Hybrid numerical methods for nonlinear conservation laws

L. Bois, E. Franck (INRIA), V. Michel-Dansac, L. Navoret, V. Vigon

Workshop EMS-TAG SciML, Annual meeting, 25/03/2025

Contents

PDE and numerical methods

ML for numerical methods

- **ML and scientific computing**: ML-like numerical methods seek to approximate infinite-dimensional objects (functions, operators, etc.) by parametric finite-dimensional objects. To achieve this, ML primarily relies on data, while scientific computing mainly uses physical constraints such as PDEs.
- **Deep learning**: Neural network parametric models have had a huge impact on ML. It is therefore logical to use them in scientific computing, leading to SciML.
- **SciML for numerics**:
  - **Pure neural network-based** numerical methods like PINNs, Deep Ritz, Discrete PINNs, Neural Galerkin (work by B. Peherstorfer et al).
    - **Advantages**: Mesh-free, easy to use for inverse problems and optimal control (see A. Belières–Frendos's poster), and most importantly, capable of handling high-dimensional problems.
    - **Disadvantages**: poor accuracy (better with new results in optimization, see in M. Zeinhofer's talk or N. Dimola's and J. Muller's posters) few theoretical guarantees.
  - **Hybrid methods**, where a classical numerical method is improved by placing the network within the method.
    - **Advantages**: convergence, more accurate and faster than classical methods.
    - **Disadvantages**: does not fundamentally change the problems that one can tackle.

Problems considered

Objective: Improve the accuracy of numerical methods for nonlinear conservation laws:

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U})$$

like Burgers' equation, the shallow water equations, the Euler equations, etc.

- Specificities of conservation laws with a nonlinear flux function F:
  - **Shocks**: the system develops discontinuities in finite time,
  - **Multiscale**: the system can have multiple propagation scales (low Mach problems, MHD, etc),

Scheme: In this talk, we mainly use the DG method, which discretizes the local weak form:

$$\partial_t \int_{\Omega_j} \boldsymbol{U}\psi_i d\boldsymbol{x} - \int_{\Omega_j} \boldsymbol{F}(\boldsymbol{U})\partial_x \psi_i d\boldsymbol{x} + \left[\boldsymbol{F}(\boldsymbol{U})\psi_i\right]_{\partial\Omega_j} = \int_{\Omega_j} \boldsymbol{S}(\boldsymbol{U})\psi_i d\boldsymbol{x},$$

where the boundary term is approximated by a numerical flux coupling neighboring cells, and the approximation

$$\boldsymbol{U}(x,t)|_{\Omega_j} \approx \boldsymbol{U}_{h,j}(t,x) = \sum_{i=1}^{n} \theta_{i,j}(t)\varphi_i(x)$$

in each cell.

Enriched DG method for balance laws

Problems considered

- Balance law: Shallow water with source term

$$\partial_t h + \partial_x (hu) = 0$$

$$\partial_t (hu) + \partial_x \left(hu^2 + \frac{1}{2} g h^2 \right) = -gh\partial_x z$$

- Equilibrium:

$$u = 0, \quad \partial_x \left(\frac{1}{2} g h^2 \right) = -gh\partial_x z \Leftrightarrow u = 0 \quad z + h = cst$$

- A tsunami is characterized by the initial condition:

$$h(t = 0, x) = h_{eq}(x) + \varepsilon \delta h(x)$$

- Example of a case where we want to simulate the dynamics of an equilibrium perturbation.

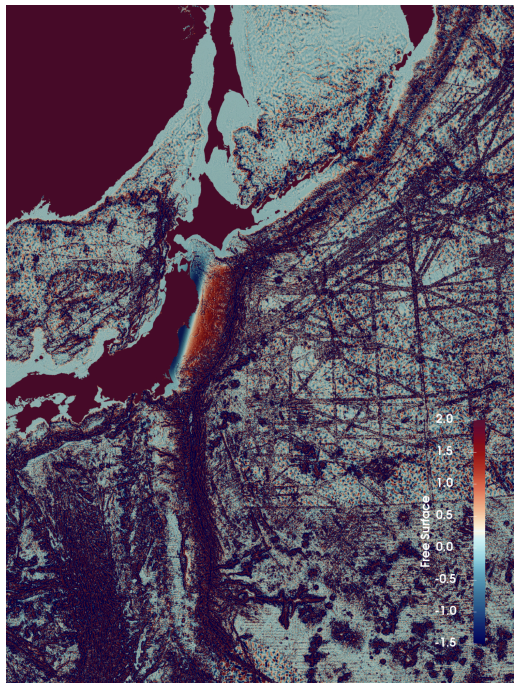


Figure 1: non-WB scheme

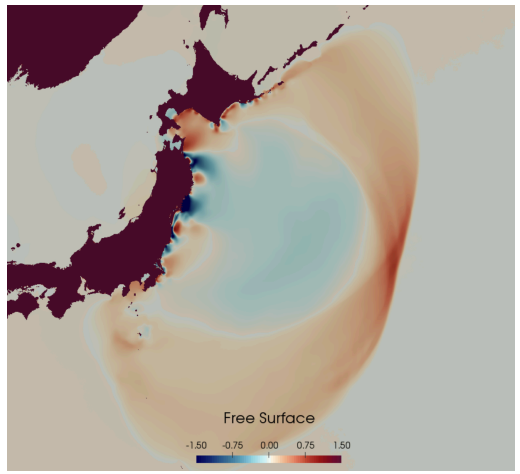


Figure 2: WB scheme (simulation by V. Michel-Dansac)

Remark: This type problem is costly to simulate with classical schemes since we need $\Delta x^p \ll \varepsilon$.
Well balanced schemes: schemes that preserve equilibria, exactly or with high accuracy.

Well-Balanced DG scheme

- **Exact WB schemes**: exist for some 1D steady states (depending on the equation), and for a few equilibria in 2D.
- **Approximate high-order WB schemes**: generally use high-order reconstruction around steady states.

Remark: In general, steady-state solutions are rewritten as solutions of algebraic or **nonlinear elliptic equations**

- Assuming $\boldsymbol{U}_{\theta,eq}$ is an approximation of the equilibrium.

Result (WB property): If $\boldsymbol{U}_{\theta,eq}$ is an equilibrum and if we choose the trial and test spaces as:

$$V_1 = \left[\boldsymbol{U}_{\theta,eq}, (\boldsymbol{x}-\boldsymbol{x}_j), ..., \frac{1}{(q-1)!}(\boldsymbol{x}-\boldsymbol{x}_j)^{q-1}\right] \text{ or } V_2 = \left[\boldsymbol{U}_{\theta,eq}, (\boldsymbol{x}-\boldsymbol{x}_j)\boldsymbol{U}_{\theta,eq}, ..., \frac{1}{q!}(\boldsymbol{x}-\boldsymbol{x}_j)^{q}\boldsymbol{U}_{\theta,eq}\right]$$

and the quadrature rules are exact, then the scheme is exactly well-balanced.

