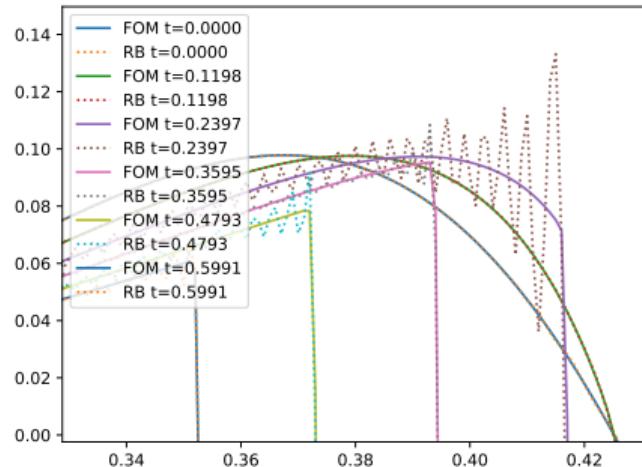
Without calibration ²		With calibration: Poly3	
RB dim	153	RB dim	50
EIM dim	335	EIM dim	60
FOM time	119 s	FOM time	314 s
RB time	50 s	RB time	35 s
RB/FOM time	42%	RB/FOM time	11%

²It does not reach the requested tolerance 10^{-3}

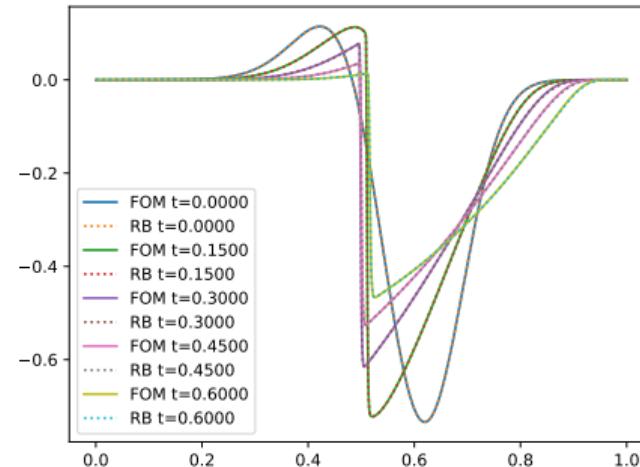
Burgers oscillation

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, D = [0, 1], T_{max} = 0.6, \text{ Dirichlet BC} \\ u_0(x, \mu) = \sin(2\pi(x + 0.1\mu_1))e^{-(60+20\mu_2)(x-0.5)^2}(1 + 0.5\mu_3x) \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([0, 1]), \mu_2, \mu_3 \sim \mathcal{U}([-1, 1]) \end{cases}$$

Without calibration



With calibration: Poly3

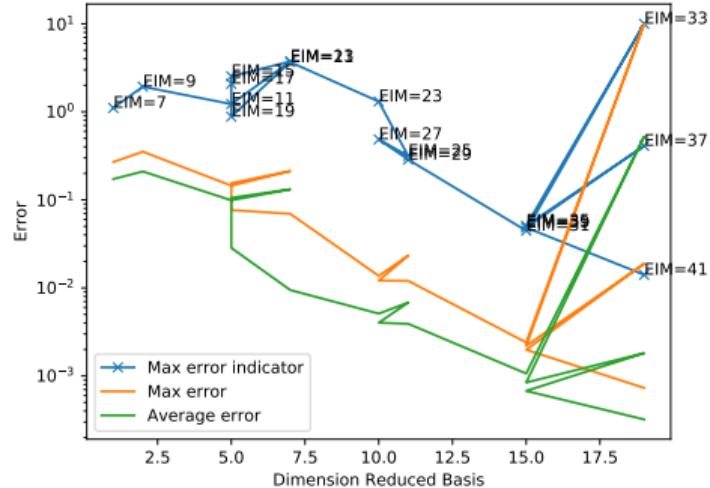


Burgers sine

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, D = [0, \pi], T_{max} = 0.15, \text{ periodic BC} \\ u_0(x, \mu) = |\sin(x + \mu_1)| + 0.1 \\ \mu_0 \sim \mathcal{U}([0, 2]), \mu_1 \sim \mathcal{U}([0, \pi]) \end{cases}$$

Without calibration

With calibration: Poly3



Burgers sine

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, \ D = [0, \pi], \ T_{max} = 0.15, \text{ periodic BC} \\ u_0(x, \mu) = |\sin(x + \mu_1)| + 0.1 \\ \mu_0 \sim \mathcal{U}([0, 2]), \ \mu_1 \sim \mathcal{U}([0, \pi]) \end{cases}$$

Without calibration		With calibration: Poly3	
RB dim	failed	RB dim	19
EIM dim	>600	EIM dim	41
FOM time	167 s	FOM time	444 s
RB time	∞	RB time	53 s
RB/FOM time	∞	RB/FOM time	11%

Burgers sine

$$\begin{cases} u_t + \mu_0(u^2/2)_x = 0, \quad D = [0, \pi], \quad T_{max} = 0.15, \text{ periodic BC} \\ u_0(x, \mu) = |\sin(x + \mu_1)| + 0.1 \\ \mu_0 \sim \mathcal{U}([0, 2]), \quad \mu_1 \sim \mathcal{U}([0, \pi]) \end{cases}$$

Without calibration

With calibration: Poly3

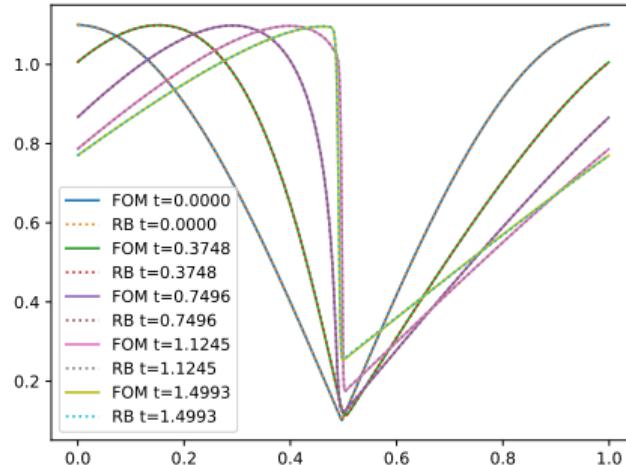


Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$

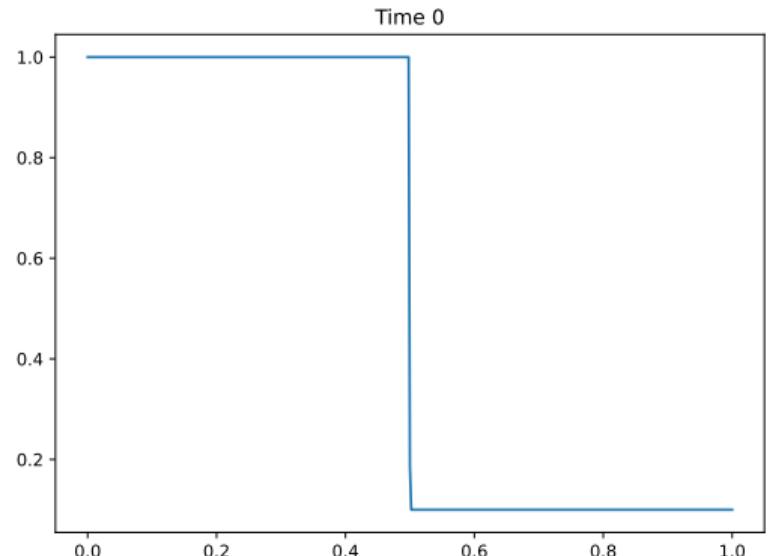
Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$



