Finite Element Circus
University of Norte Dame
October 20-21, 2023

NEURAL NETWORKs: A COUSIN OF FINITE ELEMENTS

Zhiqiang Cai

Collaborators: Min Liu, Jingshuang Chen, Junpyo Choi, Brooke Hejnal

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

PURDUE UNIVERSITY
Department of Mathematics [1]
School of Mechanical Engineering [2]

C^0 Linear Elements on fixed and moving meshes

- C^0 Linear Element on a **fixed** mesh in [a,b]

$$\mathcal{S}_1^0(\Delta) = \text{span}\,\{\phi_i(x)\}_{i=0}^n = \left\{\sum_{i=0}^n c_i\phi_i(x) : c_i \in \mathcal{R}\right\} \qquad \phi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in (x_{i-1}, x_i), \\ \dfrac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in (x_i, x_{i+1}), \\ 0, & \text{otherwise} \end{cases}$$

- C^0 Linear Element on a **moving** mesh in [a,b]

$$\mathcal{S}_1^0(n) = \left\{\sum_{i=0}^n c_i\phi_i(x; x_{i-1}, x_i, x_{i+1}) : c_i \in \mathcal{R},\ x_i \in [a,b]\right\} \qquad u(x) = x^{0.01},\ x \in [0,1]$$

$$= \left\{c_0 + c_1(x-a) + \sum_{i=2}^n c_i\sigma(x - x_i) : c_i \in \mathcal{R},\ x_i \in (a,b)\right\} \qquad \sigma(t) = \begin{cases} t, & t > 0, \\ 0, & t \le 0 \end{cases}$$

One hidden-layer NN in R ^{d}

- **One hidden-layer NN** (C^0 piecewise linear function)

$$\mathcal{M}_n(d) = \left\{ c_0 + \sum_{i=1}^{n} c_i \sigma(\boldsymbol{\omega}_i \mathbf{x} + b_i) \,:\, c_i, b_i \in \mathcal{R},\ \boldsymbol{\omega}_i \in \mathcal{S}^{d-1} \right\}$$

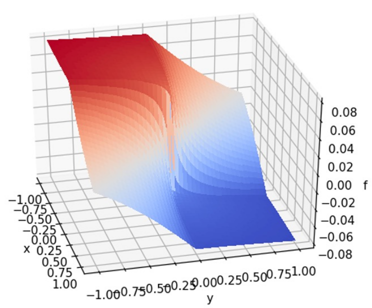
- **Breaking Hyper-Planes**

$$\mathcal{P}_i \,:\, \boldsymbol{\omega}_i \mathbf{x} + b_i = 0 \quad \text{for} \ \ i = 1, \ldots, n$$

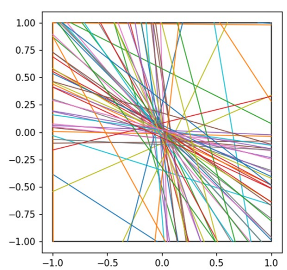
- **Linearly Independence**

$$\{\sigma(\boldsymbol{\omega}_i \mathbf{x} + b_i)\}_{i=1}^{n} \ \text{are linearly independent} \ \{\mathcal{P}_i\}_{i=1}^{n} \ \text{are distinct}$$

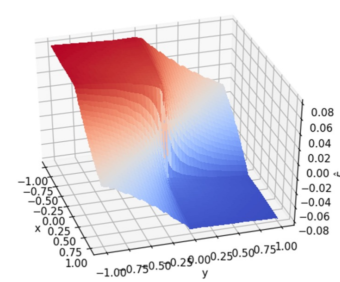
- **Physical Partition of NN approximation to Kellogg function**



(a) Target function f(x,y)



(h) Optimum break lines (69 neurons, 1286 elements)



