A data-driven approach to predict artificial viscosity in high-order solvers

Deep Ray
Email: deepray@usc.edu
Website: deepray.github.io

Universität Würzburg
May 14, 2021

USC Viterbi
School of Engineering

Machine learning for scientific computing

Numerical Methods:

- Solving ODEs/PDEs
- Uncertainty quantification
- Reduced-order modeling
- Inverse problems
- Operator infererence

Problem-dependent parameters

High cost

Lack of closed-form expression

Missing/hidden physics

Computational bottlenecks

Machine learning for scientific computing

Numerical Methods:

- Solving ODEs/PDEs
- Uncertainty quantification
- Reduced-order modeling
- Inverse problems
- Operator infererence

Problem-dependent parameters

High cost

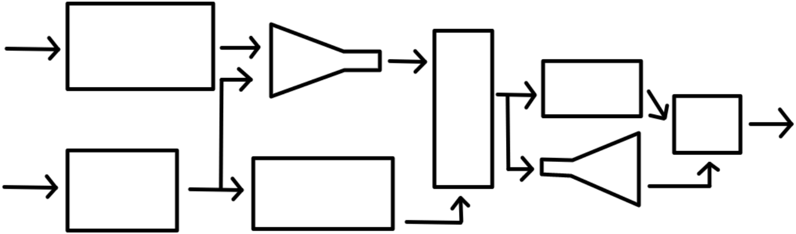
Lack of closed-form expression

Missing/hidden physics

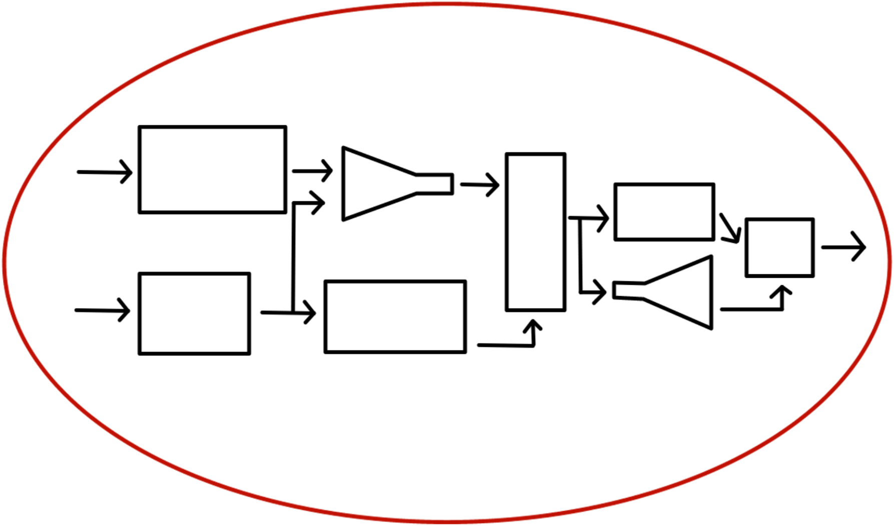
Computational bottlenecks

Data-driven solution strategies

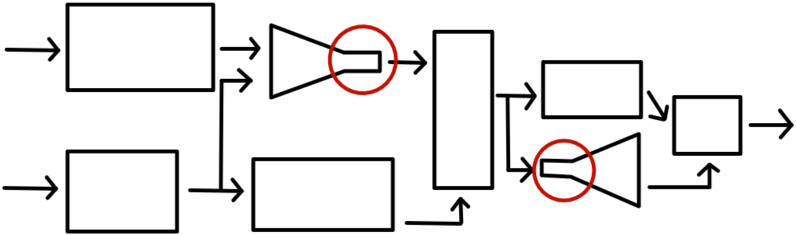
Machine learning for scientific computing



Machine learning for scientific computing



Machine learning for scientific computing

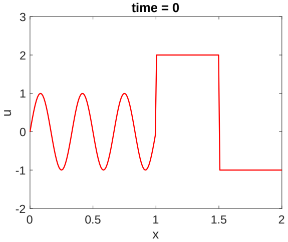


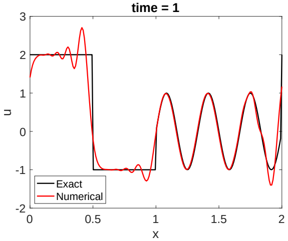
A brief outline

- Artificial viscosity in DG schemes
- Multilayer perceptrons (MLPs)
- A deep viscosity estimator for DG schemes
- Artificial viscosity for global schemes
- Extensions

Artificial viscosity for DG schemes

Handling Gibbs oscillations





Consider the conservation law

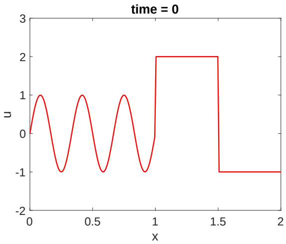
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0$$

