Training Robust Reduced-Order Models Using the Adjoint Method

Master's thesis

Francisco García Atienza

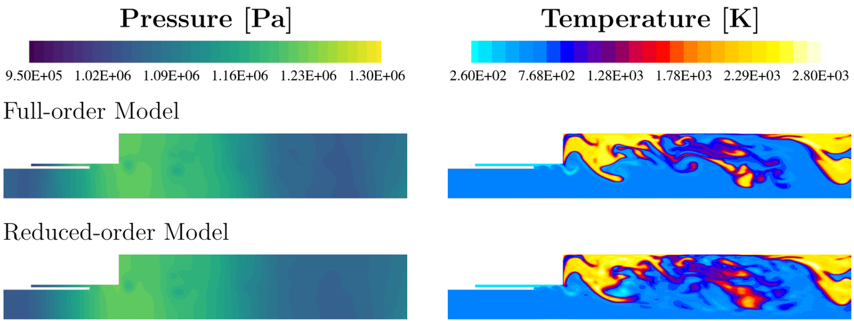
Supervisor: **Mengwu Guo**

Examiner: **Philipp Birken**

June 9, 2025

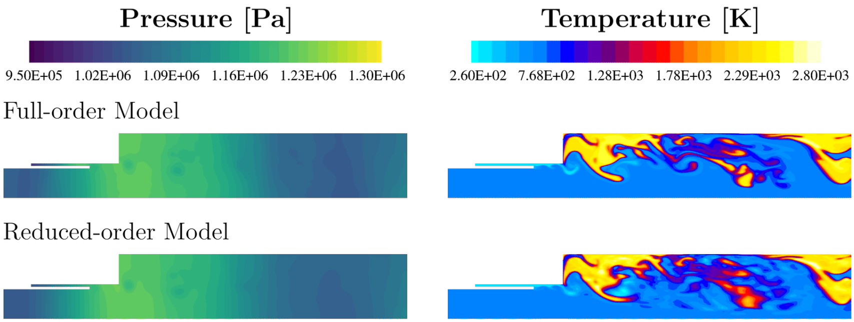
LUND UNIVERSITY

Model Order Reduction



McQuarrie, Huang, & Willcox (2021). *Data-driven reduced-order models via regularised operator inference for a single-injector combustion process*.

Model Order Reduction



McQuarrie, Huang, & Willcox (2021). *Data-driven reduced-order models via regularised operator inference for a single-injector combustion process*.

$\mathbf{q} \in \mathbb{R}^n, \quad n >> 1, k$

FOM $\xrightarrow{\text{train}}$ $\mathbf{Q} = \begin{bmatrix} | & | & & | \\ \mathbf{q}_{t_1} & \mathbf{q}_{t_2} & \cdots & \mathbf{q}_{t_k} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times k} \xrightarrow[\text{structure (SVD)}]{\text{identify}} \mathbf{V}_r \in \mathbb{R}^{n \times r} \xrightarrow{\text{project}} \hat{\mathbf{q}} \approx \mathbf{V}_r^\top \mathbf{q} \xrightarrow{\text{reduce}}$ ROM

$\dot{\mathbf{q}} = \mathbf{f}(t, \mathbf{q}(t), \mathbf{u}(t))$

$\hat{\mathbf{q}} \in \mathbb{R}^r, \quad n >> r$

$\dot{\hat{\mathbf{q}}} = \hat{\mathbf{f}}(t, \hat{\mathbf{q}}(t), \mathbf{u}(t))$

Operator Inference (OpInf)

OpInf is a **data-driven**, **non-intrusive** ROM.

Data-driven: Only high-fidelity snapshot data required ($\mathbf{Q}$).