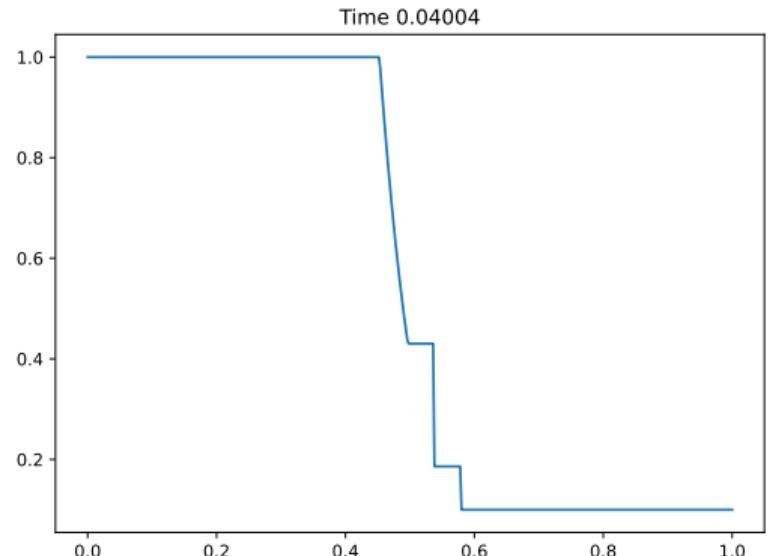
Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$



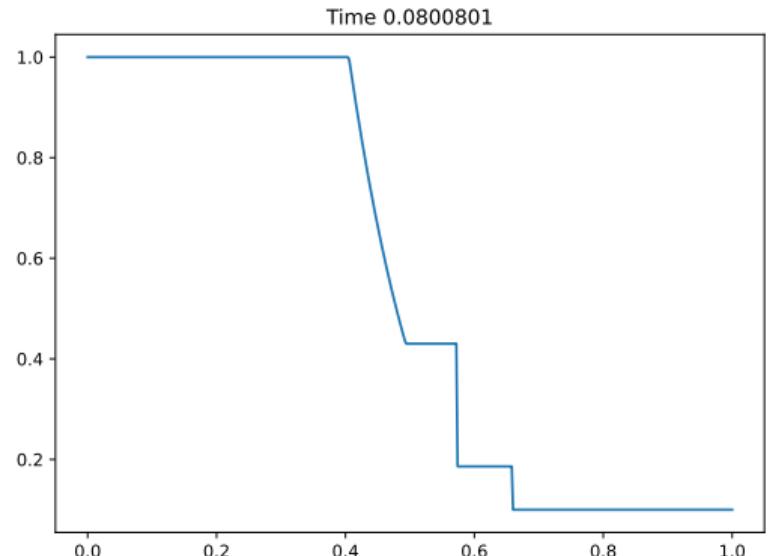
Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$



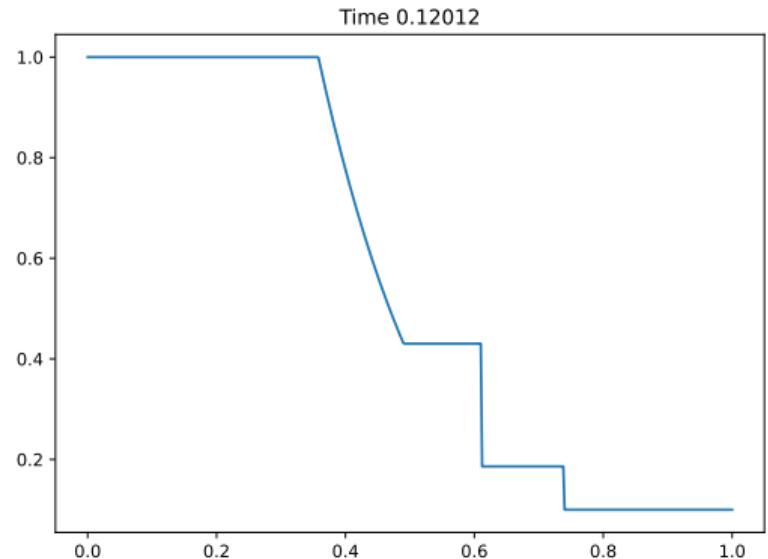
Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$



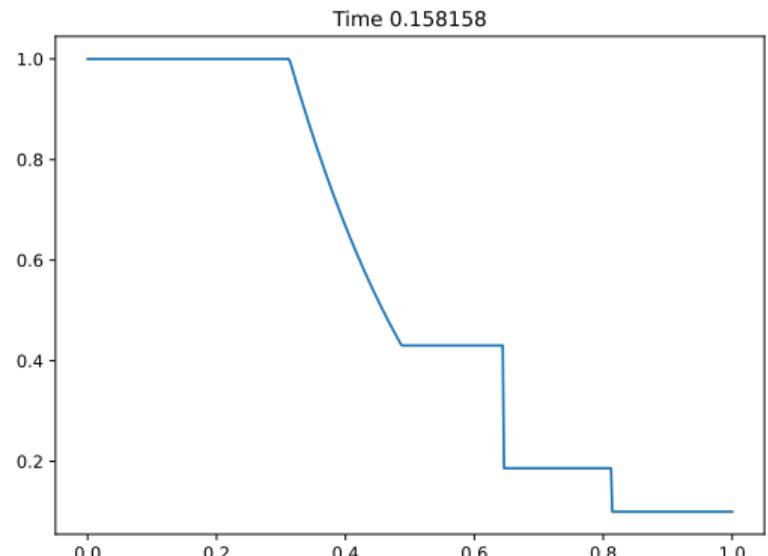
Sod shock tube test case

Euler Equations

$$\begin{cases} \partial_t \rho + \partial_x (\rho u) = 0 \\ \partial_t (\rho u) + \partial_x (\rho u^2 + p) = 0 \\ \partial_t (\rho E) + \partial_x (u(\rho E + p)) = 0 \\ + \text{EOS: } E = \frac{p}{\rho(\gamma-1)} + \frac{u^2}{2} \end{cases}$$