Discontinuity + high-order solver

$$\Downarrow$$

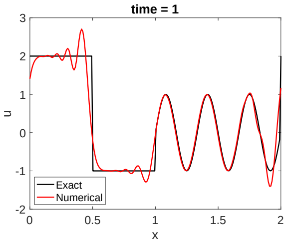
Spurious oscillations

Handling Gibbs oscillations

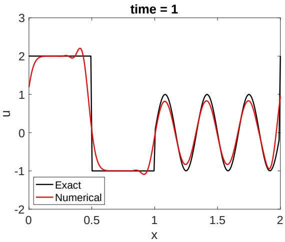


Consider the modified PDE

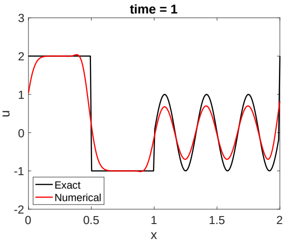
$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right)$$



$\mu = 0$

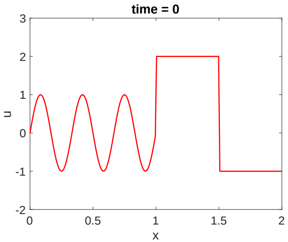


$\mu = 5.0e - 4$



$\mu = 1.0e - 3$

Handling Gibbs oscillations

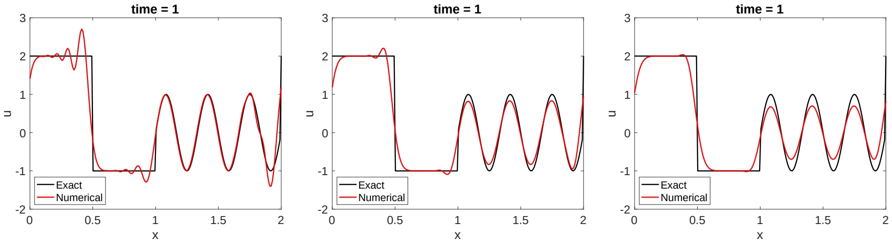


Consider the modified PDE

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right)$$

Issues:

- Where to introduce viscosity?
- How much viscosity?



$\mu = 0$ $\qquad$ $\mu = 5.0e-4$ $\qquad$ $\mu = 1.0e-3$

Artificial viscosity in DG schemes

For the modified conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \boldsymbol{f} = \nabla \cdot \boldsymbol{g}, \quad \boldsymbol{g} = \mu \boldsymbol{q}, \quad \boldsymbol{q} = \nabla u$$

Solution approximated on each element D_k, $\Omega = \bigcup_{k=1}^{K} D_k$

$$u(\boldsymbol{x}, t) \approx u_h(\boldsymbol{x}, t) = \bigoplus_{k=1}^{K} u_h^k(\boldsymbol{x}, t), \quad u_h^k(\boldsymbol{x}, t) = \sum_{i=1}^{N} u_i^k(t) \phi_i^k(\boldsymbol{x})$$

where $\{\phi_i^k\}_{i=0}^{N}$ is a basis of order m.

Solve ODE for nodal/modal coefficient vector $\boldsymbol{u}^k = [u_1^k, ..., u_N^k]^\top$

$$\frac{\mathrm{d}\boldsymbol{u}^k}{\mathrm{d}t} = \boldsymbol{L}(\boldsymbol{u}, \mu), \quad 1 \leq k \leq K$$

Popular viscosity models

Highest modal decay (MDH) model [Persson and Peraire, 2006]:

Based on the decay of modal coefficients in each element

$$S_k = \frac{\|u_h^k - \tilde{u}_h^k\|^2_{L^2(D_k)}}{\|u_h^k\|^2_{L^2(D_k)}} \quad \rightarrow \quad \text{fraction of energy in highest modes}$$

For $s_k = \log_{10}(S_k)$

$$\mu = \mu_{\max} \begin{cases} 0 & \text{if } s_k < s_0 - c_k, \\ \frac{1}{2}\left(1 + \sin\left(\frac{\pi(s_k - s_0)}{2c_k}\right)\right) & \text{if } s_0 - c_k \leq s_k \leq s_0 + c_k, \\ 1 & \text{if } s_0 + c_k \leq s_k \end{cases}$$

where $s_0 = -c_A - 4\log_{10}(m)$.

Popular viscosity models

Highest modal decay (MDH) model [Persson and Peraire, 2006]:

Based on the decay of modal coefficients in each element

$$S_k = \frac{\|u_h^k - \tilde{u}_h^k\|^2_{L^2(D_k)}}{\|u_h^k\|^2_{L^2(D_k)}} \quad \rightarrow \quad \text{fraction of energy in highest modes}$$

For $s_k = \log_{10}(S_k)$

$$\mu = \mu_{\max} \begin{cases} 0 & \text{if } s_k < s_0 - c_k, \\ \frac{1}{2}\left(1 + \sin\left(\frac{\pi(s_k - s_0)}{2c_k}\right)\right) & \text{if } s_0 - c_k \leq s_k \leq s_0 + c_k, \\ 1 & \text{if } s_0 + c_k \leq s_k \end{cases}$$

where $s_0 = -c_A - 4\log_{10}(m)$.