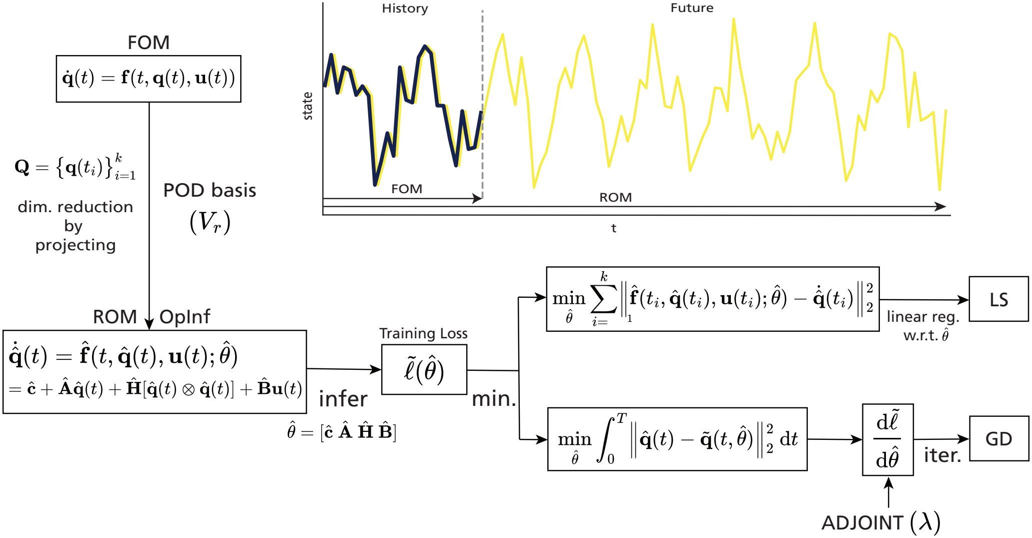
Non-intrusive: No access needed to the underlying full-order governing equations.

Example: 1D - Burgers' Eq.

$$\frac{\partial}{\partial t} q(x,t) = \nu \frac{\partial^2}{\partial x^2} q(x,t) - \frac{\partial}{\partial x} \frac{q(x,t)^2}{2} \xrightarrow[\text{}]{\substack{\text{Spatial Discr.} \\ + \\ \text{MOR}}} \dot{\hat{\mathbf{q}}}(t) = \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}\left(\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)\right).$$

We infer $\hat{\boldsymbol{\theta}} = [\hat{\mathbf{A}}, \hat{\mathbf{H}}] \in \mathbb{R}^{r \times (r+r^2)}$ by: $\min_{\hat{\boldsymbol{\theta}}} \sum_{i=1}^{k} \left\| \left[\hat{\mathbf{A}}\hat{\mathbf{q}}(t_i) + \hat{\mathbf{H}}\left(\hat{\mathbf{q}}(t_i) \otimes \hat{\mathbf{q}}(t_i)\right)\right] - \dot{\hat{\mathbf{q}}}(t_i) \right\|_2^2.$

Training Reduced Operators - Inference Process



Motivation & Objectives

Motivation

- Traditional OpInf: Limited by noise, incomplete data, nonlinear dynamics
- Real-world measurements $\Rightarrow$ Need robust ROMs with built-in noise attenuation
- Key challenge: Avoid unstable derivative approximations

Objectives

- **Reformulate OpInf** via:
  - Integral-based loss functional
  - Adjoint-based optimization
- Investigate **time integration** as low-pass filter

- Achieve:
  - Noise-robust training
  - Cost efficiency
  - Nonlinear compatibility
- Validate on **Burgers' eq.** (shocks) and **Fisher-KPP** (reaction-diffusion)

Problem Set-Up

We define the training loss

$$\ell : \mathcal{C}^1(\mathcal{T})^r \to \mathbb{R}\,, \quad \hat{\mathbf{q}}(\cdot) \mapsto \int_0^T \overbrace{\left\| \hat{\mathbf{q}}(t) - \hat{\mathbf{q}}_{\mathrm{true}}(t) \right\|_2^2}^{g(\hat{\mathbf{q}}(t),t)} \, dt,$$

and the reduced training loss

$$\tilde{\ell} : \mathbb{R}^d \to \mathbb{R}\,, \quad \hat{\boldsymbol{\theta}} \mapsto \int_0^T \left\| \tilde{\mathbf{q}}(t, \hat{\boldsymbol{\theta}}) - \hat{\mathbf{q}}_{\mathrm{true}}(t) \right\|_2^2 dt,$$