Well-Balanced DG scheme

Result (A-WB property): We consider a scalar problem. If you choose V_2 to be the same space as before, we obtain the projection error:

$$\| u - P_h(u) \|_{L^2} < C \left| \frac{u}{u_{\theta,eq}} \right|_{H^{q+1}(\Omega)} (\Delta x)^{q+1} \| u_\theta \|_{L^\infty(\Omega)}$$

- We expect, for V_1 a result like:

$$\| u - P_h(u) \|_{L^2} < C \, | u - u_{\theta,eq} |_{H^{q+1}(\Omega)} (\Delta x)^{q+1} \| u_\theta \|_{L^\infty(\Omega)}$$

- The result can be generalized to systems as well.

Conclusion: The better the approximation of the equilibrium in the basis (in semi norm H^{q+1}), the smaller the error around the equilibrium (by continuity).

Prior on the equilibrium and PINNs

Question: How to construct a good approximation (in the H^{q+1} sense) of a large family of equilibria ?

- **Our proposition**: solve a parametric problem for the equilbrium with **PINNs**.
- Advantages:
  - Able to tackle large dimensional problems and, consequently, parametric problems,
  - Provide smooth approximations,
  - Learn with a physical loss function so that the derivatives are well-reconstructed. We can add losses for the residual derivatives,
  - Easy to add data if necessary.
- We consider a parametric problem:

$$\partial_x F(\boldsymbol{U}, \alpha) = \boldsymbol{S}(\boldsymbol{U}, \beta), \;\; P\boldsymbol{U} = \boldsymbol{g}$$

  with the parameters $\boldsymbol{\mu} = (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{g})$
- We solve:

$$\min_{\theta} \int_{\mathbb{R}^p} \int_{\Omega} \| \, \partial_x F(\boldsymbol{U}_\theta(\boldsymbol{x}, \boldsymbol{\mu})), \boldsymbol{\alpha}) - \boldsymbol{S}(\boldsymbol{U}_\theta(\boldsymbol{x}, \boldsymbol{\mu})) \, \|^2 \, d\boldsymbol{x} d\mathbb{P}(\boldsymbol{\mu}) \approx \sum_{i=1}^{N} \| \, \partial_x F(\boldsymbol{U}_\theta(\boldsymbol{x}_i, \boldsymbol{\mu}_i)), \boldsymbol{\alpha}_i) - \boldsymbol{S}(\boldsymbol{U}_\theta(\boldsymbol{x}_i, \boldsymbol{\mu}_i)) \, \|^2$$

where we impose, in the parametric model (network), $P\boldsymbol{U}_\theta = \boldsymbol{g}$.

Results I

- Shallow water with source term

$$\partial_t h + \partial_x Q = 0$$

$$\partial_t Q + \partial_x \left(\frac{Q^2}{h} + \frac{1}{2} g h^2 \right) = -gh\partial_x z$$

- Equilibrium:

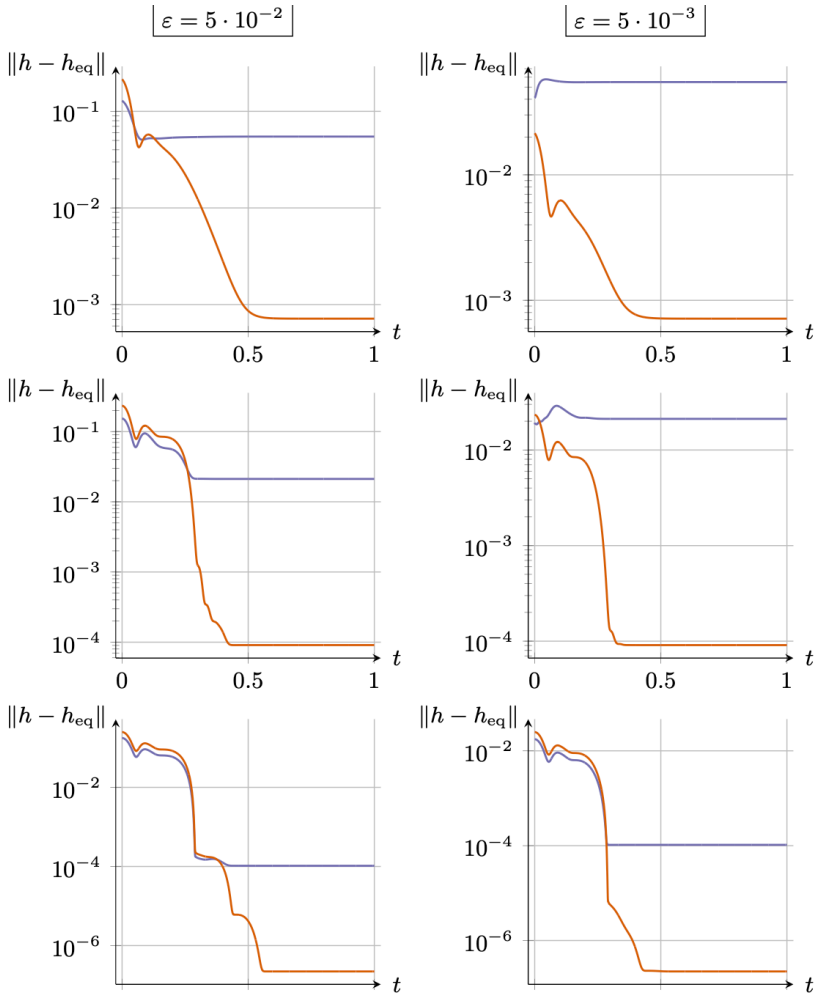
$$Q = Q_0, \quad \left(1 - \frac{Q_0^3}{g h_{eq}(x, \mu)} \right) \partial_x h(x, \mu) + \partial_x z(x, \alpha, \beta) = 0$$

avec $\mu = (h_0, Q_0, \alpha, \beta)$ and h_0 the left bc.

- **Gain for mesh with 10 cells** between classical and enriched DG:

	minimum gain		average gain		maximum gain	
q	h	Q	h	Q	h	Q
0	21.28	17.40	309.84	269.59	1562.20	1628.39
1	7.47	5.47	161.16	129.90	845.97	729.03
2	4.37	5.02	96.54	102.36	707.41	704.55

- Pertubation of steay states:



Results II

- Radial Euler equation with gravity

$$\partial_t \rho + \partial_r Q = 0$$

$$\partial_t Q + \partial_x \left(\frac{Q^2}{\rho} + p \right) = -\frac{2}{r}\frac{Q^2}{\rho} - \rho \partial_r \varphi$$

$$\partial_t E + \partial_r \left(\frac{Q}{r}(E+p) \right) = -\frac{2}{r}\frac{Q}{r}(E+p) - Q\partial_r \varphi$$

$$\frac{1}{r^2}\partial_r(r^2 \partial_r \varphi) = 4\pi\rho G$$

with $p(\rho) = \kappa\rho^\gamma$.