Problem-dependent parameters

Popular viscosity models

Averaged modal decay (MDA) model [Klökner et al., 2011]:

Can be seen as an improvement over MDH.

$$\mu = \mu_{\text{max}} \begin{cases} 1 & \text{if } \tau < 1; , \\ 1 - \dfrac{\tau - 1}{2} & \text{if } 1 \leq \tau < 3, \\ 0 & \text{if } 3 \leq \tau \end{cases}$$

where the "local" smoothness estimator τ is obtained by solving local optimization problems.

Problem-dependent parameter

Popular viscosity models

Entropy viscosity (EV) model [Guermond et al., 2011]:

Based on the entropy residual for the pair $(\eta(u), \mathcal{F}(u))$

$$\mu = \min\{\mu_\eta, \mu_{\max}\}, \quad \mu_\eta = \frac{C_\eta}{A}\left(\frac{h}{m}\right)^2 \max\left(\max_{D_k}|R(u)|, \max_{D_k}|H(u)|\right)$$

where

$$R = \frac{\eta^n - \eta^{n-1}}{\Delta t} + \frac{\nabla \cdot \mathcal{F}^n + \nabla \cdot \mathcal{F}^{n-1}}{2}$$

$$H = \left(\frac{m}{h}\right) [\![\mathcal{F}]\!] \cdot \boldsymbol{n}$$

$$A = \max_{\Omega}\left|\eta - \frac{1}{|\Omega|}\int_\Omega \eta \, d\Omega\right|$$

Problem-dependent parameters

Popular viscosity models

If problem-dependent parameters are not appropriately prescribed:

- Re-appearance of spurious oscillations.
- Excessive smearing in smooth regions.

Idea: Let a neural network estimate the local viscosity.

Multilayer perceptrons

Neural networks: supervised learning

Aim: Approximate a function

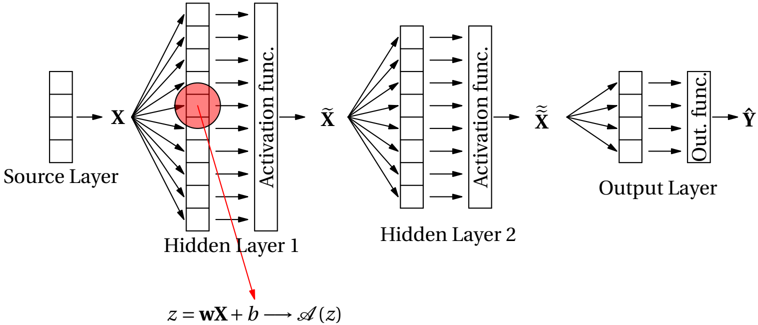
$$\boldsymbol{F} : \boldsymbol{X} \mapsto \boldsymbol{Y}, \quad \boldsymbol{X} \in \mathbb{R}^n, \ \boldsymbol{Y} \in \mathbb{R}^m \qquad \text{given} \qquad \mathbb{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_i.$$

Neural networks: supervised learning

Aim: Approximate a function

$$\boldsymbol{F} : \boldsymbol{X} \mapsto \boldsymbol{Y}, \quad \boldsymbol{X} \in \mathbb{R}^n, \ \boldsymbol{Y} \in \mathbb{R}^m \qquad \text{given} \qquad \mathbb{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_i.$$

- Multilayer perceptron (MLP):

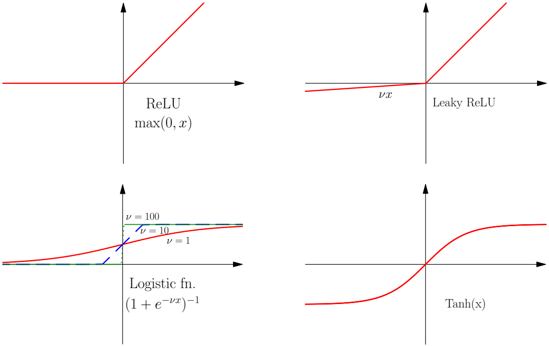


Neural networks: supervised learning

Aim: Approximate a function

$$\boldsymbol{F} : \boldsymbol{X} \mapsto \boldsymbol{Y}, \quad \boldsymbol{X} \in \mathbb{R}^n, \ \boldsymbol{Y} \in \mathbb{R}^m \qquad \text{given} \qquad \mathbb{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_i.$$

- Multilayer perceptron (MLP):



Neural networks: supervised learning

Aim: Approximate a function

$$\boldsymbol{F} : \boldsymbol{X} \mapsto \boldsymbol{Y}, \quad \boldsymbol{X} \in \mathbb{R}^n, \ \boldsymbol{Y} \in \mathbb{R}^m \qquad \text{given} \qquad \mathbb{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_i.$$

