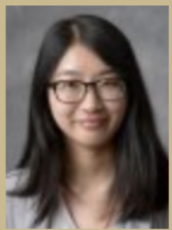
Least-Squares Neural Network (LSNN) Method for Scalar Hyperbolic Conservation laws (HCLs)

Zhiqiang Cai[1]



Jingshuang Chen

Min Liu

PURDUE UNIVERSITY

Department of Mathematics[1]

School of Mechanical Engineering [2]

Outline

- Scalar Hyperbolic Conservation Laws (HCLs)
- Neural Networks
- Least-Squares Neural Network (LSNN) Method

https://www.math.purdue.edu/~caiz/paper.html

Scalar Hyperbolic Conservation Laws

- **Scalar Nonlinear Hyperbolic Conservation Laws**

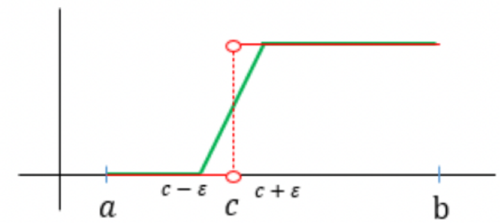
$$
\begin{cases}
u_t(\mathbf{x},t) + \nabla_{\mathbf{x}} \cdot \mathbf{f}(u) & = & 0, & \text{in} \quad \Omega \times I, \\
u & = & g, & \text{on} \quad \Gamma_-, \\
u(\mathbf{x},0) & = & u_0(\mathbf{x}), & \text{in} \quad \Omega,
\end{cases}
$$

- **Numerical Difficulties**
  - Issues on mathematical theory of PDE
  - Solutions are discontinuous without a priori knowledge of locations

Approximation to Unit Step Function with Unknown Interface

- **Unit step function with unknow interface c**

$$f(x) = \begin{cases} 0, & x \in (a, c), \\ 1, & x \in [c, b). \end{cases}$$



- **Best continuous piece-wise linear approximation**

$$p(x) = \begin{cases} 0, & x \in (a, c-\varepsilon), \\ \frac{x-(c-\epsilon)}{2\varepsilon}, & x \in [c-\varepsilon, c+\varepsilon], \\ 1, & x \in (c+\varepsilon, b). \end{cases}$$

- **Error estimate**

$$\|f-p\|_{L^\infty(I)} = \frac{1}{2} \quad \text{and} \quad \|f-p\|_{L^p(I)} = \frac{\varepsilon^{1/p}}{2^{1-1/p}(1+p)^{1/p}}.$$

Approximation to Unit Step Function with Unknown Interface

- Unit step function and its best CPL approximation

$$f(x) = \begin{cases} 0, & x \in (a, c), \\ 1, & x \in [c, b). \end{cases} \qquad p(x) = \begin{cases} 0, & x \in (a, c-\varepsilon), \\ \frac{x-(c-\epsilon)}{2\varepsilon}, & x \in [c-\varepsilon, c+\varepsilon], \\ 1, & x \in (c+\varepsilon, b). \end{cases}$$

- Error estimate

$$\|f-p\|_{L^\infty(I)} = \frac{1}{2} \quad \text{and} \quad \|f-p\|_{L^p(I)} = \frac{\varepsilon^{1/p}}{2^{1-1/p}(1+p)^{1/p}}.$$

- CPL approximations on **fixed** quasi-uniform mesh
  - very fine mesh-size h=ε
  - overshooting and oscillation
- What is a **right** class of approximating functions?

Neural Network (NN)

Fully-connected (Multi-Layer Perceptron) NN (Rosenblatt 1958)

- **DNN function (models)**

$$\mathbf{y} = N(\mathbf{x}) = N^{(L)} \circ \cdots \circ N^{(2)} \circ N^{(1)}(\mathbf{x}) : \mathcal{R}^d \longrightarrow \mathcal{R}^o$$

$$N^{(l)}(\mathbf{x}^{(l-1)}) = \sigma(\boldsymbol{\omega}^{(l)}\mathbf{x}^{(l-1)} - \mathbf{b}^{(l)}) : \mathcal{R}^{n_{l-1}} \longrightarrow \mathcal{R}^{n_l}$$

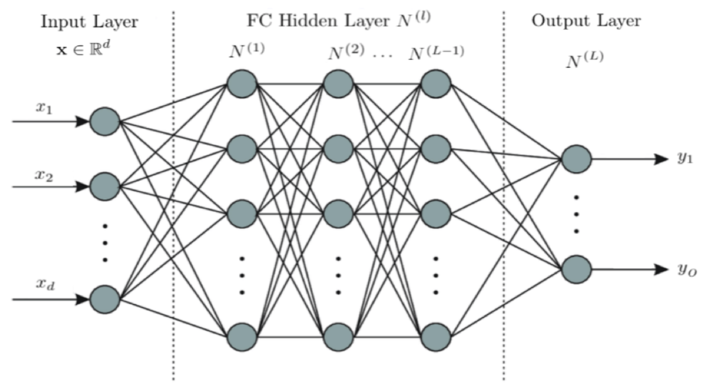
- **The number of parameters**

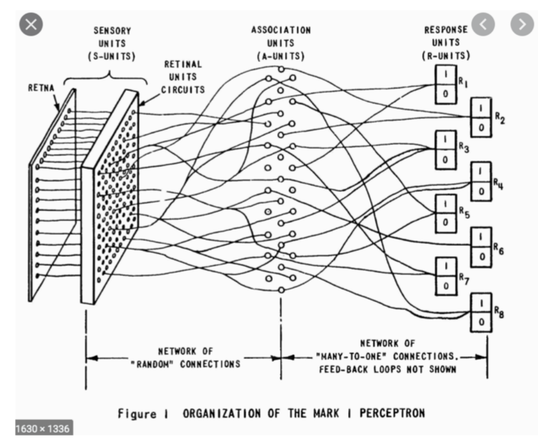
$$N = \sum_{l=1}^{L} n_l \times (n_{l-1} + 1)$$

- **Activate functions**
  - ReLU^k
  - Sigmoids
  - Etc.

$$\sigma(t) = \begin{cases} t^k, & t > 0 \\ 0, & t < 0 \end{cases}$$

$$\sigma(t) = \frac{1}{1 + e^x}$$





Approximation Property of NN

- **Universal Approximation Theorem** (Cybenko (1989), Hornik-Stinchcombe-White (1989))

  $\mathcal{M}(\sigma, d) = \{v(\mathbf{x}) \in \mathcal{M}_n(\sigma, d) : n \in \mathbb{Z}_+\}$ is dense in $C(K)$ for any compact set $K \in \mathcal{R}^d$, provided that σ is not a polynomial.