- Equilibrium:

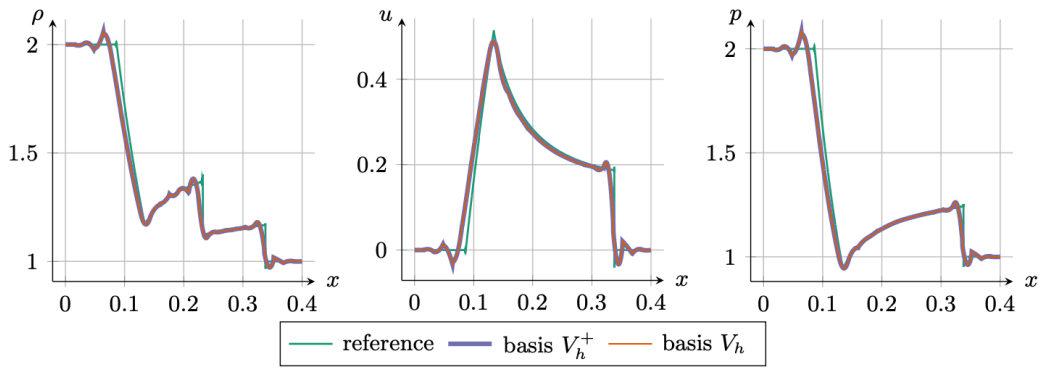
$$Q = Q_0, \ \partial_r(r^2 \kappa\gamma\rho^{\gamma-2}\partial_r\rho) = 4\pi\rho G$$

avec $\mu = (\kappa, \rho)$ and $\rho(0,\mu) = 1$, $\partial_r\rho(0,\mu) = 0$

- **Gain for mesh with 10 cells** between classical and enriched DG:

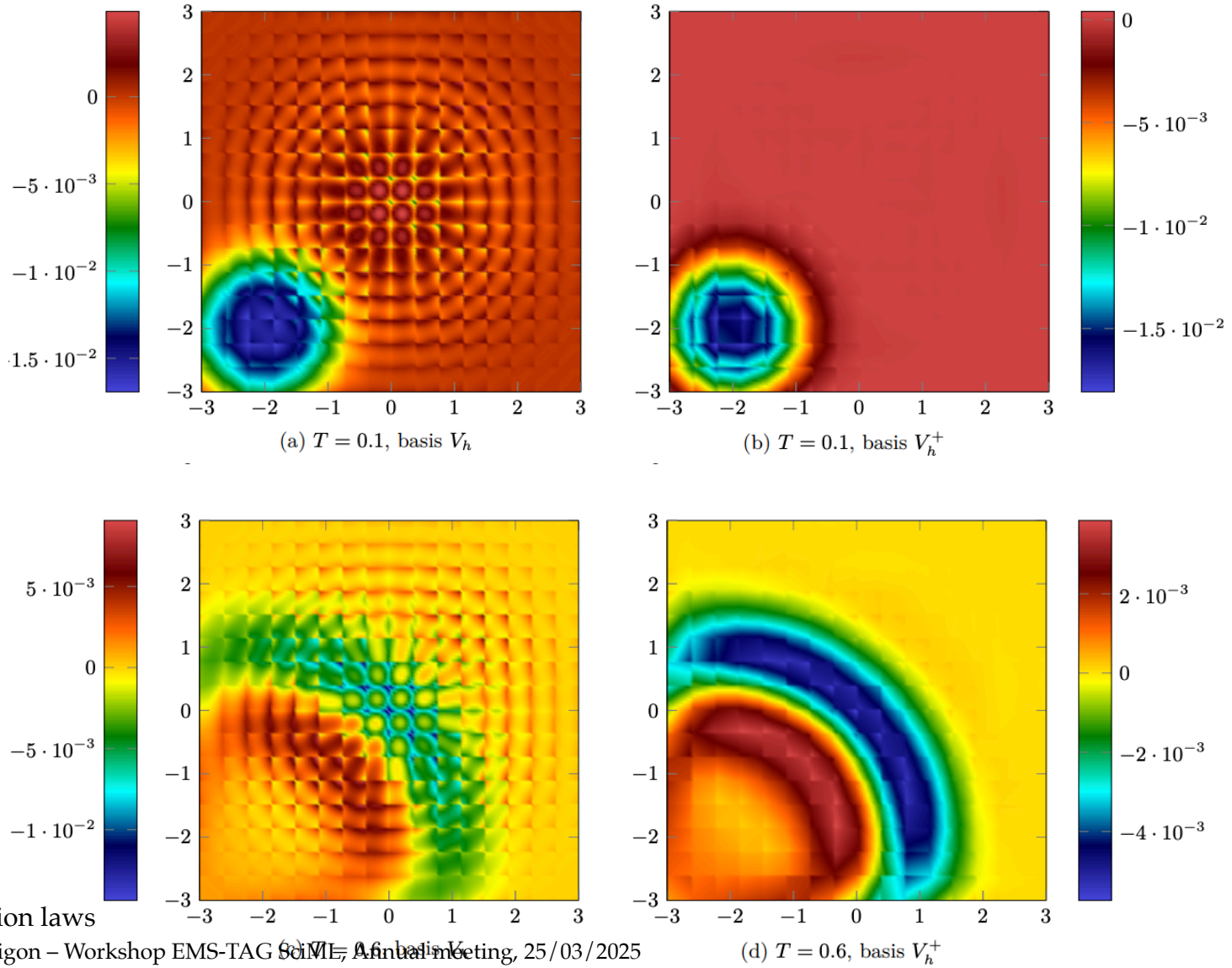
	minimum gain			average gain			maximum gain		
q	ρ	Q	E	ρ	Q	E	ρ	Q	E
0	19.14	2.33	17.04	233.48	3.73	197.28	510.42	4.48	371.87
1	7.61	8.28	6.98	158.25	188.92	130.57	1095.68	1291.90	1024.59
2	0.14	0.22	2.99	12.11	16.55	23.73	89.47	109.93	169.28

- Similar results for temperature dependant pressure law and steady states.
- Time evolution and new basis:



Results III

- Last test case:
  - 2D shallow water,
  - 3 parameters (5D problem),
  - divergence-free mometum (non-treated in a satisfying way by the WB comunity):



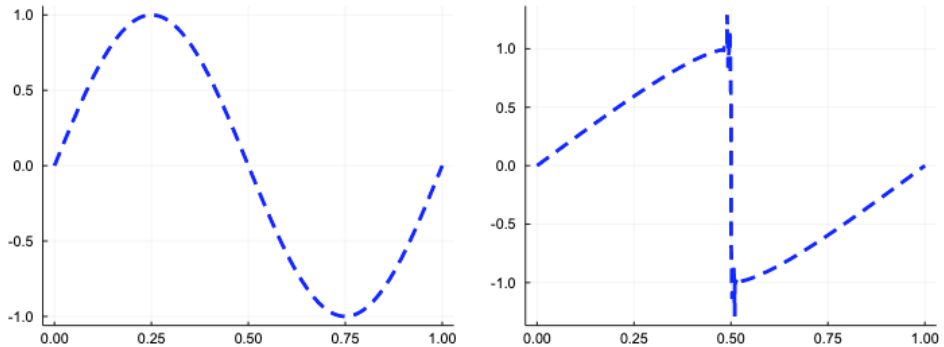
(a) $T = 0.1$, basis V_h (b) $T = 0.1$, basis V_h^+

(c) $T = 0.6$, basis V_h (d) $T = 0.6$, basis V_h^+

Stabilization for conservation laws

Shock, oscillations, and the discontinuous Galerkin method

- Nonlinear conservation laws create discontinuities in finite time.
- **Low-order methods**, like finite volumes, are too dissipative;
- **high-order methods**, like DG, can produce oscillations around shocks.