- Multilayer perceptron (MLP):

$$\widehat{\boldsymbol{Y}} = \mathcal{O} \circ H^L \circ \mathcal{A} \circ H^{L-1} \circ ... \circ \mathcal{A} \circ H^1(\boldsymbol{X}), \quad H^l(\tilde{\boldsymbol{X}}) = \boldsymbol{W}^l \tilde{\boldsymbol{X}} + \boldsymbol{b}^l.$$

- Output function $\mathcal{O}$:

Regression	$\longrightarrow$ Identity
Regression+positivity	$\longrightarrow$ Softplus: $x_i \mapsto \ln(1 + e_i^x)$
Classification	$\longrightarrow$ Softmax: $x_i \mapsto \dfrac{e_i^x}{\sum_j e_j^x}$

Neural networks: supervised learning

Aim: Approximate a function

$$\boldsymbol{F} : \boldsymbol{X} \mapsto \boldsymbol{Y}, \quad \boldsymbol{X} \in \mathbb{R}^n, \ \boldsymbol{Y} \in \mathbb{R}^m \qquad \text{given} \qquad \mathbb{T} = \{(\boldsymbol{X}_i, \boldsymbol{Y}_i)\}_i.$$

Training: Find the network parameters $\boldsymbol{\theta} = \{\boldsymbol{W}^l, \boldsymbol{b}^l\}_{1 \leq l \leq L}$ such that a suitable loss function

$$\mathcal{L} := \mathcal{L}(\boldsymbol{Y}_i, \widehat{\boldsymbol{Y}}(\boldsymbol{X}_i; \boldsymbol{\theta}))$$

is minimized over the training set $\mathbb{T}$.

Hyperparameters:

- Network size – depth and width
- Activation function
- Loss function
- Regularization technique – to avoid overfitting
- Training and validation datasets
- Optimizer: Stochastic gradient descent, AdaGrad, ADAM, etc.

A deep viscosity estimator

Training an MLP to predict viscosity

Target function $\boldsymbol{F}$: solution in $D_k \xrightarrow{\text{regression}} \mu$ in D_k

Network architecture:

- Input $\boldsymbol{X} \in \mathbb{R}^N$: nodal values of u in D_k, $N = N(m)$.
- 5 hidden layers with Leaky ReLU activation.
- Output $\boldsymbol{Y} \in \mathbb{R}^N$: nodal values of μ in D_k.

Scaling is important for generalization:

$$\boxed{\mathbf{u}} \xrightarrow{\text{scale}} \boxed{\frac{\mathbf{u}}{\max\{|u^i|\}}} \xrightarrow{\text{MLP}} \boxed{\frac{\mu}{h\Lambda}} \xrightarrow{\text{rescale}} \boxed{\mu}$$

$$\Lambda = \max|\mathbf{f}'(\mathbf{u})|$$

Training an MLP to predict viscosity

What about the training set $\mathbb{T}$?

- Using numerical solution of conservation laws (only linear adv. and Burgers).
- Target viscosity: viscosity corresponding to "best" model among MDH, MDA, EV.
- Different network for each degree m.

Note:

- The network for a given m is trained offline.
- The same network is used for any conservation law.
- No further tuning required after training.

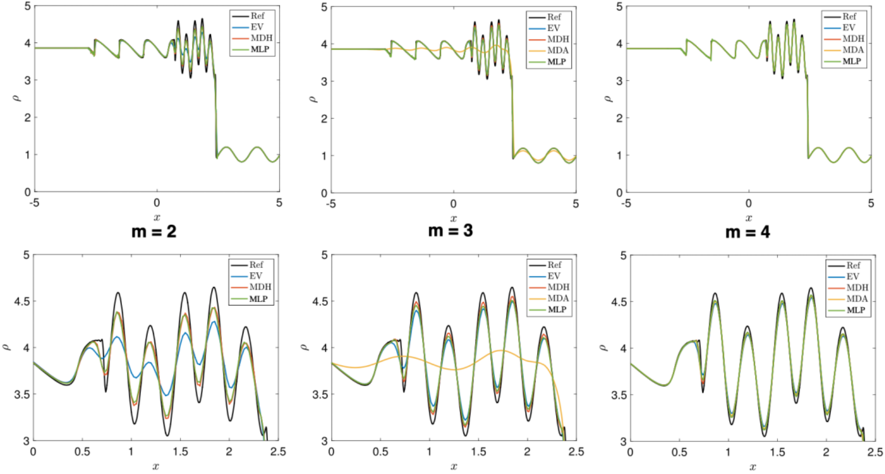
The trained network can be interpreted as an automatic, dynamic and local adaptor between MDH, MDA and EV.

Controlling oscillations in high-order Discontinuous Galerkin schemes using artificial viscosity tuned by neural networks, by Discacciati, Hesthaven and R.; JCP, 2020.