- **A Priori Error Estimate** (DeVore-Oskolkov-Petrushev (1997), DeVore-Hanin-Petrova, Yarosky, Shen-Yang-Zhang, E-Wojtowytsch, Siegle-Xu, …)

  - Why using NN instead of polynomials, finite elements, …?
  - How to design NN architecture?
  - Why using more than two layers?
  - ……

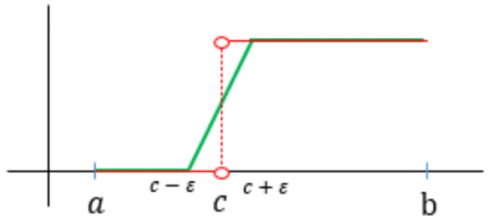
- **Two-layer NN**

$$\mathcal{M}_n(\sigma, d) = \left\{ \mathbf{c}_0 + \sum_{i=1}^{n} \mathbf{c}_i \sigma(\boldsymbol{\omega}_i \cdot \mathbf{x} - b_i) \,:\, \mathbf{c}_i \in \mathcal{R}^o,\, b_i \in \mathcal{R},\, \boldsymbol{\omega}_i \in \mathcal{S}^{d-1} \right\},$$

Approximation to Unit Step Function with Unknown Interface

- **Unit step function with unknow interface c**

$$f(x) = \begin{cases} 0, & x \in (a, c), \\ 1, & x \in [c, b). \end{cases}$$



- **Best 2-layer neural network approximation**

$$p(x) = \frac{1}{b_2 - b_1}\left\{\sigma\big(x-b_1\big) - \sigma\big(x-b_2\big)\right\}, \;\; b_1 = c - \varepsilon, \;\; b_2 = c + \varepsilon$$

!!! One-hidden layer with two neurons !!!

- **Error estimate**

$$\|f-p\|_{L^\infty(I)} = \frac{1}{2} \quad \text{and} \quad \|f-p\|_{L^p(I)} = \frac{\varepsilon^{1/p}}{2^{1-1/p}(1+p)^{1/p}}.$$

LS formulation for linear advection-reaction problem

- **Linear advection-reaction problem**

$$u_{\boldsymbol{\beta}} + \gamma\, u = f \text{ in } \Omega, \qquad u\big|_{\Gamma_-} = g$$

- **Least-squares formulation** Find $u \in V_{\boldsymbol{\beta}}(\Omega) = \{v \in L^2(\Omega) : v_{\boldsymbol{\beta}} \in L^2(\Omega)\}$ such that

$$\mathcal{L}(u; \mathbf{f}) = \min_{v \in V_{\boldsymbol{\beta}}} \mathcal{L}(v; \mathbf{f})$$

where $\mathcal{L}(v; \mathbf{f}) = \|v_{\boldsymbol{\beta}} + \gamma\, v - f\|^2_{0,\Omega} + \|v - g\|^2_{-\boldsymbol{\beta}}$

- **Coercivity and continuity** there exists positive constants α and M such that

$$\alpha\, |||v|||^2_{\boldsymbol{\beta}} \le \mathcal{L}(v; \mathbf{0}) \le M\, |||v|||^2_{\boldsymbol{\beta}}$$

De Sterck-Manteuffel-McCormick-Olson, 2004, Bochev-Gunzburger, 2016

Least-squares neural network (LSNN) method

- **LSNN method** find $u_N \in \mathcal{M}(d,n)$ such that

$$\mathcal{L}(u_N, \mathbf{f}) = \min_{v \in \mathcal{M}(d,n)} \mathcal{L}(v, \mathbf{f})$$

where $\mathcal{M}(d, 1, \lceil \log_2(d+1) \rceil + 1, n) = \mathcal{M}(d,n)$

- **Quasi-optimal approximation**

$$|||u - u_N|||_{\boldsymbol{\beta}} \le \left(\frac{M}{\alpha}\right)^{1/2} \inf_{v \in \mathcal{M}(d,n)} |||u - v|||_{\boldsymbol{\beta}}.$$

- **A priori error estimate**

$$|||u - u_N|||_{\boldsymbol{\beta}} \le C \left(\left|\alpha_1 - \alpha_2\right| \sqrt{\varepsilon} + \inf_{v \in \mathcal{M}(d,n)} |||\hat{u} + p - v|||_{\boldsymbol{\beta}} \right)$$

C.-Chen-Liu, JCP, 443 (2021), 110514; C.-Choi-Liu (2022)

LSNN Method

Numerical Issues

- Integration

  random sampling vs numerical integration

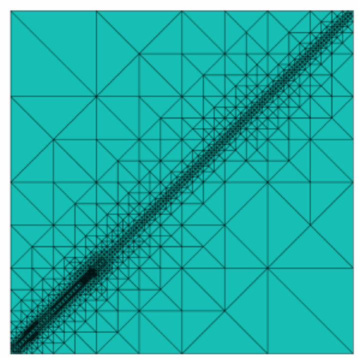
- Differentiation

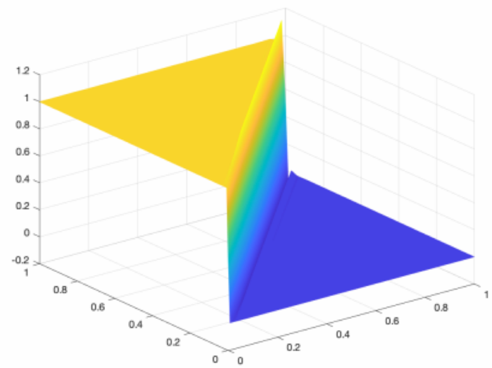
  automatic, exact, numerical, etc

- Algebraic solver (training NN)

  methods of gradient descent, ..., ???