- How to mitigate the oscillations:
  - **a priori limiting** methods: predict the oscillations and locally reduce the polynomial degree.
  - **a posteriori limiting** methods: locally reduce the polynomial degree on troubled cells and recompute the cells.
  - **artificial viscosity**: add a viscosity term to dampen the oscillations.

Objective: Design a good artificial viscosity term using ML. Two approaches:

- "Supervised approach", using known viscosities: J. Hesthaven and al.
- "Closed loop Optimal control approach": our work, and also M. Caldana's (Mox, poster).

Artificial viscosity design I

Remark: Assume that we can generate reference solutions $\boldsymbol{U}_{ref}(t,x)$ (e.g. fine numerical simulations).

- **Goal**: Minimize the error between the numerical and reference solutions for a subset of trajectories associated with initial data distribution $\mathbb{P}(\boldsymbol{U}_0)$:

$$\int \int_0^T \| \boldsymbol{U}_{ref}(t,x) - \boldsymbol{U}_h(t,x) \|_?^2 \, dt d\mathbb{P}(\boldsymbol{U}_0)$$

under the following constraints, satisfied by $\boldsymbol{U}_h$:

$$\partial_t \boldsymbol{U}_h + \partial_x^{DG} \boldsymbol{F}(\boldsymbol{U}_h) = \partial_x^{DG}(D_\theta(\boldsymbol{U}_h)\partial_x \boldsymbol{U}_h),$$

where the control is the viscosity $D_\theta(\boldsymbol{U}_h)$ parametrized, by a neural network.

Scheme (Key point of the method): We not compute the adjoint to compute the gradient. We code the scheme in a differential framwork and the automatic differentiation passes through the time steps.

Artificial viscosity design II

Remark: The greater the number of time steps through which we differentiate, the more expensive the cost of computing the gradient. But, to see positive and negative effect to the viscosity, we need to differentiate through a significant number of time steps (around 100).

- **Details**:
  - We use a small convolutive neural network with $\boldsymbol{U}$ and $\boldsymbol{F}(\boldsymbol{U})$ at each Gauss point as features.
  - We generate random initial conditions by taking a sum of random Fourier modes.
  - The loss used, computed on the fine grid, is:

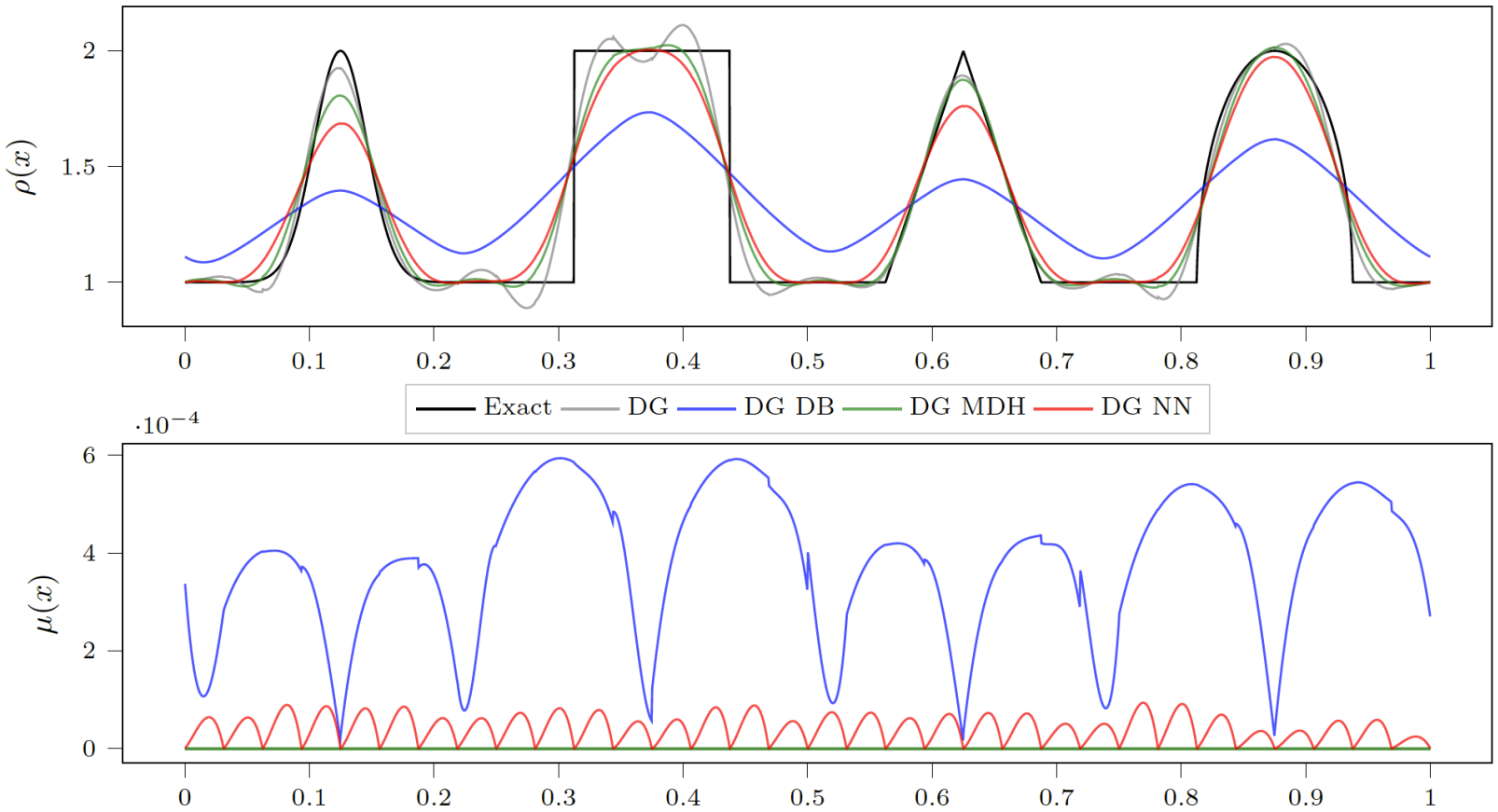
$$\| \boldsymbol{U}_{ref}(t,x) - \boldsymbol{U}_h(t,x) \|_?^2 = \omega_{osc} \| \Delta \boldsymbol{U}_{ref}(t,x) - \Delta \boldsymbol{U}_h(t,x) \|_1 + \omega_{acc} \| \boldsymbol{U}_{ref}(t,x) - \boldsymbol{U}_h(t,x) \|_{L^2} + \omega_{vis} \| D(\boldsymbol{U}_h) \|_{L^2}$$

  - The viscosity model is mutiplied by a scaling depending on h.
  - The first loss penalizes oscillations, the second the accuracy, and the third large visosities. In practice, the second loss also penalizes large viscosities.
  - We learn on coarse meshes.