Numerical setup

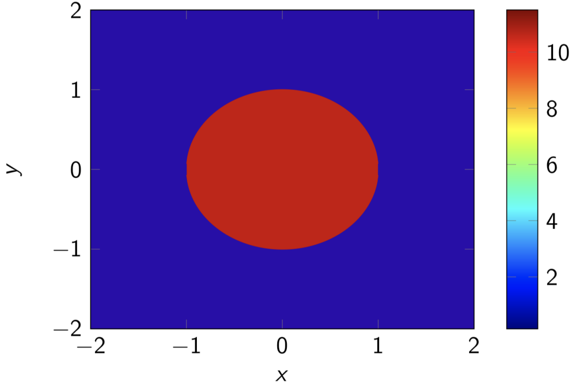
- DG scheme with Legendre basis (Jacobi polynomials in 2D).
- Triangular mesh in 2D.
- Local Lax-Friedrich numerical flux.
- Appropriate proxy variable for systems (ρ for Euler eqns.)
- Time integration with 5 stage 4th-order low-storage explicit Runge-Kutta scheme.
- Parameters for standard methods are fixed.

1D Euler equations: Shu-Osher (density based μ prediction)



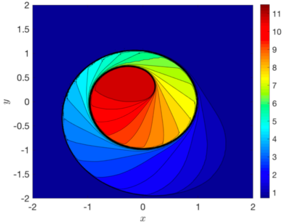
2D KPP equations: Rotating wave

$$\boldsymbol{f}(u) = (\sin u, \cos u), \quad m = 4$$

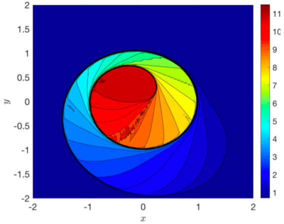


2D KPP equations: Rotating wave

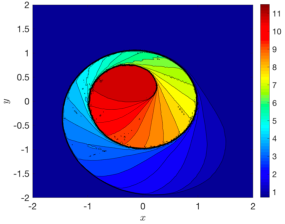
$$\boldsymbol{f}(u) = (\sin u, \cos u), \quad m = 4$$



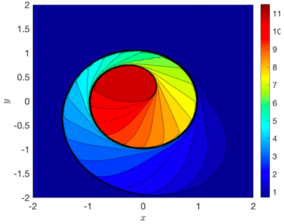
EV



MDH



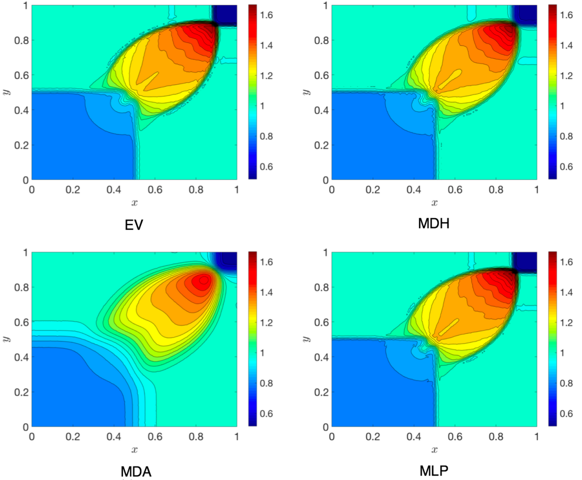
MDA



MLP

2D Euler equations: Riemann Problem config. 12

$m = 3$



Global spectral schemes

Spectral methods for conservation laws

Consider the conservation law (periodic)

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad x \in [0, 2\pi]$$

Define N collocation points (uniform mesh) $x_j = \frac{2\pi}{N} j, \;\; j = 0, ...N-1$.

Using DFT, approximate the solution using the interpolant

$$u(x,t) \approx u_h(x,t) = \frac{1}{2\pi} \sum_{k=-N/2}^{N/2} \hat{u}_k(t) e^{ikx}, \quad \hat{u}_k(t) = \frac{2\pi}{N} \sum_{j=0}^{N-1} u_h(x_j, t) e^{-ikx_j}$$

Solve for the coefficients

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}_h(t)}{\mathrm{d}t} + \boldsymbol{D}\boldsymbol{f}_h(t) &= 0 \\ \boldsymbol{u}_h(t) &= [u_h(x_0,t), ..., u_h(x_{N-1},t)]^\top \\ \boldsymbol{f}_h(t) &= [f(u_h(x_0,t)), ..., f(u_h(x_{N-1},t))]^\top \end{aligned}$$