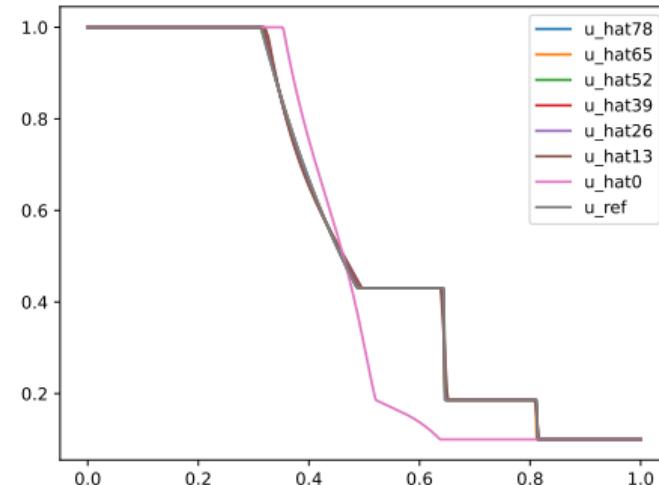
Riemann Problem: Sod Shock tube

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^T & \text{if } x < 0.5 \\ \begin{pmatrix} 0.1 & 0 & 0.125 \end{pmatrix}^T & \text{if } x > 0.5 \end{cases}$$



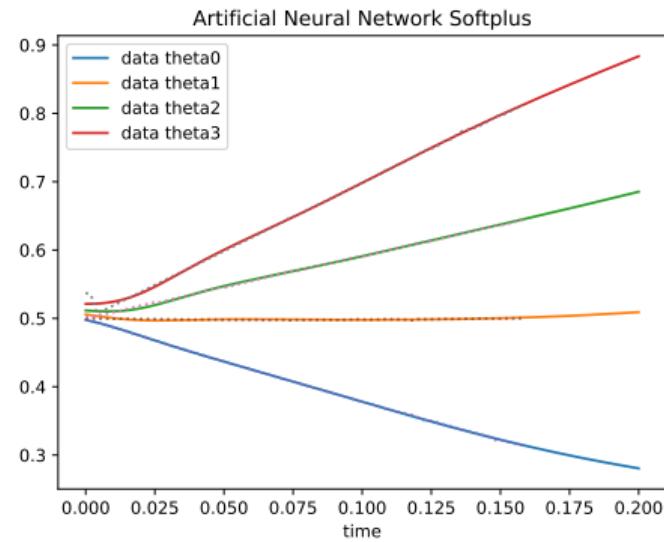
New transformations

- More features to align (shock, contact discontinuity, rarefaction wave)
- Transformation $y = T^{-1}(\theta(\mu, t), x)$ should still be monotone and smooth
- pchip Piecewise Cubic Hermite Interpolating Polynomial
- Preserve monotonicity
- Can map multiple points into reference points



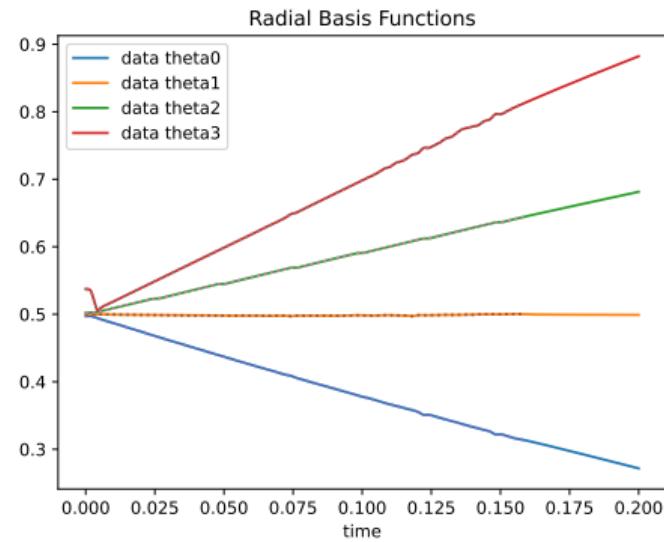
New calibration

- Choice of reference solution (final time solution)
- Optimization of the θ for training set
 - Constraint optimization (monotonicity $0 \leq \theta_i \leq \theta_{i+1} \leq 1$)
 - To be invertible: $\theta_{i-1} + \varepsilon \leq \theta_i \leq \theta_{i+1} - \varepsilon$
 - Initial Guess Fundamental (should be already close to the good solution)
 - Sequential Least Squares Programming (SLSQP) (scipy)
- Few optimized parameters and then **regression**
 - Multilayer perceptron constraining the outputs



New calibration

- Choice of reference solution (final time solution)
- Optimization of the θ for training set
 - Constraint optimization (monotonicity $0 \leq \theta_i \leq \theta_{i+1} \leq 1$)
 - To be invertible: $\theta_{i-1} + \varepsilon \leq \theta_i \leq \theta_{i+1} - \varepsilon$
 - Initial Guess Fundamental (should be already close to the good solution)
 - Sequential Least Squares Programming (SLSQP) (scipy)
- Few optimized parameters and then regression
 - Multilayer perceptron constraining the outputs
 - Radial Basis Functions



Results: POD

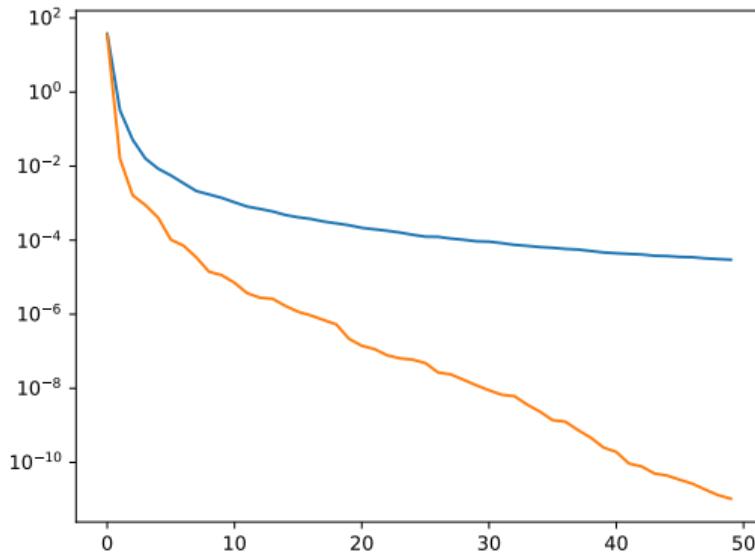
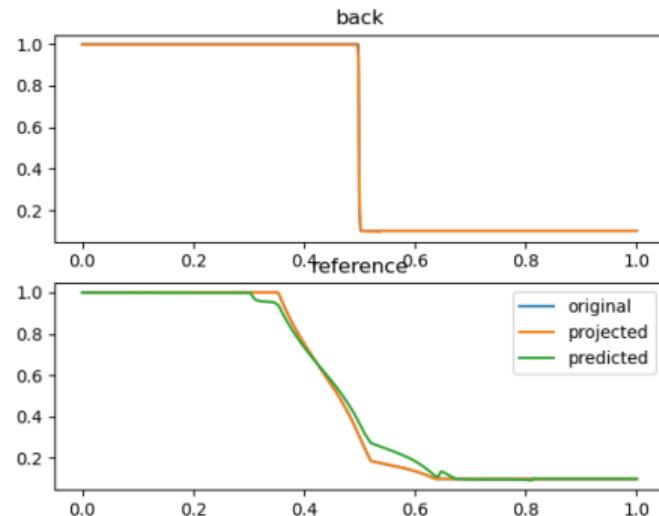