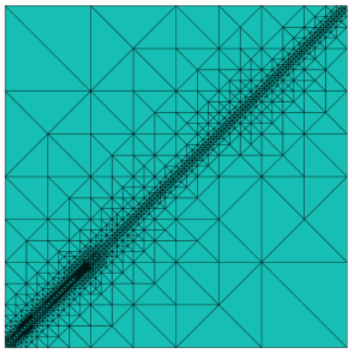
Linear scalar HCL: $f(u)=au$, i.e., $u_t+au_x=0$

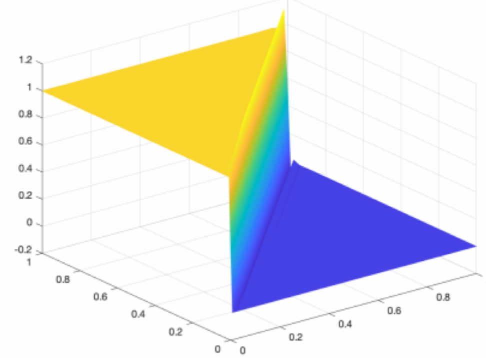


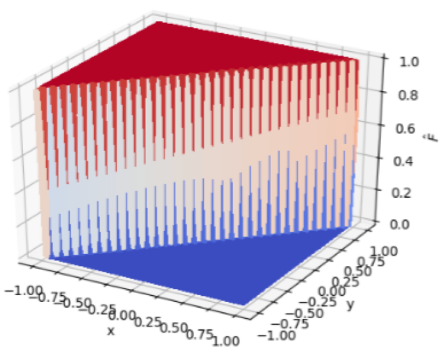


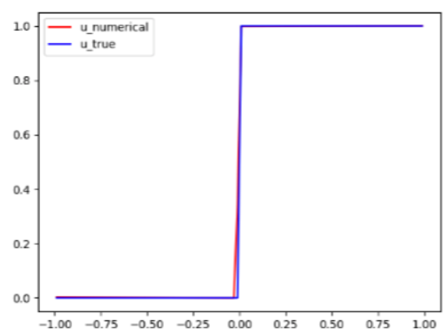
Liu-Zhang, CMAME, 2020

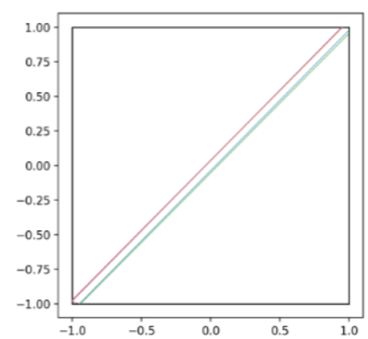
Famous Transport Equation $u_t + u_x = 0$