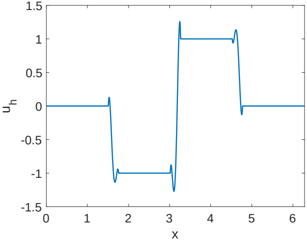
Spectral methods for conservation laws

<u>DG scheme</u>

- Local basis in each element.
- For smooth solutions

$$\textit{Error} \sim \mathcal{O}(h^{m+1/2})$$

- Local Gibbs oscillations

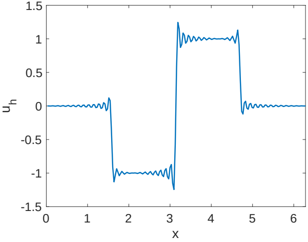


<u>Fourier spectral scheme</u>

- Global basis.
- For smooth solutions

$$\textit{Error} \sim \mathcal{O}(h^p) \ \forall \ p \geq 0$$

- Global Gibbs oscillations



Handling Gibbs oscillations: filtering

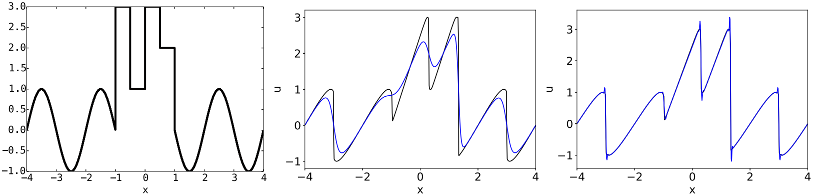
Define the filter of order p

$$\sigma(x) = \begin{cases} e^{(-\beta x^p)} & \text{if } 0 \leq x \leq 1, \\ 0, & \text{if } x > 1 \end{cases} \qquad (\text{fix } \beta = 10)$$

Modify the Fourier coefficients as

$$\hat{u}_k \to \sigma_p\left(\frac{|k|}{N/2}\right) \quad \forall -\frac{N}{2} \leq k \leq \frac{N}{2}.$$

For example, solving the Burgers equation (N=600)



(a) Init. cond. (b) p=2 (c) p=4

Handling Gibbs oscillations: artificial viscosity

Modify scheme as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{u}_h(t)}{\mathrm{d}t} + \boldsymbol{D}\boldsymbol{f}_h(t) &= \boldsymbol{D}(\boldsymbol{\mu}_h(t) \odot \boldsymbol{D}\boldsymbol{u}_h(t)) \\ \boldsymbol{\mu}_h(t) &= [\mu_h(x_0, t), ..., \mu_h(x_{N-1}, t)]^\top \end{aligned}$$

Issue: Determining $\mu_h(x_j, t)$ in the absence of local regularity estimates.

One approach: Estimate $\mu_h(x_j, t)$ based on entropy production (EV)

- Existence of an entropy pair
- Parameter tuning

Handling Gibbs oscillations: artificial viscosity + filtering

Proposed approach:

1. Train a network to classify the "local" regularity τ_j on a stencil S_j^7

$$\tau_j = \begin{cases} 1 & \text{if } u \text{ is discontinuous} \\ 2 & \text{if } u \in C^0 \setminus C^1 \\ 3 & \text{if } u \in C^1 \setminus C^2 \\ 4 & \text{if } u \in C^2 \end{cases}$$

2. Apply a high order exponential filter of order $p = 14$
3. Estimate artificial viscosity (inspired by MDA)

$$\mu_j = \mathcal{Q}(\tau_j) h \max_{x \in \mathcal{S}_7^r} |f'(u)|, \quad \mathcal{Q}(\tau_j) = \begin{cases} 0.5, & \text{if } \tau_j = 1, \\ 0.25, & \text{if } \tau_j = 2, \\ 0.0, & \text{if } \tau_j \geq 3 \end{cases}$$

Handling Gibbs oscillations: artificial viscosity + filtering

Proposed approach:

Input stencil S_j^7 with 7 points.

Training data sampled from the following periodic functions on $[0, 2\pi]$

$$f_1(x) = \begin{cases} a_1 & \text{if } |x - \pi| \leq a_3, \\ a_2 & \text{if } |x - \pi| > a_3, \end{cases}$$

$$f_2(x) = \begin{cases} a_1|x - \pi| - a_1 a_3 & \text{if } |x - \pi| \leq a_3, \\ a_2|x - \pi| - a_2 a_3 & \text{if } |x - \pi| > a_3, \end{cases}$$

$$f_3(x) = \begin{cases} 0.5a_1|x - \pi|^2 - a_1 a_3 & \text{if } |x - \pi| \leq a_3, \\ a_2|x - \pi|^2 - a_2 a_3 - 0.5a_3^2(a_1 - a_2) & \text{if } |x - \pi| > a_3 \end{cases}$$

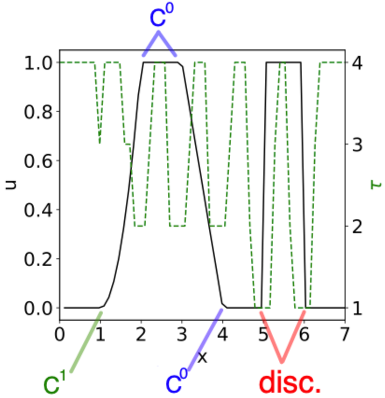
$$f_4(x) = \sin(2xa)$$

We don't use solutions to conservation laws!

Handling Gibbs oscillations: artificial viscosity + filtering

Proposed approach:

- Input $\boldsymbol{X} \in \mathbb{R}^7$.
- Output $\boldsymbol{Y} \in \mathbb{R}^4$ – class probability.
- 3 hidden layers with ELU activation function.