(i) Optimum NN model of 69 neurons, $\xi = 0.008476$

PURDUE UNIVERSITY

Neural Networks (NNs): a class of new approximating functions

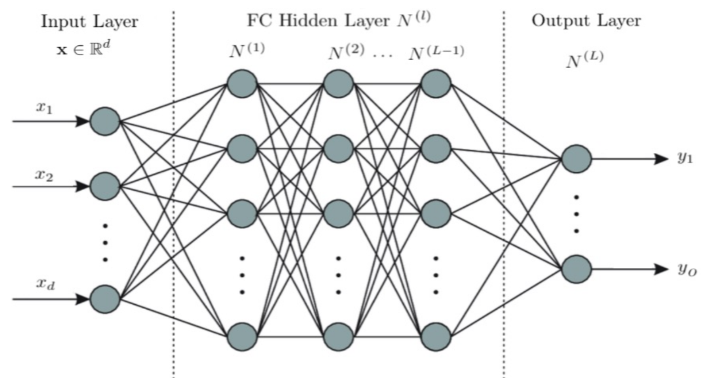
Fully-connected (Multi-Layer Perceptron) NN (Rosenblatt 1958)

- **DNN function (models)**

Let $\mathbf{x}^{(0)} = \mathbf{x}$ and $\mathbf{x}^{(i)} = \sigma\left(W^{(i)}_{n\times(n+1)}\begin{bmatrix}1\\ \mathbf{x}^{(i-1)}\end{bmatrix}\right)$ for $i = 1, \dots, l$

$$u(\mathbf{x}; W) = W^{(l+1)}_{1\times(n+1)}\begin{bmatrix}1\\ \mathbf{x}^{(l)}\end{bmatrix}$$

where $W = \left[W^{(1)}_{n\times(d+1)}, W^{(2)}_{n\times(n+1)}, \dots, W^{(l)}_{n\times(n+1)}, W^{(l+1)}_{1\times(n+1)}\right]$



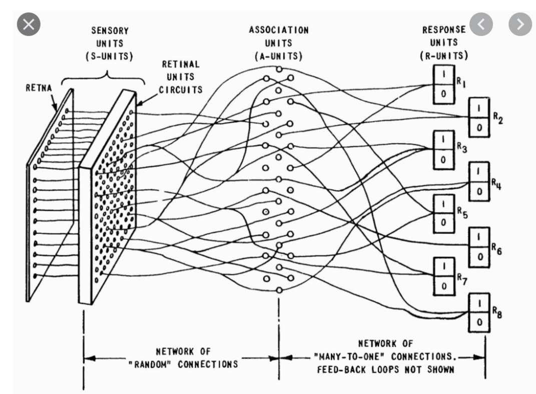


Figure I ORGANIZATION OF THE MARK I PERCEPTRON

- **ReLU Activate function**

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

Scalar Hyperbolic Conservation Laws

- **Scalar Nonlinear Hyperbolic Conservation Laws**

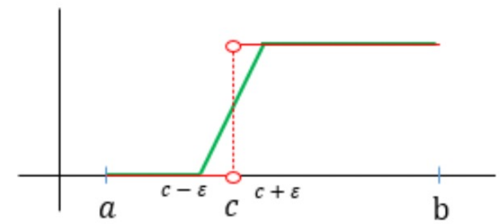
$$\begin{cases} u_t(\mathbf{x},t) + \nabla_{\mathbf{x}} \cdot \mathbf{f}(u) &= 0, & \text{in} \quad \Omega \times I, \\ u &= g, & \text{on} \quad \Gamma_-, \\ u(\mathbf{x},0) &= u_0(\mathbf{x}), & \text{in} \quad \Omega, \end{cases}$$

- **Mathematical and Numerical Difficulties**
    - Mathematical theory of PDE
    - Solutions are discontinuous with unknown locations

Approximation to Unit Step Function with Unknown Interface

- **Unit step function and its CPWL approximation**

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



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

- **How to compute or approximate p(x) when c is unknown?**

(1) On fixed quasi-uniform mesh
- very fine mesh-size: $h = \varepsilon$
- overshooting, oscillation, etc.