- **Work by M. Caldana**: other networks, other inputs and outputs, other losses and more 2D cases.

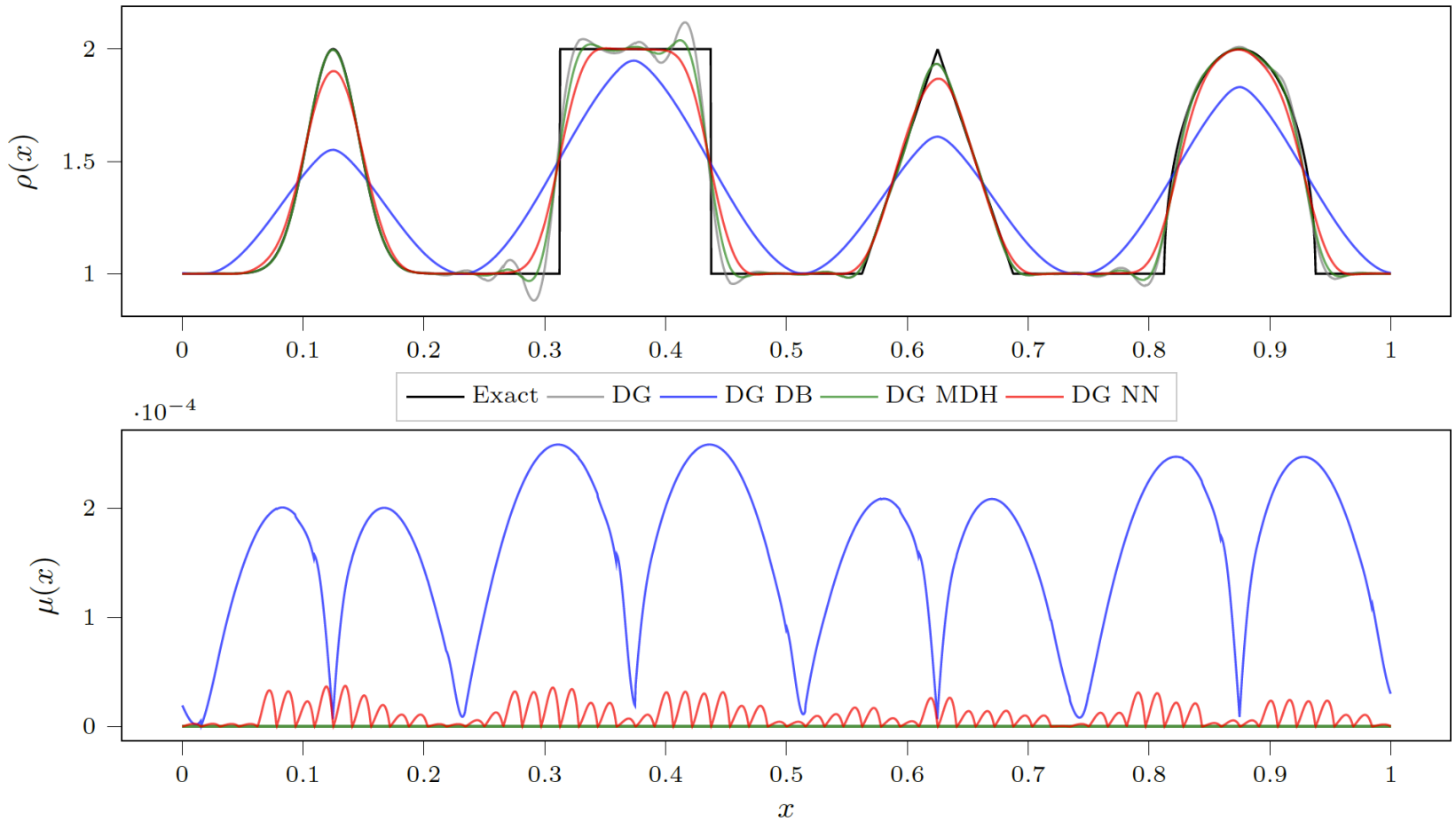
Results I

- We train and solve on the **advection equation**.
- We learn on 32 cells and test on varying meshes (below, 32 cells).



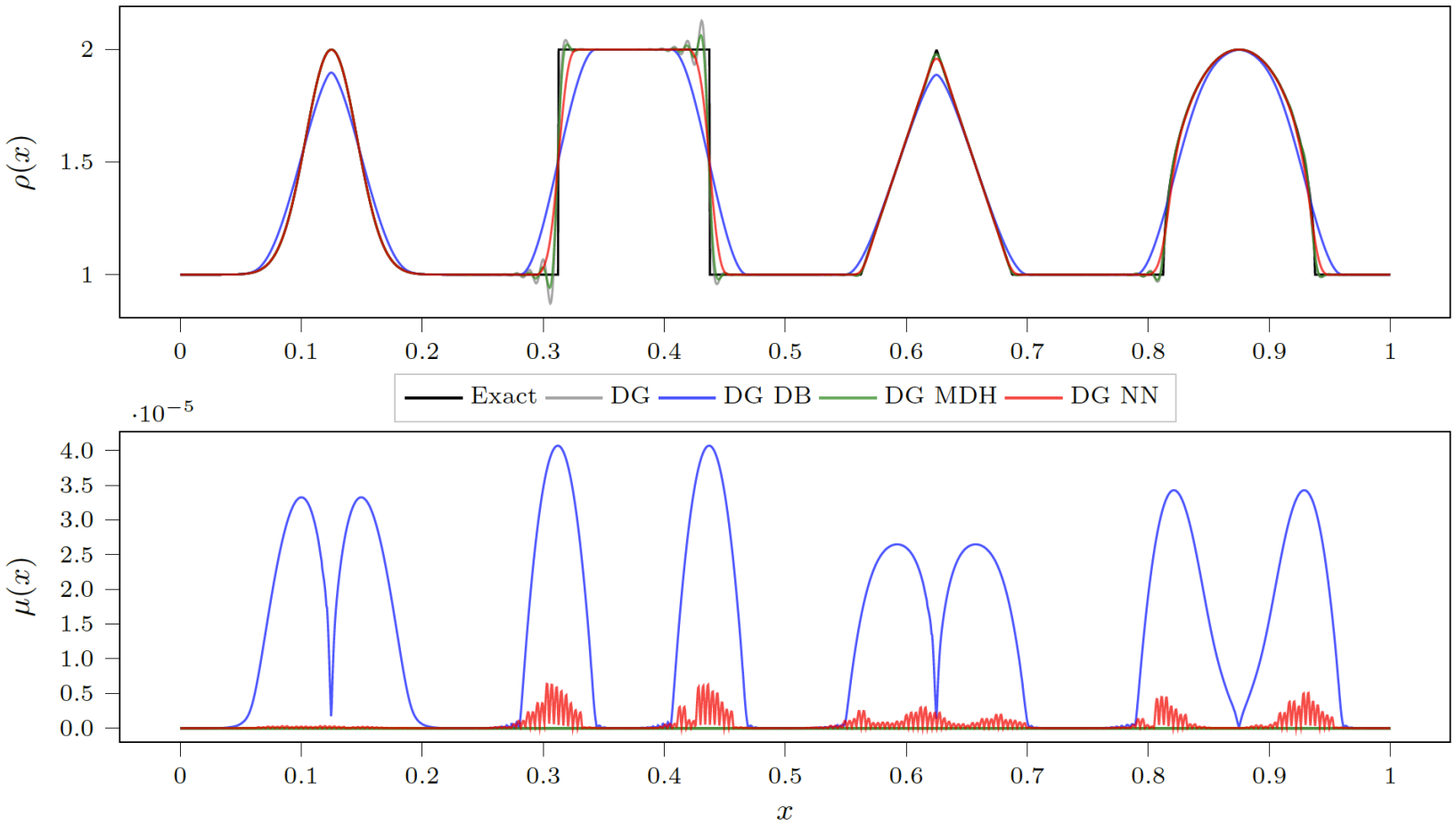
Results I

- We train and solve on **advection equation**.
- We learn on 32 cells and test on varying meshes (below, 64 cells).