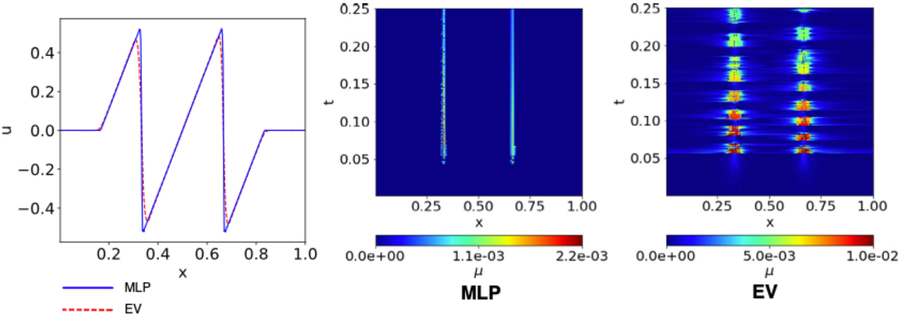
Numerical setup

- Same network for all problems.
- Time-marching using SSPRK-4.
- In multi-D, network applied in a dimension-by-dimension manner + nodal minimum.

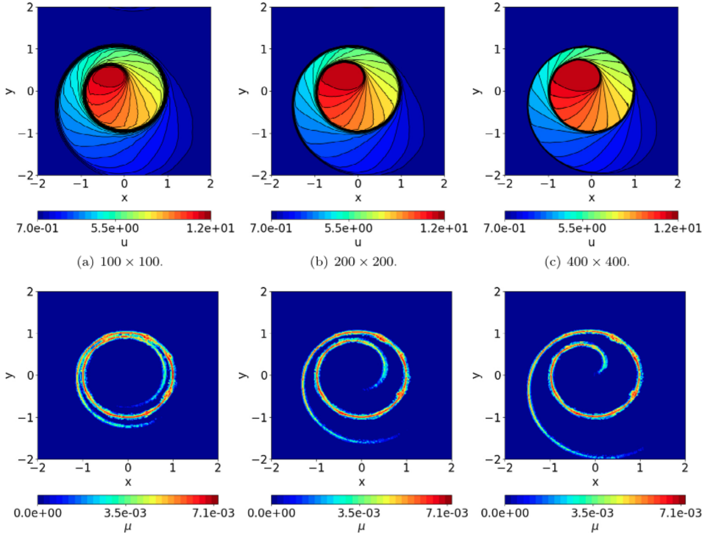
Controlling oscillations in spectral methods by local artificial viscosity governed by neural networks, by Schwander, Hesthaven and R.; JCP, 2021.

Burgers equation: EV vs. MLP

$$u_0(x) = \begin{cases} -\sin(6\pi x) & \text{if } \frac{1}{6} \leq x \leq \frac{5}{6}, \\ 0 & \text{otherwise} \end{cases}$$



2D KPP equations: Rotating wave



(a) 100×100.

(b) 200×200.

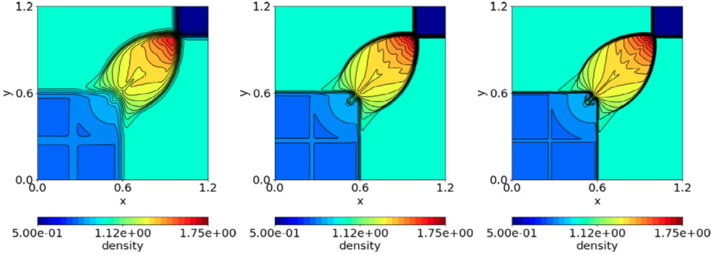
(c) 400×400.

(d) 400×400, $t = 0.2$

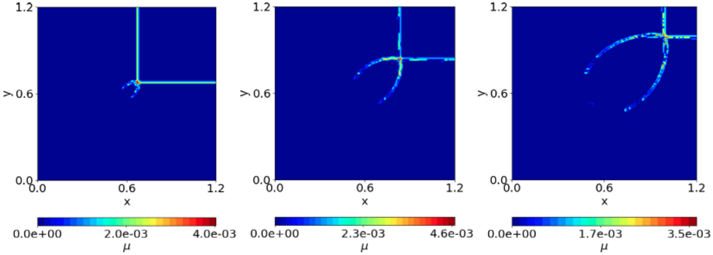
(e) 400×400, $t = 0.6$

(f) 400×400, $t = 1$

2D Euler equations: : Riemann Problem config. 12 (ρ based)



(a) 100×100. (b) 200×200. (c) 300×300.



(d) $300 \times 300,\ t = 0.05$ (e) $300 \times 300,\ t = 0.15$ (f) $300 \times 300,\ t = 0.25$

Conclusion

- Neural networks are useful for shock-capturing.
  - Estimate viscosity
  - Detect troubled-cells (R. and Hesthaven, 2019)
- Deep learning can be used to enhance numerical algorithms.
- Neural networks are great at detecting patterns.
- Domain knowledge is valuable in constructing datasets.