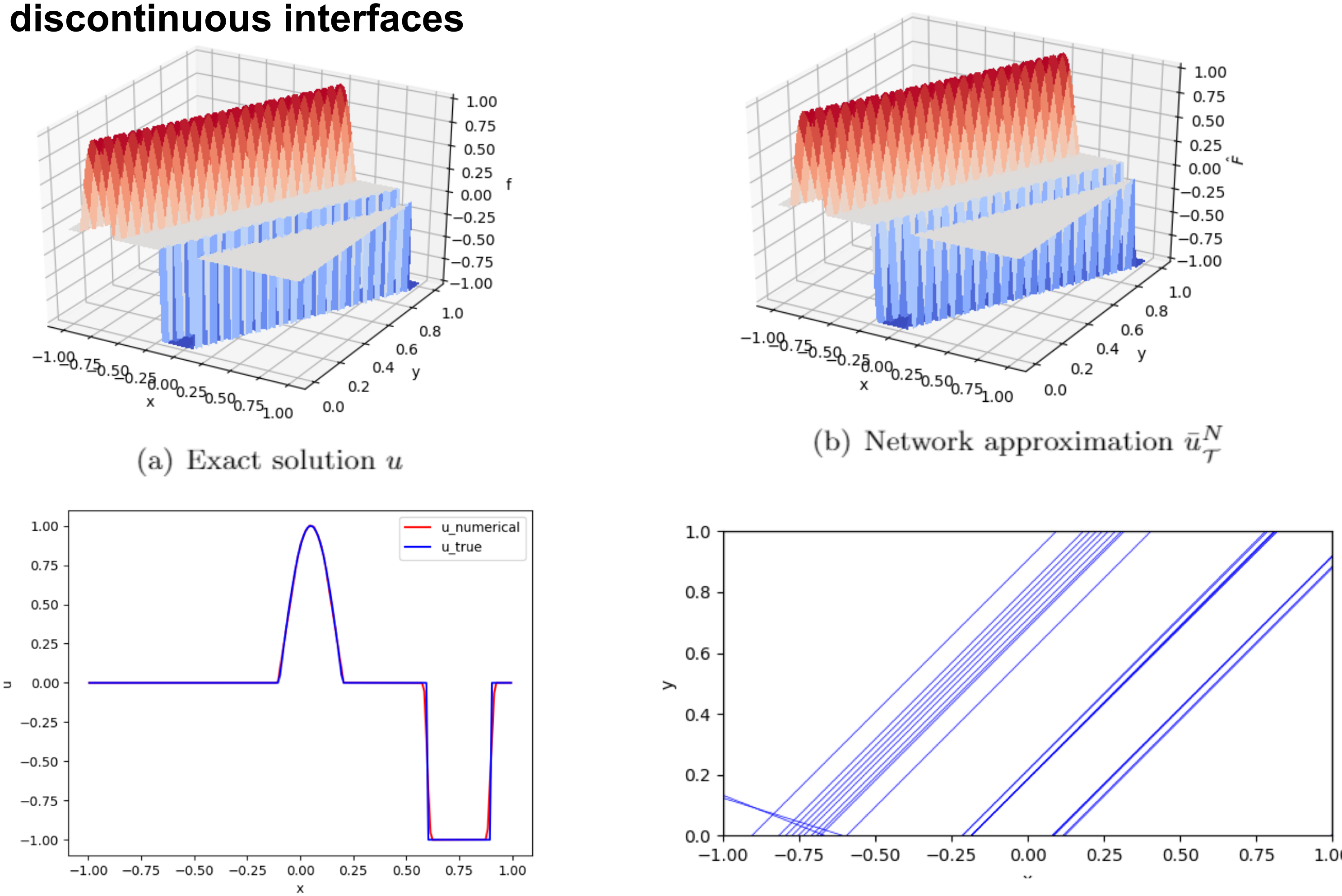




(2-6-1)

C.-Chen-Liu, JCP, 2021

Two discontinuous interfaces



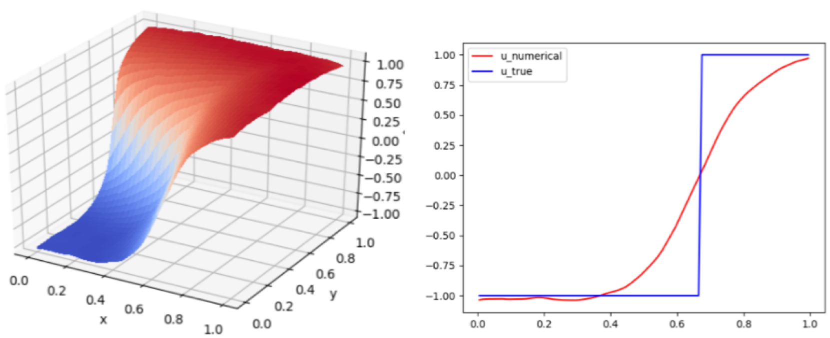
(a) Exact solution u

(b) Network approximation $\bar{u}_{\mathcal{T}}^N$

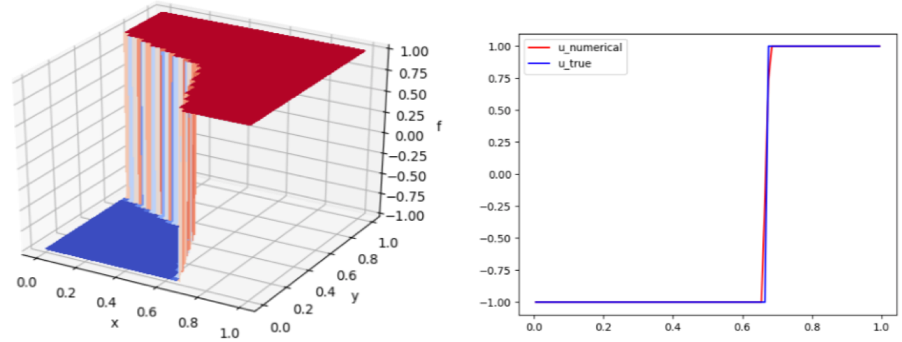
(c) Traces of the exact solution and approximation $\bar{u}_{\mathcal{T}}^N$ on the plane $y = 0.8$

(d) Network breaking lines

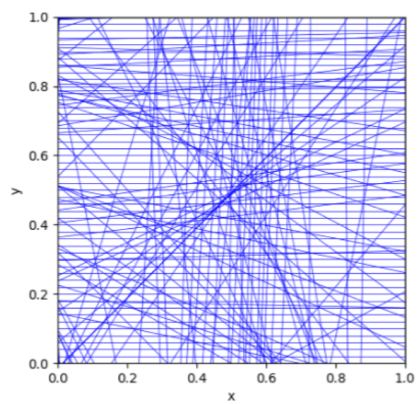
(2-31-1)



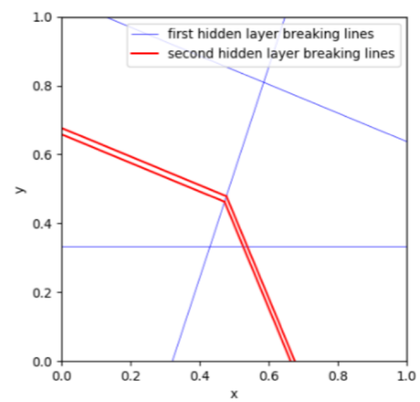
(2-200-1)



(2-5-5-1)