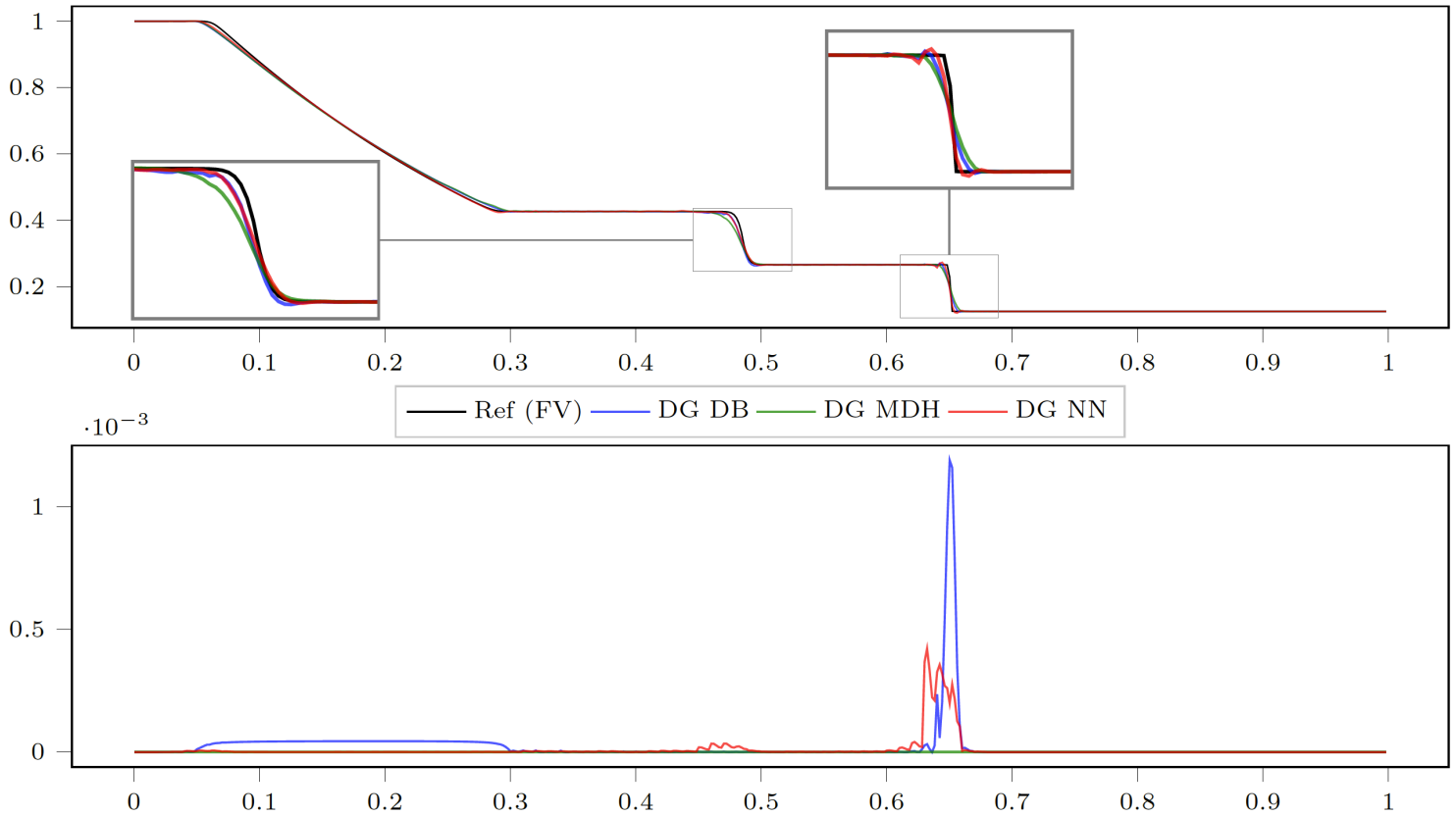
Results I

- We train and solve on **advection equation**.
- We learn on 32 cells and test on varying meshes (below, 256 cells).