where $\tilde{\mathbf{q}}(t, \hat{\boldsymbol{\theta}}) := \tilde{\mathbf{q}}(0) + \int_0^t \hat{\mathbf{f}}(\tilde{\mathbf{q}}(\tau), \hat{\boldsymbol{\theta}}) \, d\tau, \quad t \in [0, T]$. Thus,

$$(*) \quad \min_{\hat{\boldsymbol{\theta}}} \; \tilde{\ell}(\hat{\boldsymbol{\theta}}) \qquad \equiv \qquad \begin{aligned} &\min_{\hat{\boldsymbol{\theta}}, \hat{\mathbf{q}}(\cdot)} \; \ell(\hat{\mathbf{q}}(\cdot)) \\ &\text{s.t.} \;\; \dot{\hat{\mathbf{q}}}(t) = \hat{\mathbf{f}}(\hat{\mathbf{q}}(t), \hat{\boldsymbol{\theta}}), \quad \hat{\mathbf{q}}(0) = \hat{\mathbf{q}}_0 \end{aligned}$$

Adjoint & Loss-Gradient Equations

Theorem. Assume $\hat{\mathbf{f}}, g \in \mathcal{C}^1\big((\hat{\mathbf{q}}, \hat{\boldsymbol{\theta}})\big)$, $\hat{\mathbf{q}}(0) = \hat{\mathbf{q}}_0$, (indep. of $\hat{\boldsymbol{\theta}}$). The adjoint variable $\boldsymbol{\lambda}(t) \in \mathcal{C}^1([0, T])^r$ associated to $(*)$ satisfies a back propagation ODE

$$\dot{\boldsymbol{\lambda}}(t) = -\left(\frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{q}}}\right)^{\top} \boldsymbol{\lambda}(t) - \left(\frac{\partial g}{\partial \hat{\mathbf{q}}}\right)^{\top}, \quad \boldsymbol{\lambda}(T) = \mathbf{0},$$

and the gradients w.r.t. $\hat{\boldsymbol{\theta}}$ of the reduced training ($\tilde{\ell}$) and the Lagrange cost ($\mathscr{L}$),

$$\mathscr{L}\big(\hat{\mathbf{q}}(\cdot), \hat{\boldsymbol{\theta}}, \boldsymbol{\lambda}(\cdot)\big) := \ell(\hat{\mathbf{q}}(\cdot)) - \int_0^T \boldsymbol{\lambda}(t)^{\top}\left(\dot{\hat{\mathbf{q}}}(t) - \hat{\mathbf{f}}(\hat{\mathbf{q}}(t), \hat{\boldsymbol{\theta}})\right) dt - \boldsymbol{\lambda}(0)^{\top}(\hat{\mathbf{q}}(0) - \hat{\mathbf{q}}_0),$$

have the form

$$\frac{\mathrm{d}\tilde{\ell}}{\mathrm{d}\hat{\boldsymbol{\theta}}} = \frac{\partial \mathscr{L}}{\partial \hat{\boldsymbol{\theta}}} = \int_0^T \boldsymbol{\lambda}(t)^{\top} \frac{\partial \hat{\mathrm{f}}}{\partial \hat{\boldsymbol{\theta}}}\, dt.$$

Proof (Lagrangian Formulation)

Differentiating w.r.t. $\hat{\boldsymbol{\theta}}$,

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \hat{\boldsymbol{\theta}}} &= \int_0^T \left[\frac{\partial g}{\partial \hat{\boldsymbol{\theta}}} + \frac{\partial g}{\partial \hat{\mathbf{q}}}\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}} - \boldsymbol{\lambda}(t)^\top \left(\frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}} - \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\boldsymbol{\theta}}} - \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{q}}}\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\right)\right] dt - \boldsymbol{\lambda}(0)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(0) \\ &= \int_0^T \left[\frac{\partial g}{\partial \hat{\boldsymbol{\theta}}} + \boldsymbol{\lambda}(t)^\top \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\boldsymbol{\theta}}} + \left(\frac{\partial g}{\partial \hat{\mathbf{q}}} + \boldsymbol{\lambda}(t)^\top \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{q}}} - \boldsymbol{\lambda}(t)^\top \frac{\mathrm{d}}{\mathrm{d}t}\right)\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\right] dt - \boldsymbol{\lambda}(0)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(0).\end{aligned}$$