(2) On moving mesh (neural network)
- two neurons
- no overshooting or oscillation

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

Approximation to Unit Step Function with Unknown Interface in R^d

- **Piecewise Constant function with unknow interface**

C., J. Choi, and M. Liu (2022) (d=2, 3, L=1; d=4,…,8, L=2)

Let $\chi(x)$ be a piecewise constant function with C^0 piecewise smooth interface I, then there exists a CPWL function p(x) generated by a DNN with L= $\lceil \log_2(d+1) \rceil$ hidden layers such that for any given $\varepsilon > 0$, we have

$$|||\chi - p|||_{\boldsymbol{\beta}} \le \sqrt{2|I|}\,\left|\alpha_1 - \alpha_2\right|\sqrt{\varepsilon},$$

P. Petersen and F. Voigtlaender (2018) (For C^1 and d=2, L=36)

Theorem 3.5. *For $r \in \mathbb{N}$, $d \in \mathbb{N}_{\ge 2}$, and $p, \beta, B > 0$, there are constants $c = c(d, r, p, \beta, B) > 0$ and $s = s(d, r, p, \beta, B) \in \mathbb{N}$, such that for any $K \in \mathcal{K}_{r,\beta,d,B}$ and any $\varepsilon \in (0, 1/2)$, there is a neural network Φ^K_ε with at most $(3 + \lceil \log_2 \beta \rceil) \cdot (11 + 2\beta/d)$ layers, and at most $c \cdot \varepsilon^{-p(d-1)/\beta}$ nonzero, (s, ε)-quantized weights such that*

$$\|\mathrm{R}_\varrho(\Phi^K_\varepsilon) - \chi_K\|_{L^p([-1/2,1/2]^d)} < \varepsilon \quad \text{and} \quad \|\mathrm{R}_\varrho(\Phi^K_\varepsilon)\|_{\sup} \le 1.$$

Remark 3.6. *Theorem 3.5 establishes approximation rates for piecewise constant functions. It should be noted that the number of required layers is fixed and only depends on the dimension d and the regularity parameter β; in particular, it does* not *depend on the approximation accuracy ε.*

Physics-Informed Neural Network (PINN), a statistical approach

Psichogios-Ungar (92), Lagaris-Likas-Ftiadis (98), Rasissi-Perdikaris-Karniadakis (19), ...

PDE: $\mathcal{L}(u) = 0$ in $\Omega \in \mathcal{R}^d$ and $\mathcal{B}(u) = 0$ on $\partial\Omega$

training data: $\{x_i^u\}_{i=1}^{N_u} \subset \Omega$ and $\{x_i^b\}_{i=1}^{N_b} \subset \partial\Omega$

l^2 **residual:**

$$L(u) = \frac{1}{N_u}\sum_{i=1}^{N_u}\left(\mathcal{L}(u(x_i^u))\right)^2 + \frac{1}{N_b}\sum_{i=1}^{N_b}\left(\mathcal{B}(u(x_i^b))\right)^2$$

(mean squares error)

PINN:

$$u_{\mathcal{N}} = \arg\min_{v\in\mathcal{N}} L(v)$$

Why PINN is uncompetitive?

Issues for NN-based Methods

- **What is a proper formulation of a given PDE?**

  various least-squares formulations

- **How to choose NN architecture for a given problem?**

  adaptive neural enhancement method (ANE)

- **Numerical Issues (unlike finite elements)**
  - Numerical Integration (important): adaptive numerical integration
  - Numerical Differentiation (critical): discrete differential operator
  - Algebraic solver (training NN) (critical): ???