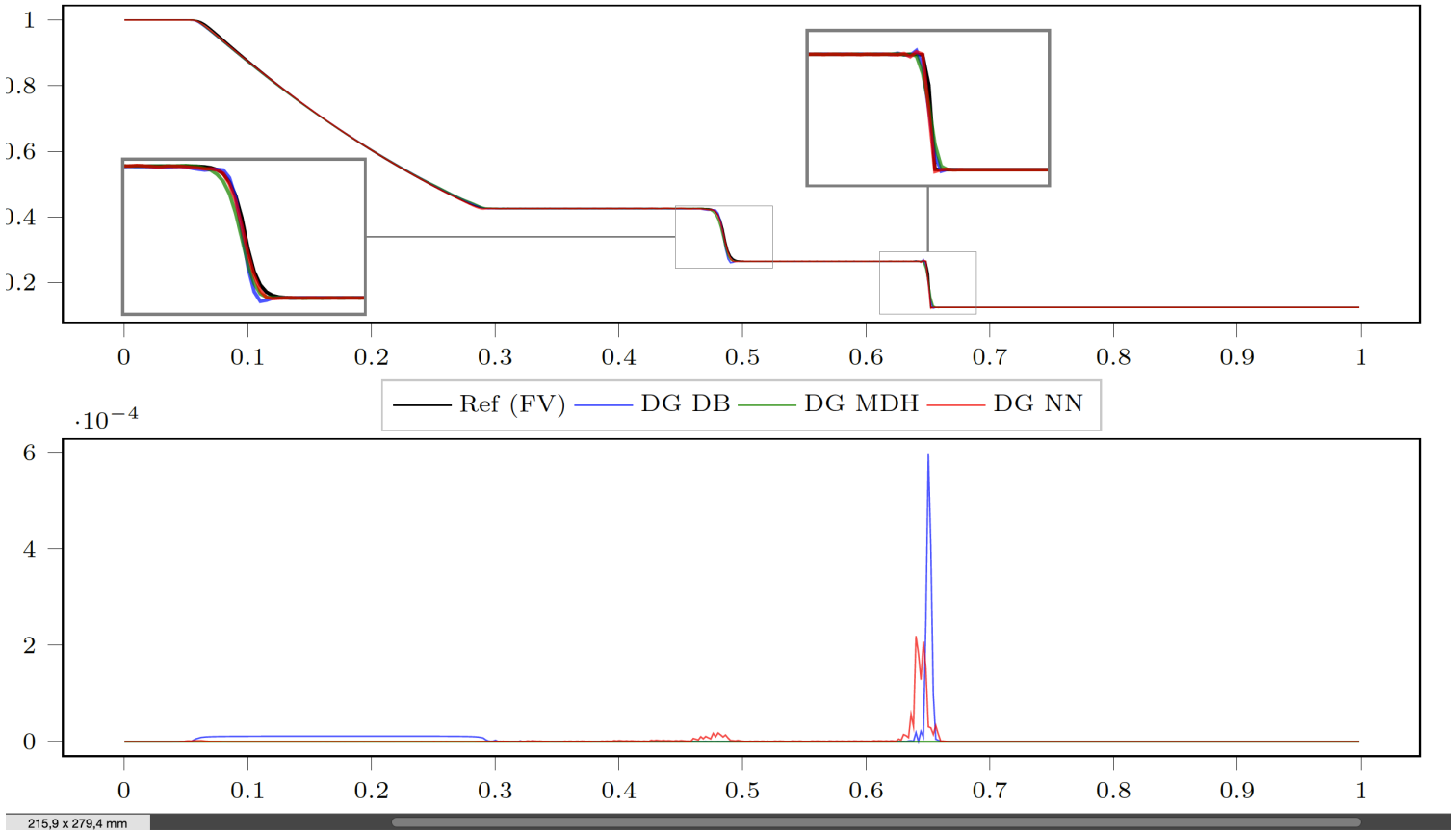
Results II

- Euler equations, Sod problem: mesh with 32 cells



Results II

- Euler equations, Sod problem: mesh with 64 cells



LBM scheme

Objective: Apply a similar idea to another type of scheme where arficial viscosity modeling is less well-understood: the Lattice Boltzmann method.

- **LBM**: We replace a nonlinear PDE by a larger linear PDE with a nonlinear relaxation term, applying a very simple scheme on the new PDE and take the limit.

- Burgers' equation

$$\partial_t \rho + \partial_x \left(\frac{\rho^2}{2} \right) = 0$$

- D1Q2 model

$$\partial f_- - \lambda \partial_x f = -\frac{f_- - f_-^{eq}(\rho)}{\tau}$$

$$\partial f_+ + \lambda \partial_x f = -\frac{f_+ - f_+^{eq}(\rho)}{\tau}$$

When $\tau \to 0$, the quantity $\rho_h = f_- + f_+$ tends towards the solution ρ of Burgers' equation.

Remark (Relaxation model): We can write this discrete kinetic model for any consevation laws.

LBM scheme and artificial visocisity

Scheme (LBM idea): Use a splitting scheme for the second model, which allows us to have:

- A constant transport that we can solve exactly, if $\Delta t = \lambda \Delta x$,
- A local relaxation step solved with a θ-scheme, which is explicit since the relaxation step conserves ρ.

- **Time scheme** for

$$\partial_t \boldsymbol{f} + \Lambda \partial_x \boldsymbol{f} = -\frac{\boldsymbol{f} - \boldsymbol{f}^{eq}(\boldsymbol{U})}{\tau}$$

- Transport:

$$\boldsymbol{f}^*(x_j) = \boldsymbol{f}^n(x_j - \Lambda \Delta t)$$

- Relaxation:

$$\boldsymbol{f}^{n+1}(x_j) = \boldsymbol{f}^*(x_j) + \omega\big(\boldsymbol{f}^{eq}\big(\boldsymbol{U}(x_j)\big) - \boldsymbol{f}^*(x_j)\big)$$

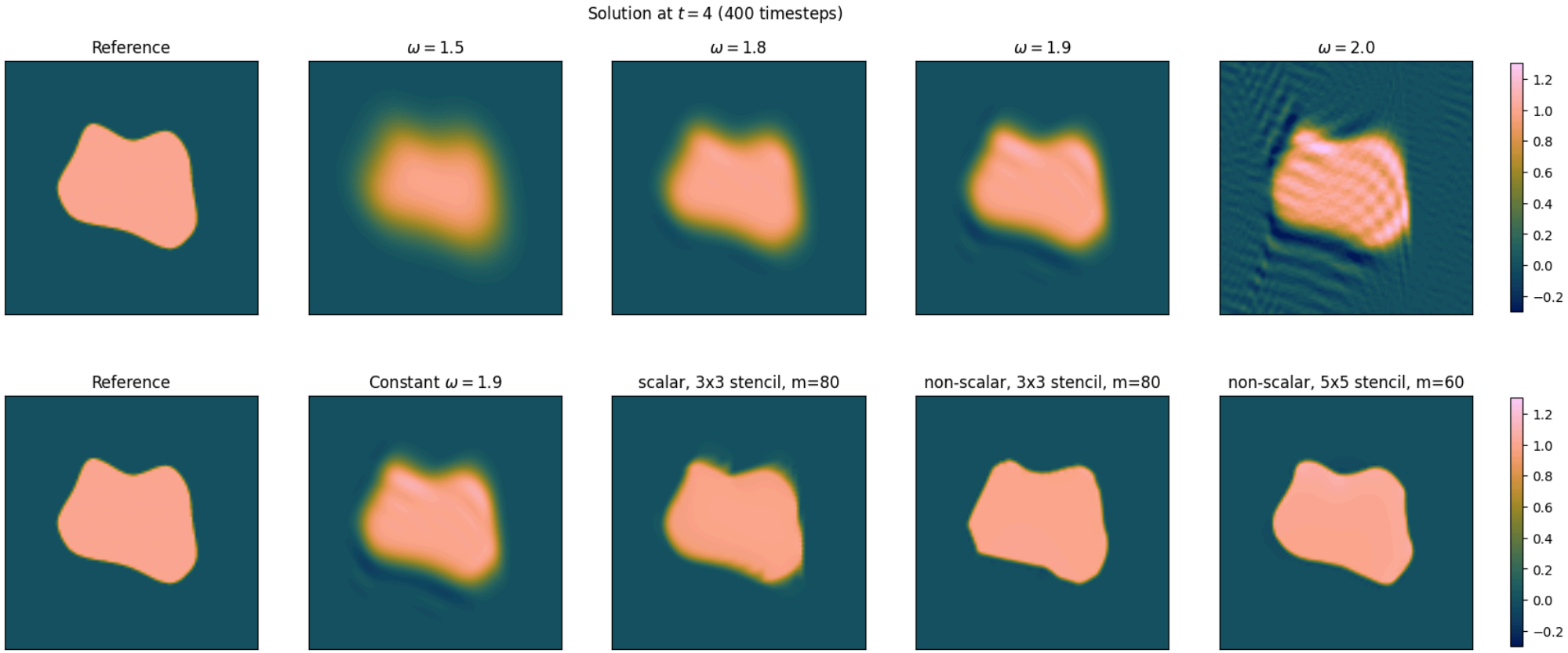
- $\omega = 2$ kills the dissipation but generates dispersive effects.
- To tune the viscosity of the scheme we replace the relaxation step by:

$$\boldsymbol{f}^{n+1}(x_j) = \boldsymbol{f}^*(x_j) + \omega(\boldsymbol{f}, \rho)\big(\boldsymbol{f}^{eq}\big(\boldsymbol{U}(x_j)\big) - \boldsymbol{f}^*(x_j)\big)$$

Objective: Using similar method as before, we can learn the viscosity term (local or nonlocal).