Integrating by parts,

$$\int_0^T -\boldsymbol{\lambda}(t)^\top \frac{\mathrm{d}}{\mathrm{d}t}\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\, dt = \boldsymbol{\lambda}(0)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(0) - \boldsymbol{\lambda}(T)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(T) + \int_0^T \left(\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t}\right)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\, dt.$$

Plugging into the above expression,

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \hat{\boldsymbol{\theta}}} = \int_0^T &\left[\frac{\partial g}{\partial \hat{\boldsymbol{\theta}}} + \boldsymbol{\lambda}(t)^\top \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\boldsymbol{\theta}}} + \underbrace{\left(\frac{\partial g}{\partial \hat{\mathbf{q}}} + \boldsymbol{\lambda}(t)^\top \frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{q}}} + \dot{\boldsymbol{\lambda}}(t)^\top\right)}_{\text{make } = 0}\frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}\right] dt \\ &- \boldsymbol{\lambda}(0)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(0) + \boldsymbol{\lambda}(0)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(0) - \underbrace{\boldsymbol{\lambda}(T)^\top \frac{\mathrm{d}\hat{\mathbf{q}}}{\mathrm{d}\hat{\boldsymbol{\theta}}}(T)}_{\text{make } = 0}. \qquad \square\end{aligned}$$

Equations of the Three Gradients

For the ROM-OpInf RHS $\hat{\mathbf{f}} : \mathbb{R}^r \times \mathbb{R}^d \to \mathbb{R}^r$ and the loss integrand $g : \mathbb{R}^r \times [0, T] \to \mathbb{R}$,

$$\hat{\mathbf{f}}(\hat{\mathbf{q}}(t), \hat{\theta}) = \hat{\mathbf{c}} + \hat{\mathbf{A}}\hat{\mathbf{q}}(t) + \hat{\mathbf{H}}\left(\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t)\right) + \hat{\mathbf{B}}\mathbf{u}(t)$$

$$g(\hat{\mathbf{q}}(t), t) = \left\|\hat{\mathbf{q}}(t) - \hat{\mathbf{q}}_{\mathrm{true}}(t)\right\|_2^2,$$

$$\nabla_{\hat{\mathbf{q}}}\, \hat{\mathbf{f}}(\hat{\mathbf{q}}, \hat{\theta}) = \hat{\mathbf{A}}^\top + 2\left(\hat{\mathbf{H}}(\hat{\mathbf{q}}(t) \otimes \mathbf{I}_r)\right)^\top = \hat{\mathbf{A}}^\top + 2\mathbf{M}(\hat{\mathbf{q}}(t)) \ \in \mathbb{R}^{r \times r}$$

$$\nabla_{\hat{\mathbf{q}}}\, g(\hat{\mathbf{q}}, t) = 2\left(\hat{\mathbf{q}}(t) - \hat{\mathbf{q}}_{\mathrm{true}}(t)\right) \ \in \ \mathbb{R}^r$$

$$\nabla_{\hat{\theta}}\, \hat{\mathbf{f}}(\hat{\mathbf{q}}, \hat{\theta}) = \left[\mathbf{I}_r,\ \hat{\mathbf{q}}(t)^\top \otimes \mathbf{I}_r,\ (\hat{\mathbf{q}}(t) \otimes \hat{\mathbf{q}}(t))^\top \otimes \mathbf{I}_r,\ \mathbf{u}(t)^\top \otimes \mathbf{I}_r\right]^\top \ \in \ \mathbb{R}^{d \times r}.$$

Gradient Descent Optimization

Algorithm 1: Armijo Backtracking Line Search + Gradient Descent

Step 1. Initialization: Choose $\alpha \in (0,1)$, $\beta \in (0,1)$, and initial step size $\eta_0 > 0$. Set iteration counter $j = 0$;

Step 2. Compute descent direction: Evaluate gradient $\mathscr{G}^j = \nabla \tilde{\ell}(\hat{\boldsymbol{\theta}}^j)$;