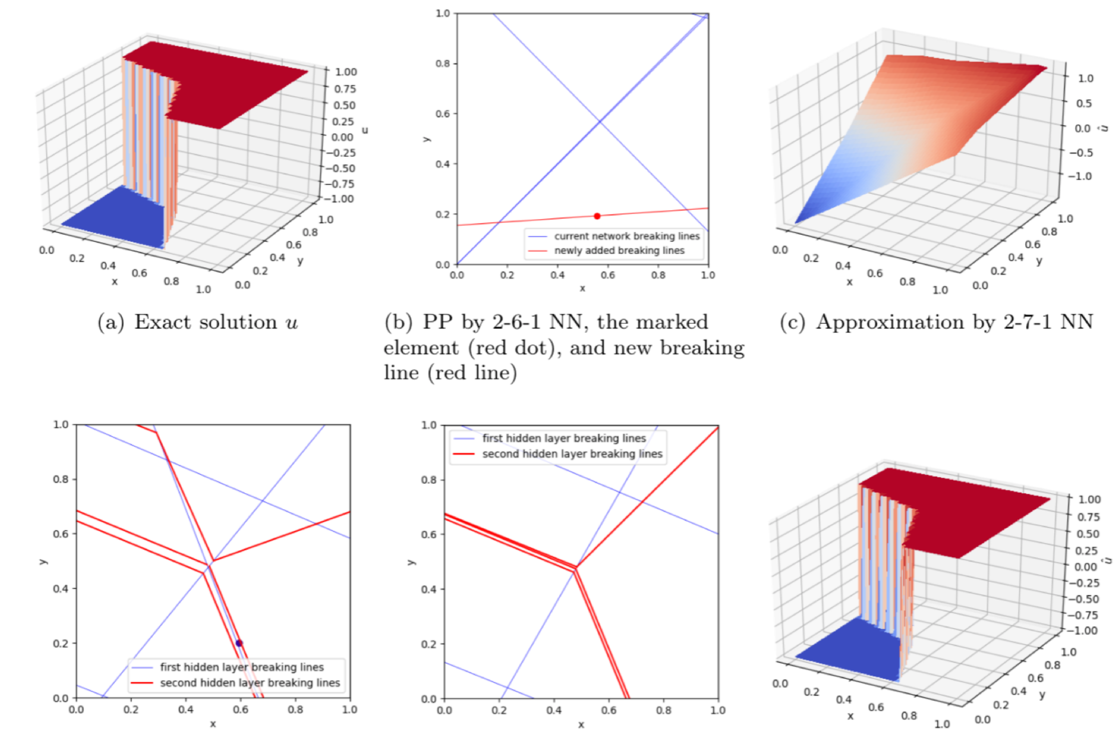
(e) 2-layer NN breaking lines



(f) 3-layer NN breaking lines

C.-Chen-Liu, LSNN method for linear advection-reaction equation, JCP, 443(2021), 110514.

Adaptive 3-Layer NN for Linear Advection-Reaction Problem



(a) Exact solution u (b) PP by 2-6-1 NN, the marked element (red dot), and new breaking line (red line) (c) Approximation by 2-7-1 NN

(d) PP by 2-7-3-1 NN and the marked element (e) PP by adaptive 2-7-4-1 NN (f) Approximation using adaptive 2-7-4-1 NN

Liu-C.-Chen, CAMWA, 2022
Liu-C., CAMWA, 2022,
C.-Chen-Liu, JCP, 2022.

LSNN method for scalar nonlinear HCLs

- **Scalar nonlinear hyperbolic conservation laws**

$$u_t(\mathbf{x},t) + \nabla_{\mathbf{x}} \cdot \mathbf{f}(u) = 0, \text{ in } \Omega \times I, \quad u\big|_{\Gamma_-} = g, \quad u(\mathbf{x},0)\big|_{\Omega} = u_0(\mathbf{x})$$

- **Least-squares formulation**

Find $u \in V_{\mathbf{f}} = \left\{v \in L^2(\Omega \times I) \,|\, (\mathbf{f}(v), v) \in H(\text{div}; \Omega \times I)\right\}$ such that

$$\mathcal{L}(u; \mathbf{g}) = \min_{v \in V_{\mathbf{f}}} \mathcal{L}(v; \mathbf{g})$$

where $\mathcal{L}(v;\mathbf{g}) = \|v_t + \nabla_{\mathbf{x}} \cdot \mathbf{f}(v)\|^2_{0,\Omega\times I} + \|v - g\|^2_{0,\Gamma_-} + \|v(\mathbf{x},0) - u_0(\mathbf{x})\|^2_{0,\Omega}$

- **Well-posedness???**

C.-Chen-Liu, ANM (2022); arXiv:2011.10895v2[math.NA]

LSNN method for scalar nonlinear HCLs

- **Least-squares formulation**

Find $u \in V_{\mathbf{f}} = \left\{ v \in L^2(\Omega \times I) \,|\, (\mathbf{f}(v), v) \in H(\text{div}; \Omega \times I) \right\}$ such that

$$\mathcal{L}\big(u;\,\mathbf{g}\big) = \min_{v \in V_{\mathbf{f}}} \mathcal{L}\big(v;\,\mathbf{g}\big)$$

where $\mathcal{L}(v;\mathbf{g}) = \|v_t + \nabla_{\mathbf{x}} \cdot \mathbf{f}(v)\|^2_{0,\Omega\times I} + \|v - g\|^2_{0,\Gamma_-} + \|v(\mathbf{x},0) - u_0(\mathbf{x})\|^2_{0,\Omega}$

- **LSNN method** finding $u^N(\mathbf{z};\boldsymbol{\theta}^*) \in \mathcal{M}_N$ such that

$$\mathcal{L}\big(u^N(\cdot;\boldsymbol{\theta}^*);g\big) = \min_{v \in \mathcal{M}_N} \mathcal{L}\big(v(\cdot;\boldsymbol{\theta});g\big) = \min_{\boldsymbol{\theta} \in \mathbb{R}^N} \mathcal{L}\big(v(\cdot;\boldsymbol{\theta});g\big)$$

- **Numerical Issues:** **integration, differentiation, …**

C.-Chen-Liu, ANM (2022); arXiv:2011.10895v2[math.NA]

Discrete Divergence Operator

- Divergence operator

$$0 = u_t + \nabla_{\mathbf{x}} \cdot \mathbf{f}(u) = \mathbf{div}\,\big(u, \mathbf{f}(u)\big) = \mathbf{div}\,\mathbf{F}(u)$$

- Discrete divergence operator
  + based on conservative numerical schemes (C.-Chen-Liu (2022), ANM)
  + new discrete divergence operator (C.-Chen-Liu, arXiv:2011.10895v2[math.NA])