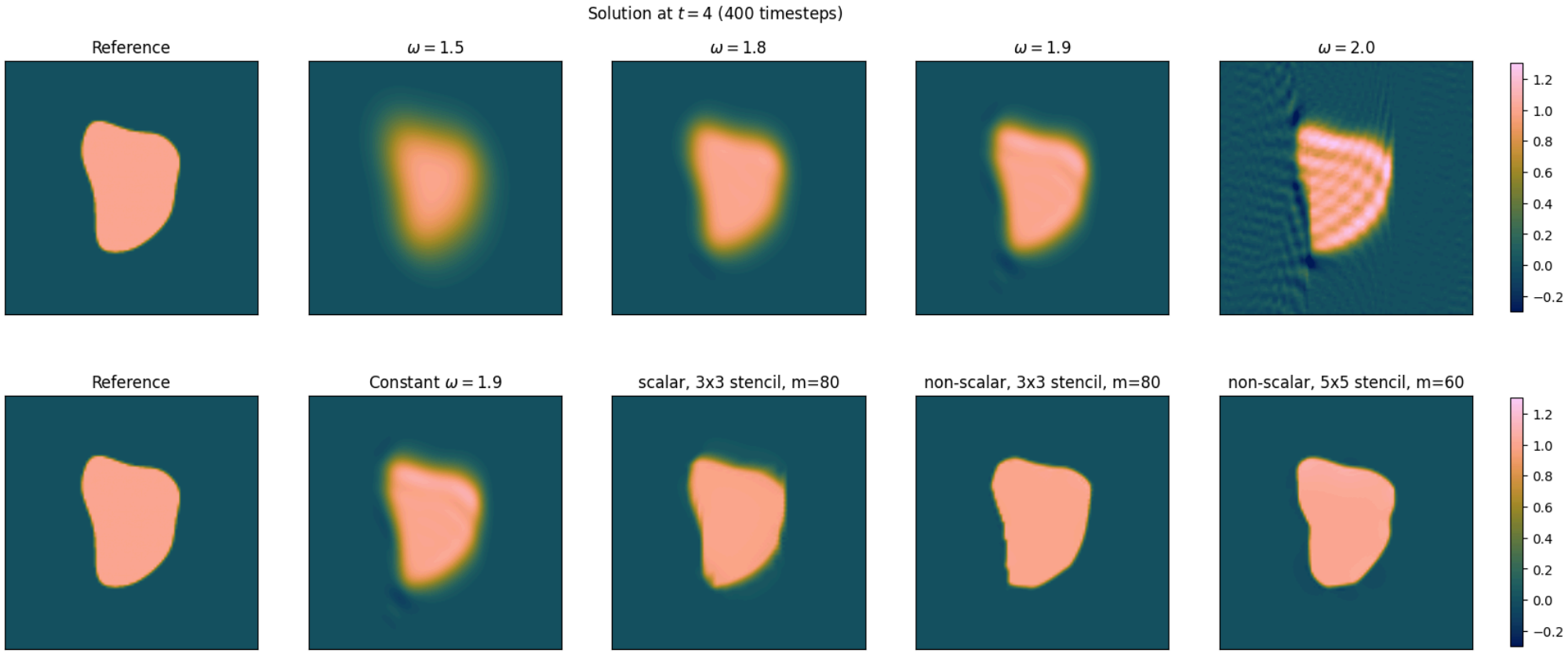
Results I

- We solve oblique advection equation with LBM and discontinuous initial conditions. Periodic BC and one complete revolution.
- Example 1:



Results II

- We solve oblique advection equation with LBM and discontinuous initial conditions. Periodic BC and one complete revolution.
- Example 2:



Remarks and open questions

Remark (LBM viscosity): The neural network proposes an interesting strategy with anti-diffusion behind the discontinuity:

- We have a sharp interface, but the shape changes. How to avoid that shortcoming?
- Can we design an analytic expression for the viscosity following the idea discovered by the neural network?
- How to keep the locality in space?

- Extension to the Euler equation is, for now, an open question.

Remark (LBM and DG): DG and LBM (applied to the Euler equations) may generate **instabilties** and crash the simulation. It this case, the training blows up. How can we treat that (PDE discovery also suffers from this)?

- Supervised pre-training (sufficient?):
  - how to have an idea of the solution ?
  - can we train on one time step ?
- Gradient-free methods like RL ?

- In both cases, we use our metric to detect the numerical oscillations. Can we learn the metric ?

Conclusion

Conclusion

Conclusion (Summary): We have proposed two hybrid methods to improve the accuracy of numerical methods for conservation laws.

- **Enriched DG**: to improve the accuracy of the scheme around equilibria.
- **DG and LBM with learned artificial viscosity**: to mitigate the oscillations around shocks waves.

Conclusion (Summary): We increased the accuracy or decreased the computational cost, keeping the convergence properties of the classical methods. For the viscosity, the gains are quite small. Can this be improved?

- unstructured meshes ?
- new metric ?
- less-studied scheme than DG ?

Conclusion (Announcement): Our team "MACARON" (Inria Strasbourg) often has Master's theses, or PhD and postdoc positions. Every year, permanent positions are open at Inria; applications are welcome!

Reference

- *An artificial neural network as a troubled-cell indicator*, D Ray, JS Hesthaven, JCP 2018,
- *Controlling oscillations in high-order discontinuous Galerkin schemes using artificial viscosity tuned by neural networks* N Discacciati, JS Hesthaven, D Ray, JCP 2020,
- *Discovering artificial viscosity models for discontinuous galerkin approximation of conservation laws using physics-informed machine learning*, Matteo Caldana, Paola F Antonietti, Luca Dede, JCP 2024
- *A machine learning approach to enhance the SUPG stabilization method for advection-dominated differential problems*, Tommaso Tassi, Alberto Zingaro, Luca Dede, Mathematics in Engineering, 2023
- *Optimal control deep learning approach for viscosity design in DG schemes*, L. Bois, E. Franck, L. Navoret, V. Vigon. Journal of Scientific Computing, 2024
- *Approximately well-balanced Discontinuous Galerkin methods using bases enriched with Physics-Informed Neural Networks*, V. Michel. Dansac, E. Franck, L. Navoret. JCP 2024
- *Accelerate Newton convergence for nonlinear elliptic PDE using PINO deep learning approach* , E. Franck, R. Hild, V. Vigon, V. Michel-Dansac, J. Aghili. Communications in Nonlinear Science and Numerical Simulation, January 2025