Step 3. Backtracking loop:

$$\eta \leftarrow \eta_0 \quad \text{while} \quad \tilde{\ell}(\hat{\boldsymbol{\theta}}^j - \eta \mathscr{G}^j) > \tilde{\ell}(\hat{\boldsymbol{\theta}}^j) - \alpha \eta \|\mathscr{G}^j\|_2^2 \quad \text{do} \quad \eta \leftarrow \beta \eta$$

Step 4. Update: Set step size $\eta_j = \eta$ and update

$$\hat{\boldsymbol{\theta}}^{j+1} = \hat{\boldsymbol{\theta}}^j - \eta_j \mathscr{G}^j .$$

Step 5. Stopping criterion: If $\|\mathscr{G}^{j+1}\|_2 \leq \epsilon$ (or another criterion) then stop; otherwise set $j \leftarrow j+1$ and go to Step 2.

Adjoint Method Algorithm

Algorithm 2: Adjoint Method for Parameter Training

Step 1. Data collection and preprocessing: Gather snapshots $\mathbf{Q} = [\mathbf{q}_{t_1}, \ldots, \mathbf{q}_{t_k}]$, $\mathbf{U} = [\mathbf{u}_{t_1}, \ldots, \mathbf{u}_{t_k}]$. [Optional] lift/scale/center the data;

Step 2. Dimensionality reduction: Compute POD basis $\mathbf{V}_r$ (cumulative energy criterion or fixed ROM dimension r). Project: $\hat{\mathbf{Q}} \approx \mathbf{V}_r^\top \mathbf{Q}$;

Step 3. Initial parameter guess: Solve OpInf regression with $\dot{\mathbf{Q}}$ to obtain $\hat{\boldsymbol{\theta}}^0_{\text{OpInf}}$;

Step 4. Gradient descent loop: Initialize $j = 0$, $\hat{\boldsymbol{\theta}}^j = \hat{\boldsymbol{\theta}}^0_{\text{OpInf}}$, and step-size bounds.

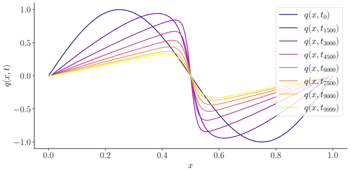
1. **Forward solve:** compute ROM solution $\tilde{\mathbf{q}}(t, \hat{\boldsymbol{\theta}}^j) = \hat{\mathbf{q}}(t)$ via reduced ODE.
2. **Gradients:** compute $\nabla_{\hat{\mathbf{q}}} \hat{\mathbf{f}}(\hat{\mathbf{q}}, \hat{\boldsymbol{\theta}}^j)$, $\nabla_{\hat{\mathbf{q}}}\, g(\hat{\mathbf{q}}, t)$ and $\nabla_{\hat{\boldsymbol{\theta}}} \hat{\mathbf{f}}(\hat{\mathbf{q}}, \hat{\boldsymbol{\theta}}^j)$.
3. **Adjoint solve:** integrate adjoint ODE backward for $\boldsymbol{\lambda}(t)$.
4. **Loss-Gradient assembly:** form $\nabla\tilde{\ell}(\hat{\boldsymbol{\theta}}^j)$.
5. **Armijo line search:** choose step size η_j by backtracking (Algorithm 1).
6. **Update:** $\hat{\boldsymbol{\theta}}^{j+1} = \hat{\boldsymbol{\theta}}^j - \eta_j \nabla\tilde{\ell}(\hat{\boldsymbol{\theta}}^j)$.
7. **Stopping criteria:** check convergence, if $\|\nabla\tilde{\ell}(\hat{\boldsymbol{\theta}}^{j+1})\|_2 \leq \epsilon$ or $j = j_{max}$, stop; else $j \leftarrow j + 1$ and repeat.

Step 5. Return $\hat{\boldsymbol{\theta}}^*$.

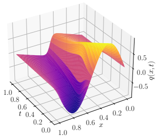
Burgers' & Fisher-KPP Equations - Training Data

$q_t + q\, q_x = \nu\, q_{xx}, \quad x \in [0,1], t \in [0,1], \nu = 0.01,$

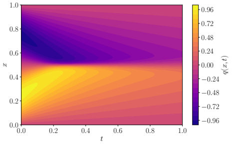
$q(x,0) = \sin(2\pi x)$, BCs: Dirichlet



(a) Profiles $q(x, t_k)$ at selected times t_k.