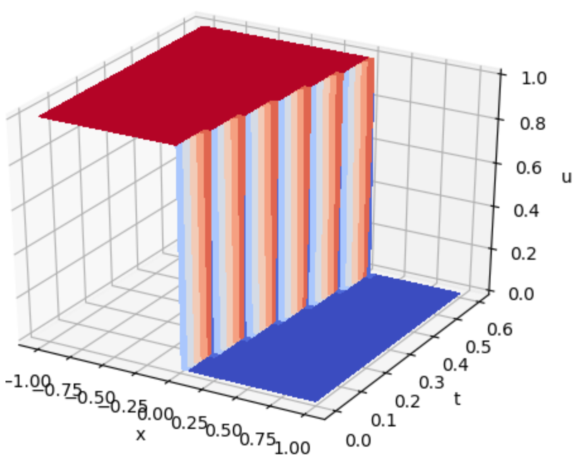
Let $\mathcal{T}$ be a partition of the domain $\Omega \subset \mathbb{R}^{d+1}$.

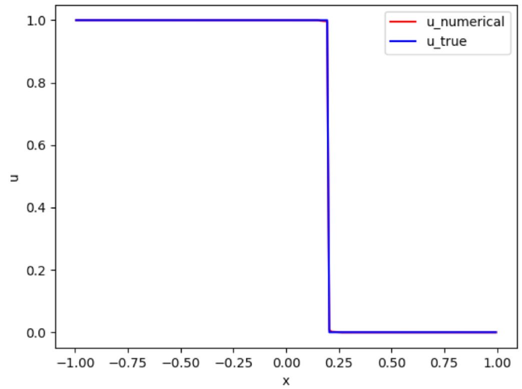
For any $K \in \mathcal{T}$, let $\mathbf{z}_K$ be the centroid of K.

$$\mathbf{div}_{\mathcal{T}}\mathbf{F}\big(u(\mathbf{z}_K)\big) \approx \mathrm{avg}_K \mathbf{div}\,\mathbf{f}(u) = \frac{1}{|K|}\int_{\partial K} \mathbf{F}(u) \cdot \mathbf{n}\, dS$$

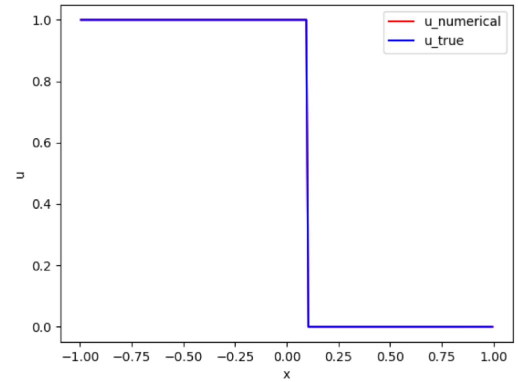
Inviscid Burger Equation $f(u) = \frac{1}{2}u^2$

Riemann Problem Shock formation: exact solution

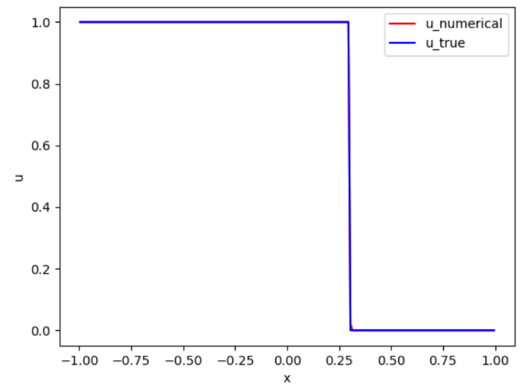




t=0.2



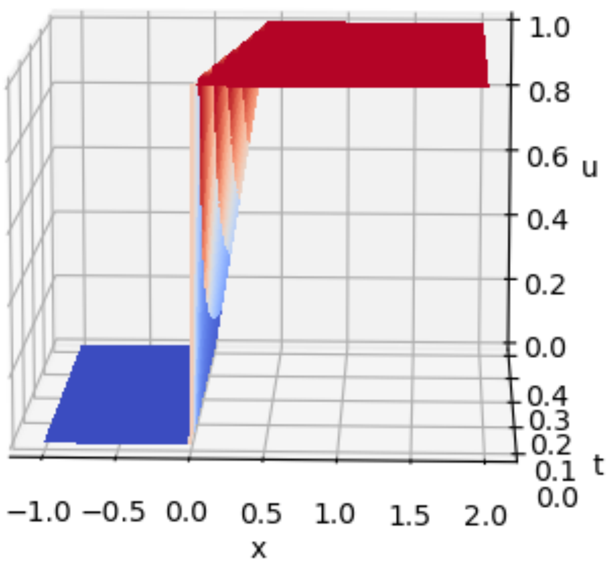
t=0.4

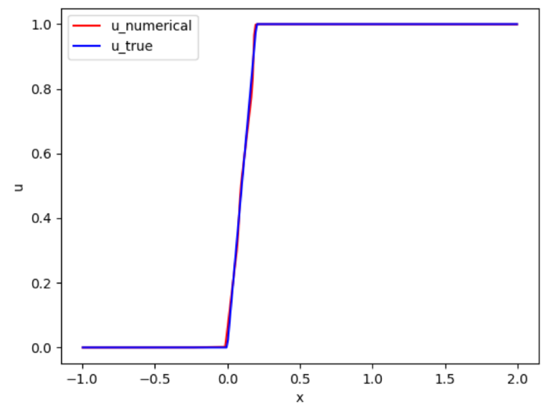


t=0.6

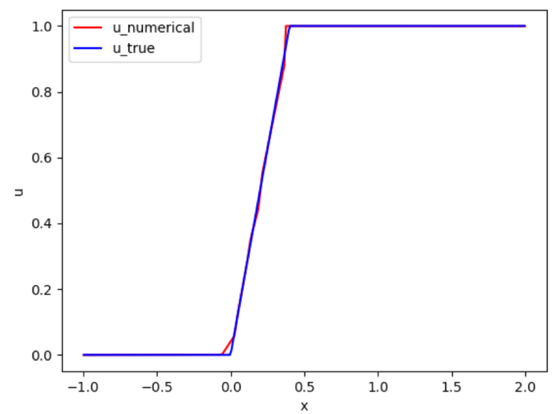
(2-10-10-1)