Future directions

Sub-cell resolution

- MLP for DG schemes trained on one-μ-per-element data.
- Generate training data using methods estimating high-order sub-cell artificial viscosity (Zeifang and Beck, 2021).

Future directions

Sub-cell resolution

- MLP for DG schemes trained on one-μ-per-element data.
- Generate training data using methods estimating high-order sub-cell artificial viscosity (Zeifang and Beck, 2021).

Boundary conditions for spectral methods

- Fourier spectral methods – domain doubling for periodicity.
- MLP works on non-periodic problems on Chebyshev grids – time-step restriction.
- Fourier continuation using Gram polynomials (Bruno, CalTech).

Future directions

Reduced order modelling

- ROMs for conservation laws also suffer from spurious oscliiations.
- Use an MLP-based viscosity model to stabilize (Yu, Beihang University)

Future directions

Reduced order modelling

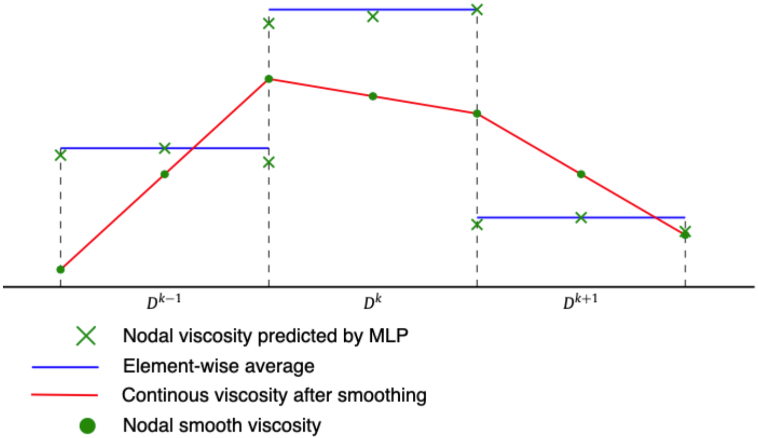
- ROMs for conservation laws also suffer from spurious osclliations.
- Use an MLP-based viscosity model to stabilize (Yu, Beihang University)

hp-adaptation

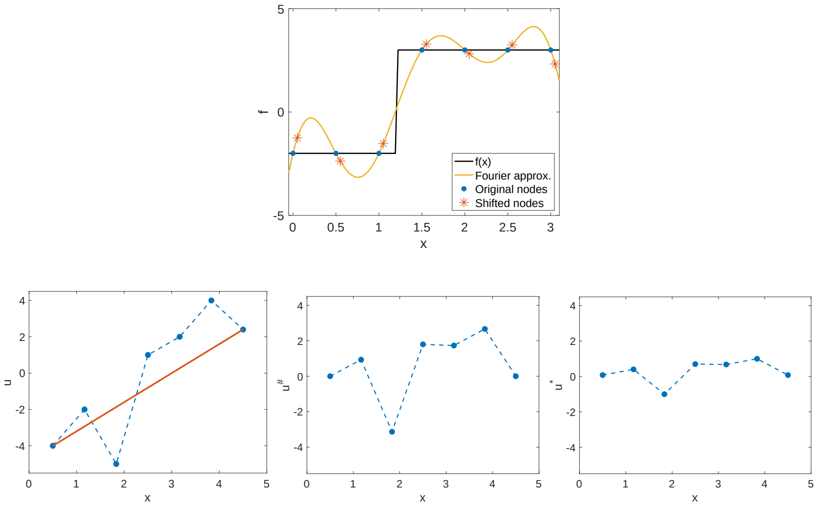
- Demonstrated that MLPs can classify local smoothness.
- Train networks to classify "best" local basis order in DG schemes (Wang and R.)

Questions?

DG viscosity smoothing

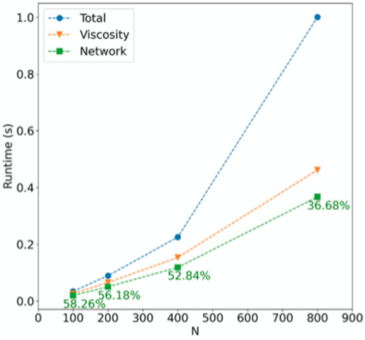


SVANN sampling and scaling

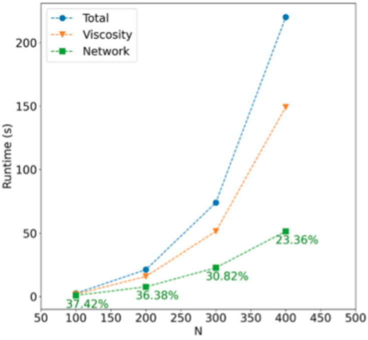


(a) Find line (b) Rotate (c) Scale

SVANN: Computational cost



(a) 1D Burgers' problem



(b) 2D KPP problem