(b) $q(x,t)$ for $\nu = 0.01$.

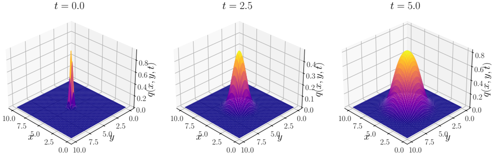


(c) Contour of $q(x,t)$.

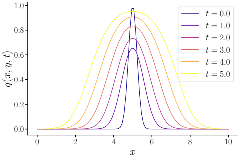
After FD discr. scheme $N_x = 2^7$, $\Delta t \approx 0.0001 \Rightarrow \mathbf{Q} \in \mathbb{R}^{130\times 10000}$.

$q_t = \mathfrak{D}(q_{xx} + q_{yy}) + \rho\, q(1-q), \quad x, y \in [0,10], \ t \in [0,5], \ \mathfrak{D} = 0.1,$

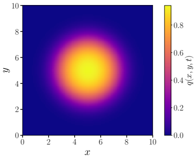
$\rho = 1, \ q(x,y,0) = \exp\Big(-10[(x-5)^2 + (y-5)^2]\Big)$, BCs: Neumann



(a) $q(x,y,t)$ for $\mathfrak{D} = 0.1, \rho = 1$.



(b) Profiles of $q(x,y,t_k)$.

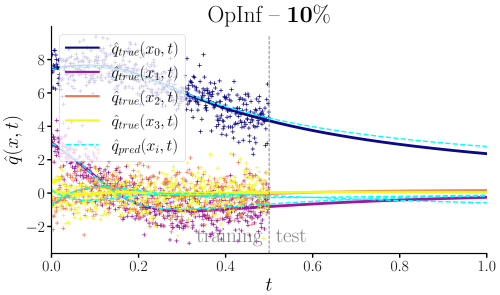


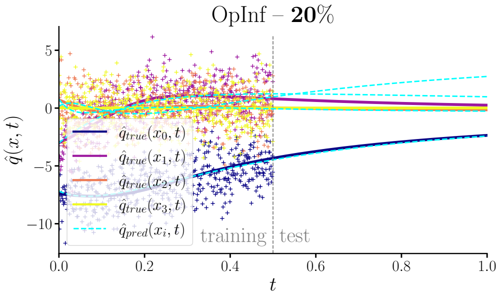
(c) Contour of $q(x,y,t=5)$.

FD discr. with $N_x = N_y = 100$, $\Delta t = 0.005 \Rightarrow \mathbf{Q} \in \mathbb{R}^{10000\times 1000}$.

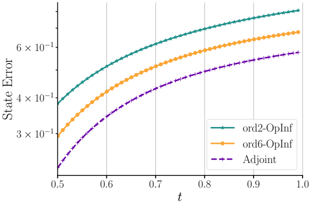
Burgers' Results I

Data perturbation: noise $\sim \mathcal{N}(0, \sigma^2)$, $\quad \sigma = \frac{\text{pct}}{100} \max_{1 \le j \le k} \|\mathbf{q}(t_j)\|_2$

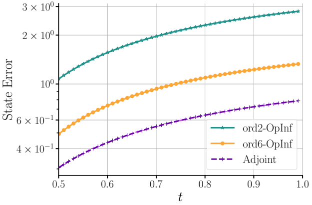




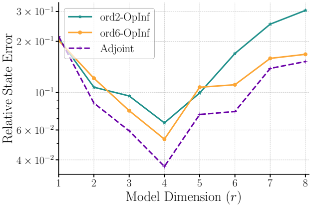
Burgers' Results II



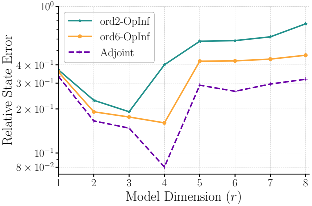
(a) 10% of noise level.