Riemann Problem
Rarefaction wave: exact solution





t=0.2

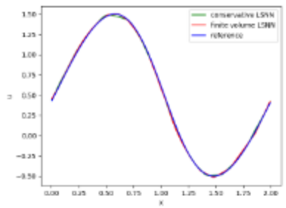


t=0.4

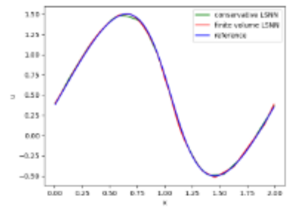
(2-10-10-1)

Inviscid Burgers equation with smooth initial

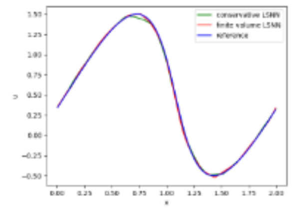
$$u_0(x) = 0.5 + \sin(\pi x).$$



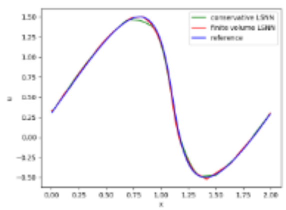
(a) Traces of reference and numerical solutions $u_{1,\mathcal{T}}$ on the plane $t = 0.05$



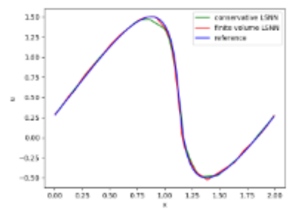
(b) Traces of reference and numerical solutions $u_{2,\mathcal{T}}$ on the plane $t = 0.1$



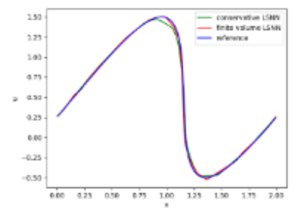
(c) Traces of reference and numerical solutions $u_{3,\mathcal{T}}$ on the plane $t = 0.15$



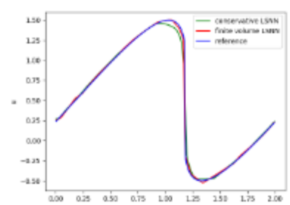
(d) Traces of reference and numerical solutions $u_{4,\mathcal{T}}$ on the plane $t = 0.2$



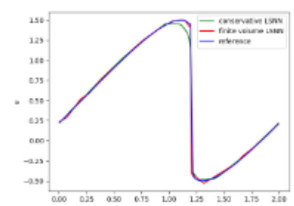
(e) Traces of reference and numerical solutions $u_{5,\mathcal{T}}$ on the plane $t = 0.25$



(f) Traces of reference and numerical solutions $u_{6,\mathcal{T}}$ on the plane $t = 0.3$



(g) Traces of reference and numerical solutions $u_{7,\mathcal{T}}$ on the plane $t = 0.35$

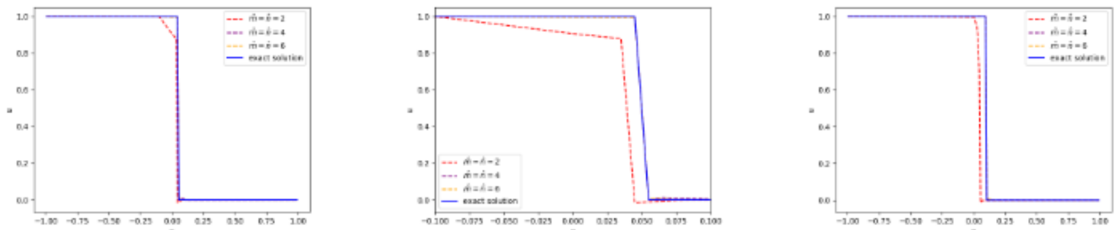


(h) Traces of reference and numerical solutions $u_{8,\mathcal{T}}$ on the plane $t = 0.4$

(2-30-30-1)

FIG. 3. *Approximation results of Burgers' equation with a sinusoidal initial condition*

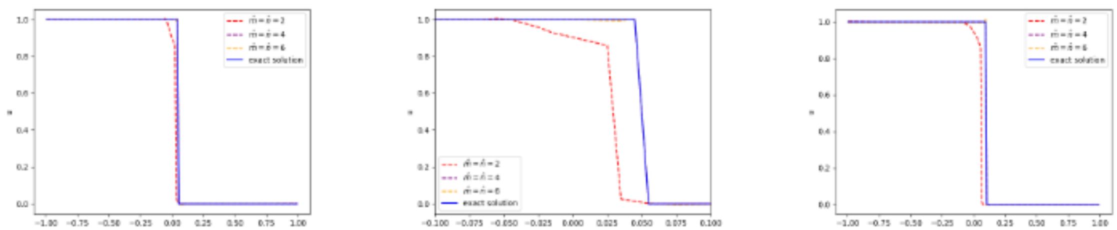
Riemann Problem with Higher order flux $f(u) = \frac{1}{4}u^4$



(a) Traces of exact and numerical solutions $u_{1,\mathcal{T}}$ using the trapezoidal rule on the plane $t = 0.2$

(b) Zoom-in plot near the discontinuous interface of sub-figure (a)

(c) Traces of exact and numerical solutions $u_{2,\mathcal{T}}$ using the trapezoidal rule on the plane $t = 0.4$



(d) Traces of exact and numerical solutions $u_{1,\mathcal{T}}$ using the mid-point rule on the plane $t = 0.2$