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

Error Estimation for LSNN

- **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**

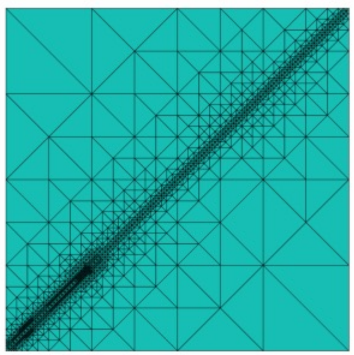
$$|||u - u_N|||_{\boldsymbol{\beta}} \le \left(\frac{M}{\alpha}\right)^{1/2} \inf_{v \in \mathcal{M}(d,n)} |||u - v|||_{\boldsymbol{\beta}}.$$

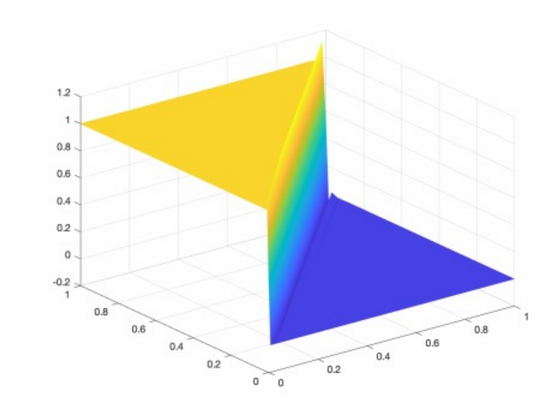
- **A priori error estimate** **(decomposition:** $u(\mathbf{x}) = \hat{u}(\mathbf{x}) + \chi(\mathbf{x}).$**)**

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

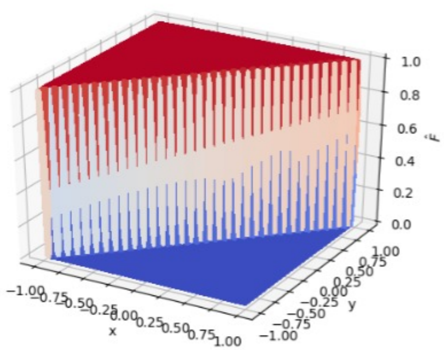
C.-Chen-Liu, JCP, 443 (2021), 110514; C.-Choi-Liu (2022)

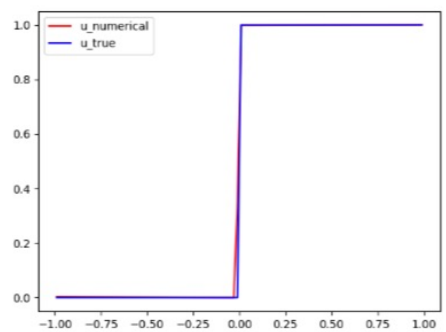
Famous Transport Equation $u_t + u_x = 0$