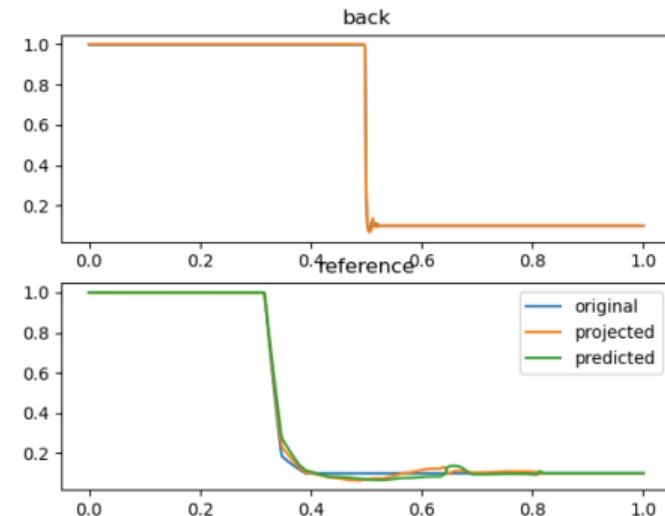
Figure: Singular values decay for original problem (blue) and transformed problem (orange)

Results: online with POD-NN

Radial Basis Functions Regression for θ

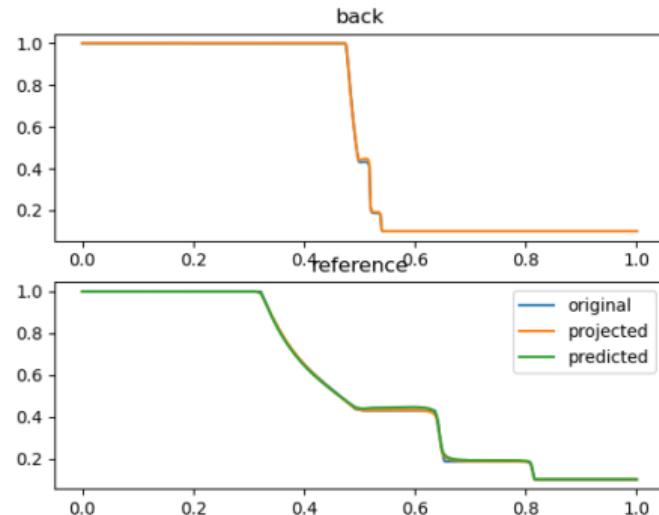


Multilayer Perceptron Regression for θ

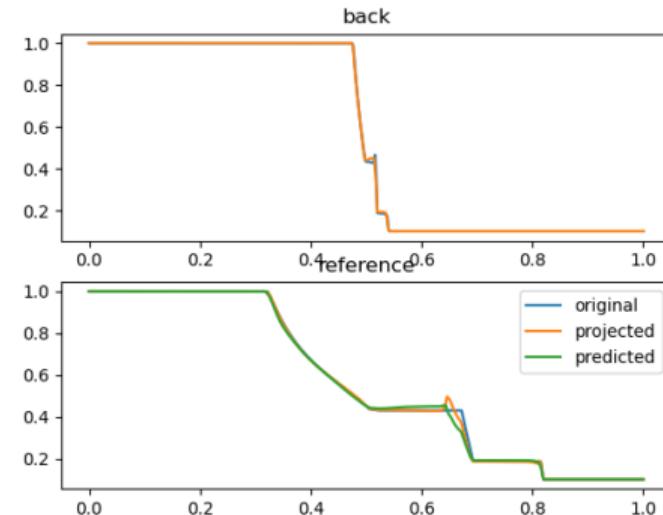


Results: online with POD-NN

Radial Basis Functions Regression for θ

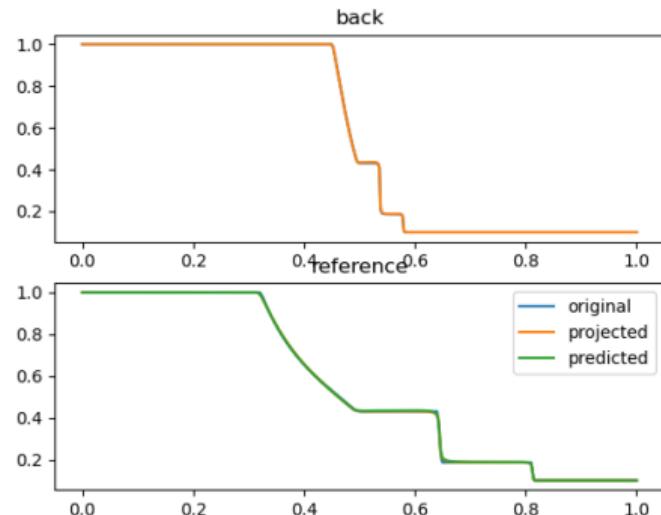


Multilayer Perceptron Regression for θ

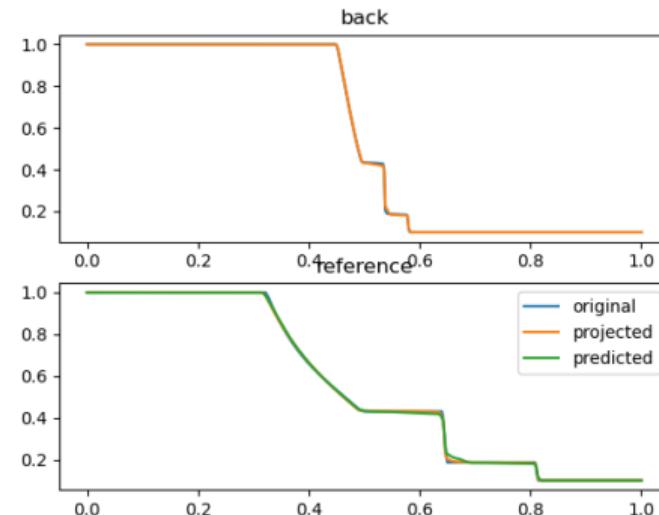


Results: online with POD-NN

Radial Basis Functions Regression for θ

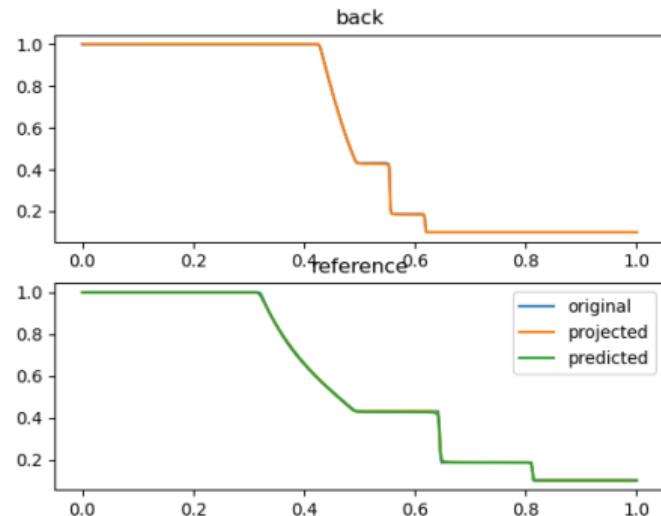


Multilayer Perceptron Regression for θ

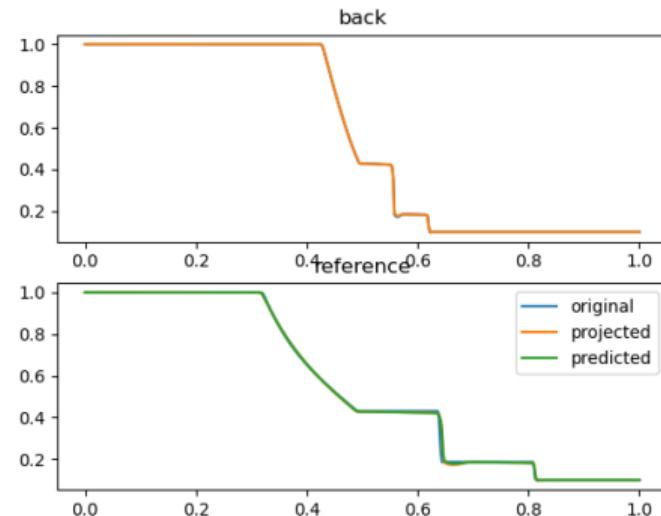


Results: online with POD-NN

Radial Basis Functions Regression for θ

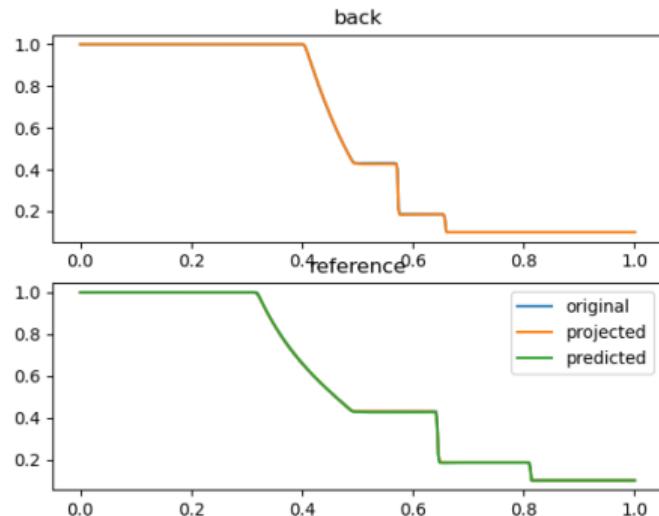


Multilayer Perceptron Regression for θ