(b) 20% of noise level.



(c) 10% of noise level.

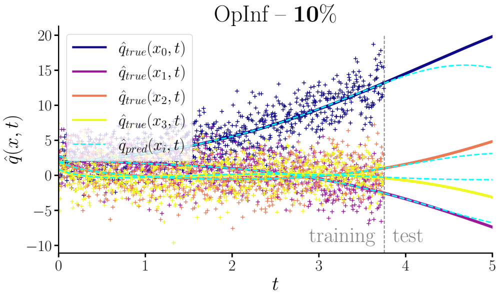


(d) 20% of noise level.

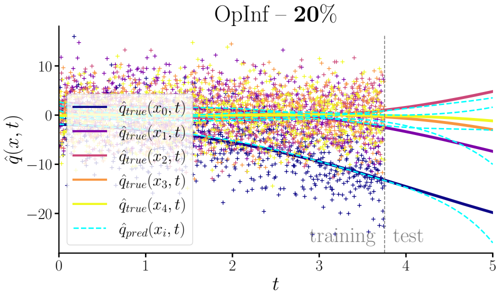
(a)-(b) Prediction error vs. time; **pred. error** $= \|\hat{\mathbf{q}}_{\text{true}}(t) - \hat{\mathbf{q}}_{\text{pred}}(t)\|_2, \quad t \in [0.5, 1]$

(c)-(d) Relative error vs r; **rel. error** $= \left(\|\hat{\mathbf{q}}_{\text{true}}(t) - \hat{\mathbf{q}}_{\text{pred}}(t)\|_2\right) / \|\hat{\mathbf{q}}_{\text{true}}(t)\|_2, \quad t \in [0, 1]$

FKPP Results I



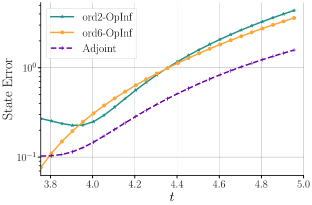
10% noise pertubation



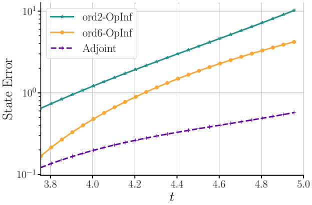
20% noise pertubation

- Across varying levels of data sparsity, no significant performance gap between OpInf/Adjoint.
- Under increasing noise perturbation (above figures), the Adjoint method demonstrates superior robustness in trajectory fidelity compared to OpInf

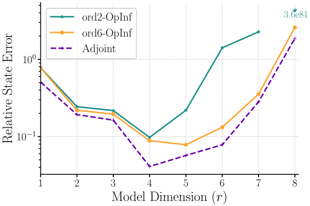
FKPP Results II



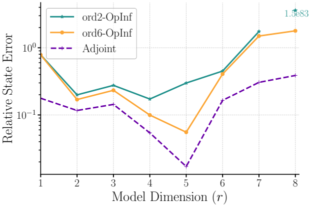
(a) 10% of noise level.



(b) 20% of noise level.



(c) 10% of noise level.



(d) 20% of noise level.

(a)-(b) Prediction error vs. time; **pred. error** $= \|\hat{\mathbf{q}}_{\text{true}}(t) - \hat{\mathbf{q}}_{\text{pred}}(t)\|_2, \ \ t \in [3.75, 5]$

(c)-(d) Relative error vs r; **rel. error** $= \left(\|\hat{\mathbf{q}}_{\text{true}}(t) - \hat{\mathbf{q}}_{\text{pred}}(t)\|_2\right) / \|\hat{\mathbf{q}}_{\text{true}}(t)\|_2, \ \ t \in [0, 5]$

Conclusions

1. **Robustness to imperfect data:**
   - Global-loss minimization filters out spurious fluctuations
   - Delivers more reliable parameter estimates than OpInf, particularly with noisy data
2. **Cost efficiency:**
   - One forward–backward solve per iteration, independent of parameter count
   - Scales favorably for large/high-fidelity models
3. **Modular flexibility:**
   - Compatible with various integrators and loss functionals
   - Easily incorporates extra physics constraints or regularization

Thank you!