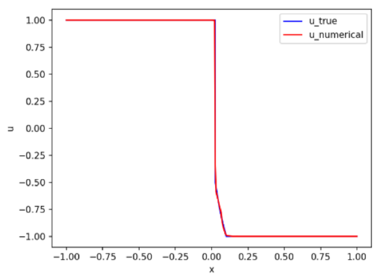
(e) Zoom-in plot near the discontinuous interface of sub-figure (d)

(f) Traces of exact and numerical solutions $u_{2,\mathcal{T}}$ using the mid-point rule on the plane $t = 0.4$

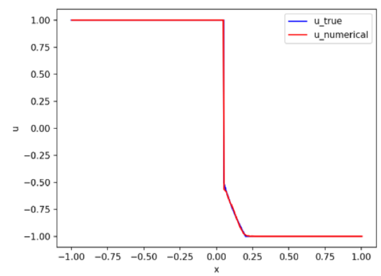
(2-10-10-1)

FIG. 5. *Numerical results of the problem with $f(u) = \frac{1}{4}u^4$ using the composite trapezoidal and mid-point rules*

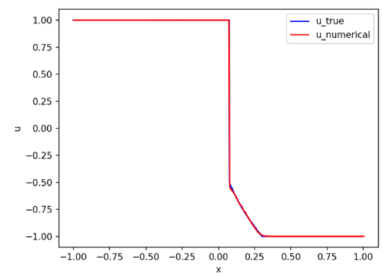
Riemann Problem with Non-convex flux $f(u) = \frac{1}{3}u^3$



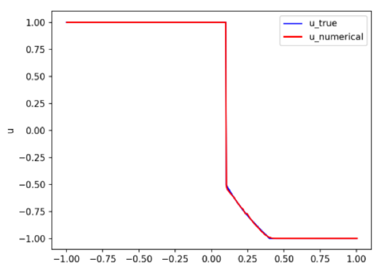
(a) Traces at $t = 0.1$



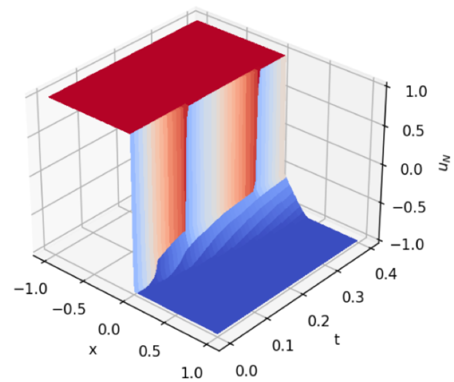
(b) Traces at $t = 0.2$



(c) Traces at $t = 0.3$



(d) Traces at $t = 0.4$

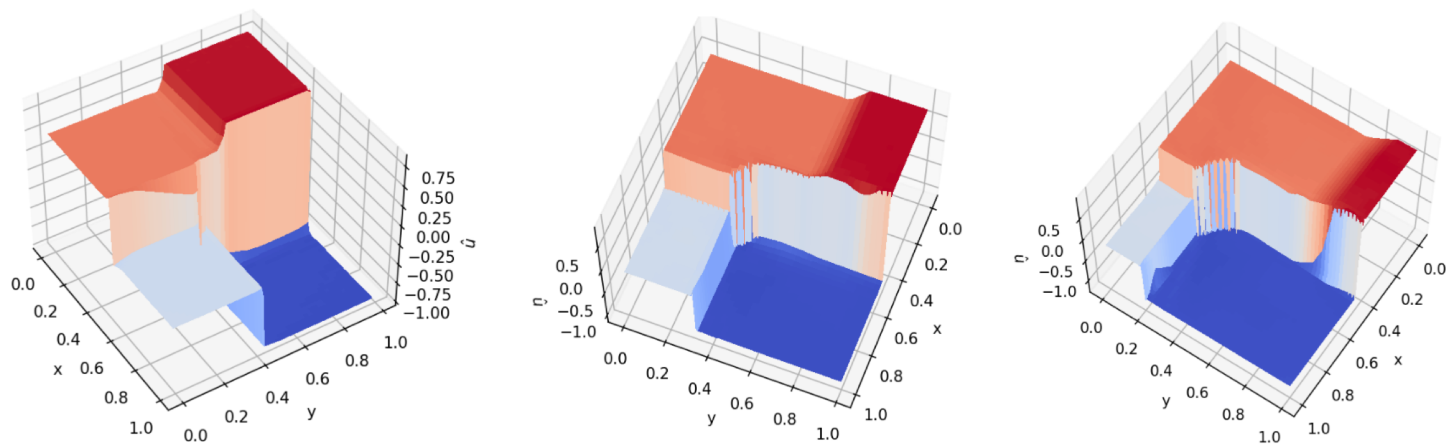


(e) Numerical Solution u_N on Ω

2-64-64-64-1

2D Inviscid Burger Equation $f(u) = \frac{1}{2}(u^2, u^2)$

Network structure	Block	$\frac{\|u^k - u^k_{\mathcal{T}}\|_0}{\|u^k\|_0}$
3-48-48-48-1	$\Omega_{0,1}$	0.093679
	$\Omega_{1,2}$	0.121375
	$\Omega_{2,3}$	0.163755
	$\Omega_{3,4}$	0.190460
	$\Omega_{4,5}$	0.213013



(a) $t = 0.1$ (b) $t = 0.3$ (c) $t = 0.5$

Summary

- **NN provides a new class of approximating functions**

  moving mesh vs uniform mesh and adaptive mesh

- **Scalar hyperbolic conservation laws**

  NN is possibly the best class of approximating functions for scalar HCLs.

- **Non-convex optimization**

  Bottleneck, the method of continuation, ...

- **Adaptive Neural Network**
  - Automatically design a relatively small NN within the prescribed tolerance
  - A natural continuation process for obtaining a good initial

THANK YOU