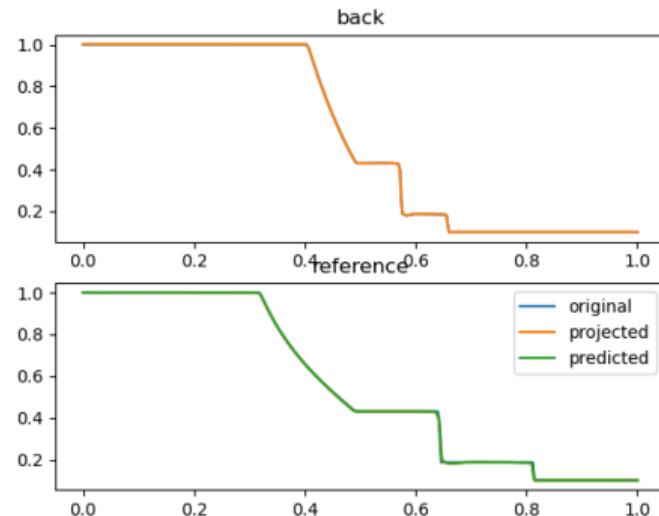


Results: online with POD-NN

Radial Basis Functions Regression for θ

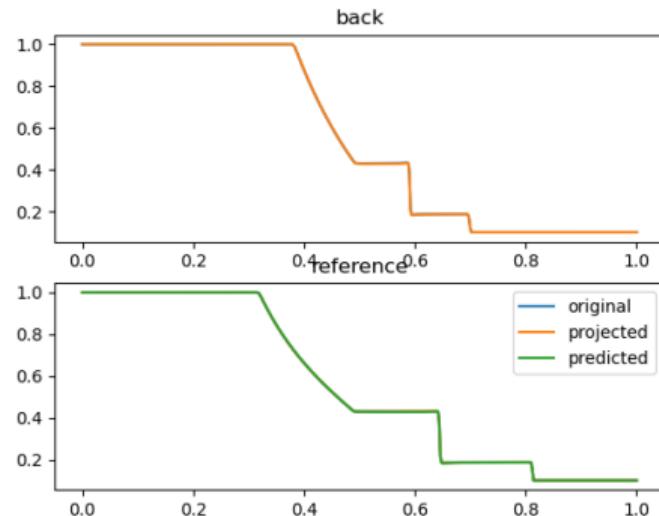


Multilayer Perceptron Regression for θ

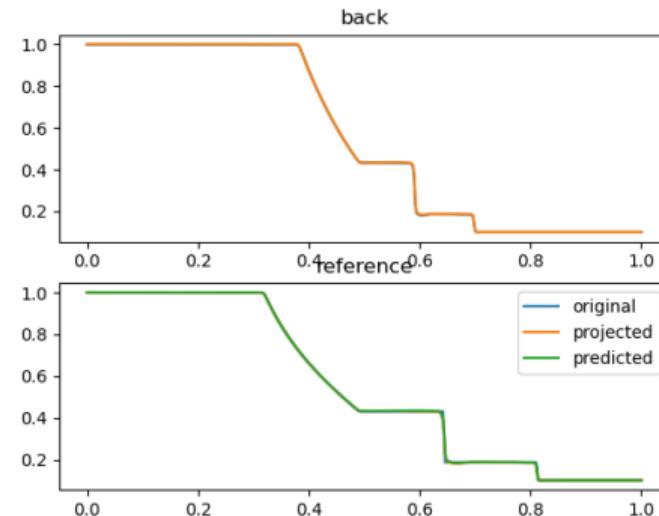


Results: online with POD-NN

Radial Basis Functions Regression for θ

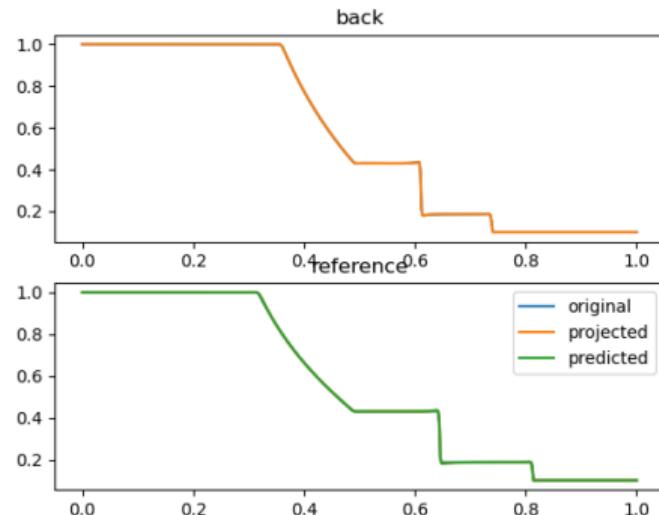


Multilayer Perceptron Regression for θ

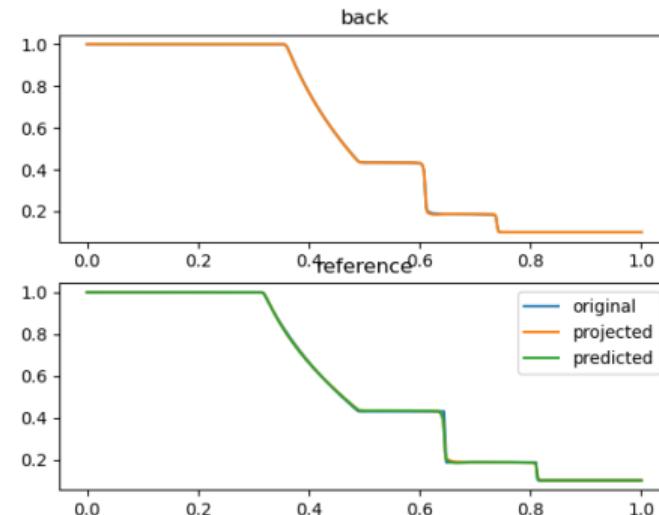


Results: online with POD-NN

Radial Basis Functions Regression for θ

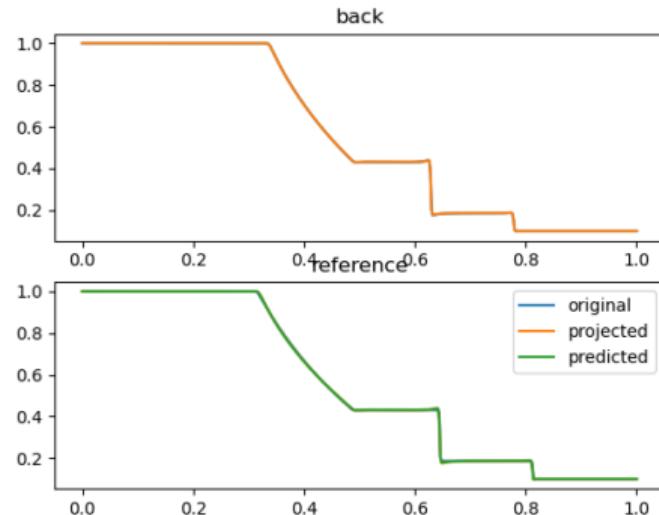


Multilayer Perceptron Regression for θ

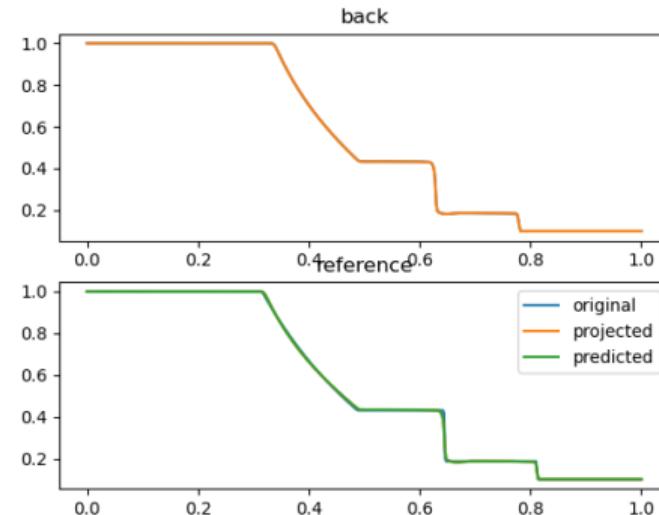


Results: online with POD-NN

Radial Basis Functions Regression for θ

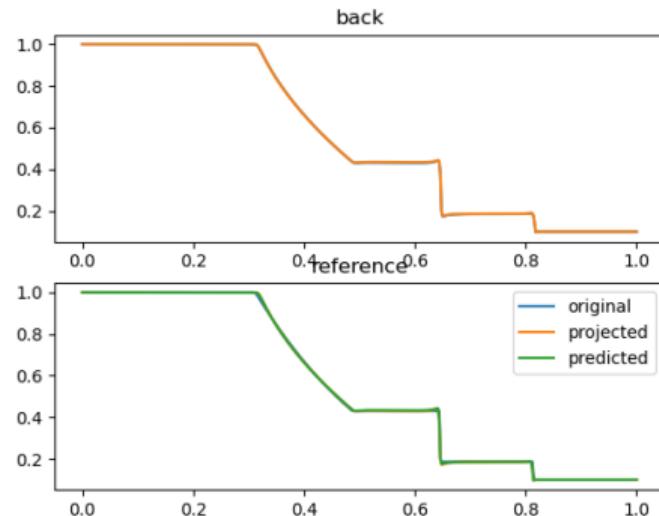


Multilayer Perceptron Regression for θ

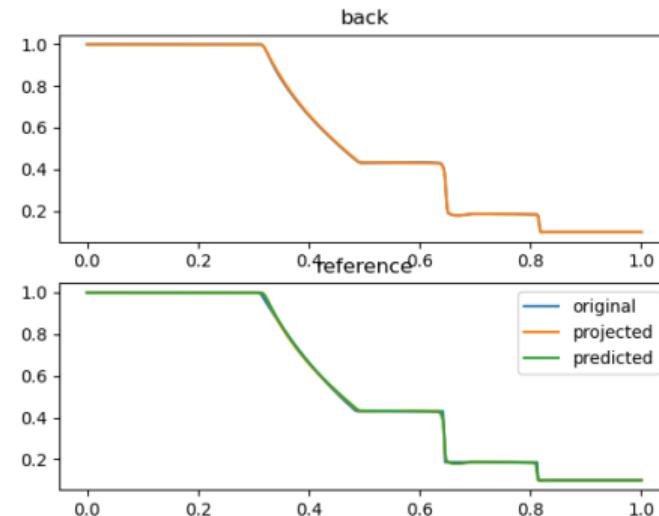


Results: online with POD-NN

Radial Basis Functions Regression for θ

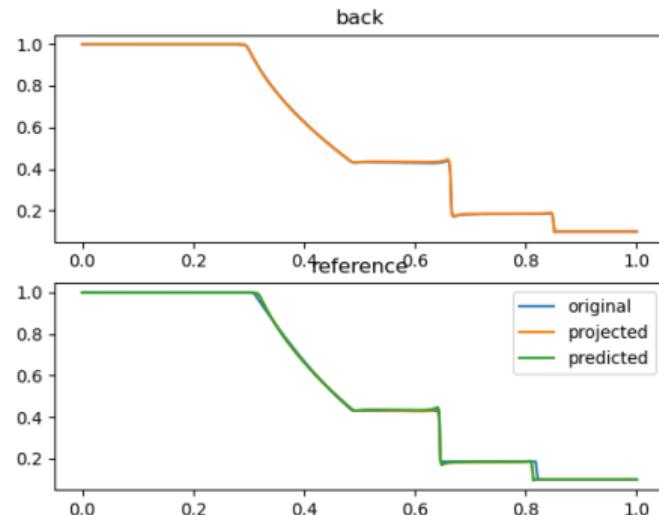


Multilayer Perceptron Regression for θ

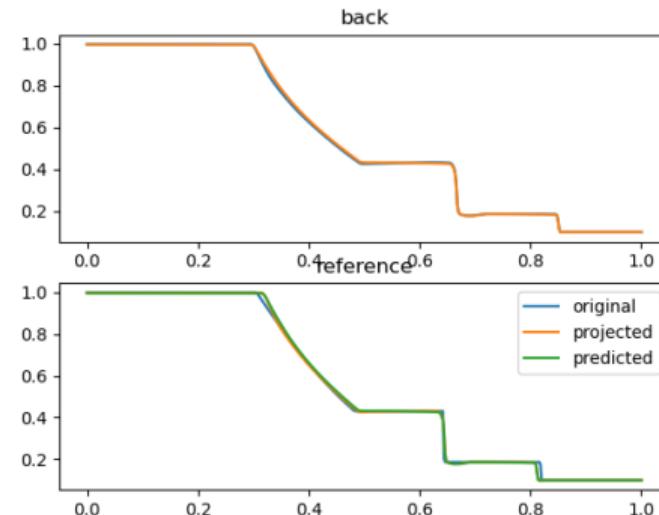


Results: online with POD-NN

Radial Basis Functions Regression for θ

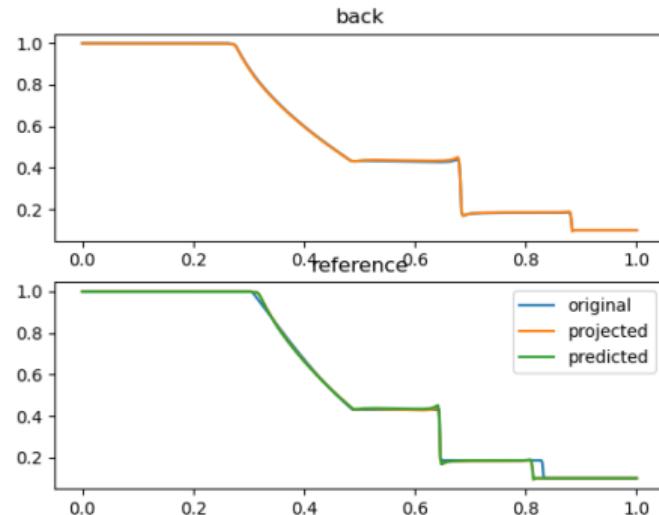


Multilayer Perceptron Regression for θ



Results: online with POD-NN

Radial Basis Functions Regression for θ



Multilayer Perceptron Regression for θ

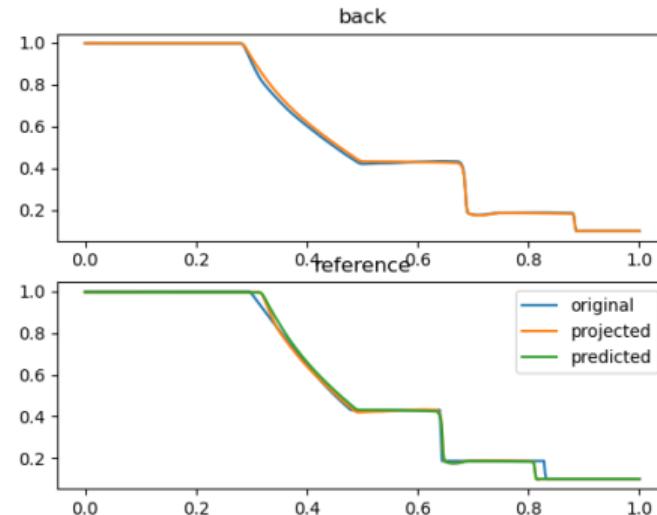


Table of contents

- ① MOR for hyperbolic problem
- ② Advection dominated problems in MOR
- ③ Solutions
- ④ ALE formulation
- ⑤ Results
- ⑥ Multiple discontinuities and optimal calibration (second part)
- ⑦ Possible extensions and limitations

Limitations

- Beginning of the Sod simulation still tricky
- More D transformation can be challenging to be described with few parameters
- Chaotic simulations
- Extrapolatory regime
- Delicate optimization

Extensions and limitations

Limitations

- Beginning of the Sod simulation still tricky
- More D transformation can be challenging to be described with few parameters
- Chaotic simulations
- Extrapolatory regime
- Delicate optimization

Extensions

- Robust optimization process in presence of multiple parameters
- Local ROM for different regimes
- More dimensions

Extensions and limitations

Limitations

- Beginning of the Sod simulation still tricky
- More D transformation can be challenging to be described with few parameters
- Chaotic simulations
- Extrapolatory regime
- Delicate optimization

Extensions

- Robust optimization process in presence of multiple parameters
- Local ROM for different regimes
- More dimensions

Thank you for the attention!

Residual Distribution

- High order
- FE based
- Compact stencil
- Explicit
- Can recast some other FV, FE, FD, DG schemes ²

²R. Abgrall. Computational Methods in Applied Mathematics, 2018

Residual Distribution

- High order
- FE based
- Compact stencil
- Explicit
- Can recast some other FV, FE, FD, DG schemes ²

$$\partial_t U + \nabla \cdot F(U) = 0 \quad (7)$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h), U|_K \in \mathbb{P}^k, \forall K \in \Omega_h\}. \quad (8)$$

$$U_h = \sum_{\sigma \in D_N} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma |_K \quad (9)$$

²R. Abgrall. Computational Methods in Applied Mathematics, 2018

Residual Distribution - Spatial Discretization

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K \nabla \cdot F(U) dx$
2. Define a nodal residual $\phi_\sigma^K \forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi_\sigma^K, \quad \forall K \in \Omega_h. \quad (10)$$

3. The resulting scheme is

$$U_\sigma^{n+1} - U_\sigma^n + \Delta t \sum_{K|\sigma \in K} \phi_\sigma^K = 0, \quad \forall \sigma \in D_N. \quad (11)$$

Residual Distribution

- High order
- Easy to code
- FE based
- Compact stencil
- No need of Riemann solver
- No need of conservative variables
- Can recast some other FV, FE schemes

Residual Distribution

- High order
- Easy to code
- FE based
- Compact stencil
- No need of Riemann solver
- No need of conservative variables
- Can recast some other FV, FE schemes

$$\partial_t U + \nabla \cdot A(U) = S(U) \quad (12)$$

$$V_h = \{U \in L^2(\Omega_h, \mathbb{R}^D) \cap \mathcal{C}^0(\Omega_h), U|_K \in \mathbb{P}^k, \forall K \in \Omega_h\}. \quad (13)$$

$$U_h = \sum_{\sigma \in D_N} U_\sigma \varphi_\sigma = \sum_{K \in \Omega_h} \sum_{\sigma \in K} U_\sigma \varphi_\sigma|_K \quad (14)$$

Residual Distribution - Spatial Discretization

Focus on steady case.

1. Define $\forall K \in \Omega_h$ a fluctuation term (total residual) $\phi^K = \int_K \nabla \cdot A(U) - S(U) dx$
2. Define a nodal residual $\phi_\sigma^K \forall \sigma \in K$:

$$\phi^K = \sum_{\sigma \in K} \phi_\sigma^K, \quad \forall K \in \Omega_h. \quad (15)$$

Often done assigning $\phi_\sigma^K = \beta_\sigma^K \phi^K$, where must hold that

$$\sum_{\sigma \in K} \beta_\sigma^K = \text{Id}. \quad (16)$$

3. The resulting scheme is

$$\sum_{K|\sigma \in K} \phi_\sigma^K = 0, \quad \forall \sigma \in D_N. \quad (17)$$

This will be called residual distribution scheme.

Residual distribution - Choice of the scheme

How to split total residuals into nodal residuals \Rightarrow choice of the scheme.

$$\begin{aligned}\phi_{\sigma}^{K,LxF}(U_h) &= \int_K \varphi_{\sigma} (\nabla \cdot A(U_h) - S(U_h)) dx + \alpha_K (U_{\sigma} - \bar{U}_h^K), \\ \bar{U}_h^K &= \int_K U_h, \quad \alpha_K = \max_{e \text{ edge } \in K} (\rho_S (\nabla A(U_h) \cdot \mathbf{n}_e)), \\ \beta_{\sigma}^K(U_h) &= \max \left(\frac{\Phi_{\sigma}^{K,LxF}}{\Phi^K}, 0 \right) \left(\sum_{j \in K} \max \left(\frac{\Phi_j^{K,LxF}}{\Phi^K}, 0 \right) \right)^{-1}, \\ \phi_{\sigma}^{*,K} &= (1 - \Theta) \beta_{\sigma}^K \phi_{\sigma}^K + \Theta \Phi_{\sigma}^{K,LxF}, \quad \Theta = \frac{|\Phi^K|}{\sum_{j \in K} |\Phi_j^{K,LxF}|}, \\ \phi_{\sigma}^K &= \beta_{\sigma}^K \phi_{\sigma}^{*,K} + \sum_{e \mid \text{edge of } K} \theta h_e^2 \int_e [\nabla U_h] \cdot [\nabla \varphi_{\sigma}] d\Gamma.\end{aligned}\tag{18}$$

Error estimator

Additional hypothesis:

- $Id + \Delta t \mathcal{L}$ is Lipschitz continuous with constant $C > 0$,
- There are N'_{EIM} extra functions and functionals that capture the evolution of the solutions.
(experimentally not so strict),
- Initial conditions are exactly represented in the reduced basis RB .

Total error estimator:

- EIM error, estimated by other N'_{EIM} basis functions f and functional τ iterating the EIM procedure after the stop, cost $\mathcal{O}(N'_{EIM})$,
- RB error given by the Lipschitz constant times residual of the small system,
- additionally one can add the projection error of the initial condition when not in RB .

Empirical interpolation method (EIM)

INPUT: $\mathcal{L}^n(U^n, \mu, t^n)$, for $\mu \in \mathcal{P}_h$, $n \leq N_t$

OUTPUT: $EIM = (\tau_k, f_k)_{k=1}^{N_{EIM}}$ where functions $f_k \in \mathbb{R}^{\mathcal{N}}$ and $\tau_k \in (\mathbb{R}^{\mathcal{N}})'$ (Examples of τ_k are point evaluations)

- Greedy iterative procedure
- At each step chooses the worst approximated function via an error estimator
$$\mathcal{L}^{worst} = \arg \max_{\mathcal{L}} \left| \left| \mathcal{L} - \sum_{k=1}^{N_{EIM}} \tau_k(\mathcal{L}) f_k \right| \right|$$
- Maximise the functional τ on the function \mathcal{L}^{worst} $\tau^{chosen} = \arg \max_{\tau} |\tau(\mathcal{L}^{worst})|$
- $EIM = EIM \cup (\tau^{chosen}, \mathcal{L}^{worst})$
- Stop when error is smaller than a tolerance

Proper orthogonal decomposition (POD)

INPUT: Collection of functions $\{f_j\}_{j=1}^N$

OUTPUT: Reduced basis spaces $RB = \arg \min_{U | \dim(U) = N_{POD}} \sum_{j=1}^N \|f_j - \mathcal{P}_U(f_j)\|_2$

- Based on SVD
- Prescribed tolerance to stop the algorithm
- Global optimizer of the problem

Greedy algorithm

INPUT: Collection of functions $\{f_j\}_{j=1}^N$

OUTPUT: Reduced basis space RB

- There is an error estimator (normally cheap) $\varepsilon_{RB}(f) \sim \|f - \mathcal{P}_{RB}(f)\|$
- Iteratively choose the worst represented function $f^{worst} = \arg \max_f \varepsilon_{RB}(f)$
- Add f^{worst} to the RB space
- Stop up to a certain tolerance