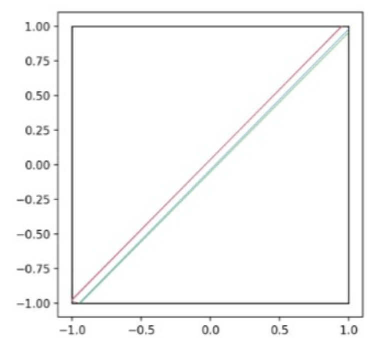




Liu-Zhang, CMAME, 2020

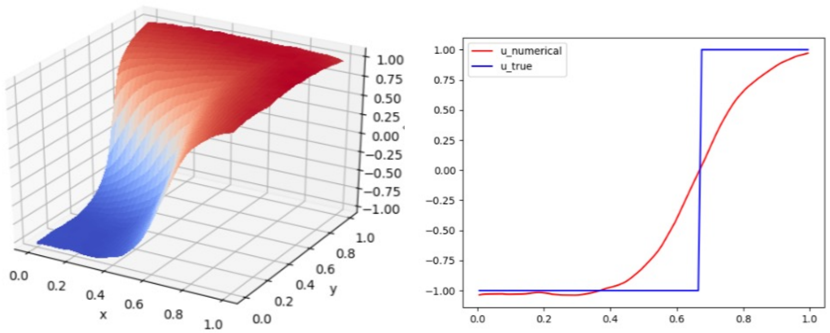




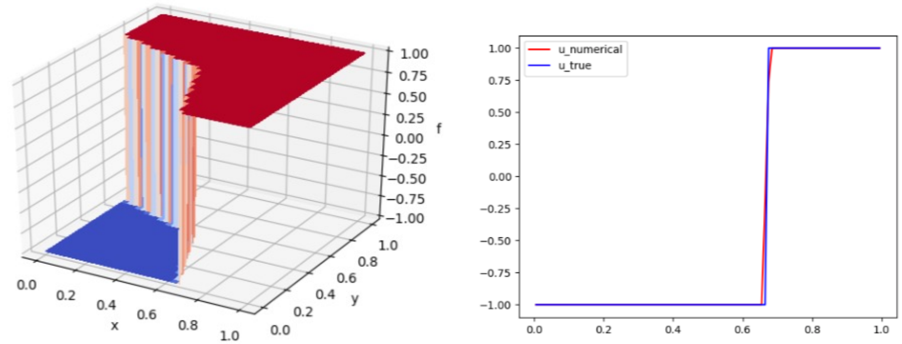


(2-6-1)

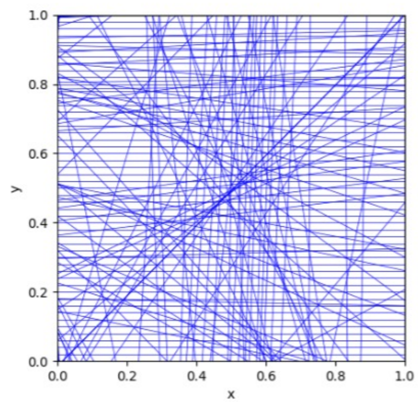
C.-Chen-Liu, JCP, 2021



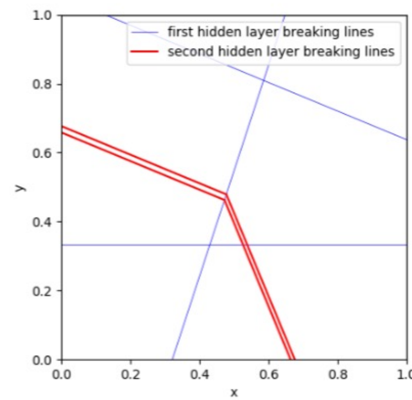
(2-200-1)



(2-5-5-1)

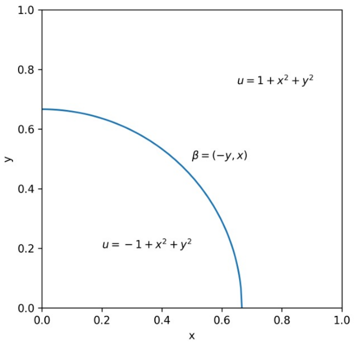


(e) 2-layer NN breaking lines

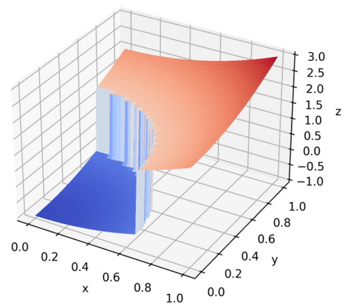


(f) 3-layer NN breaking lines

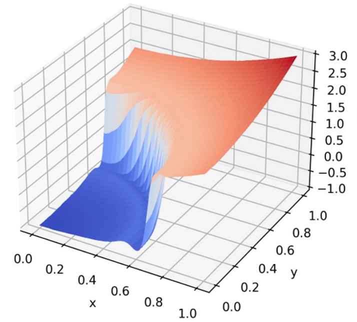
C.-Chen-Liu, LSNN method for linear advection-reaction equation, JCP, 443(2021), 110514.



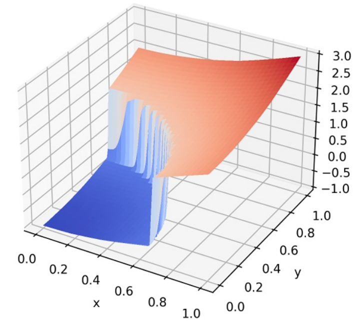
Curve interface



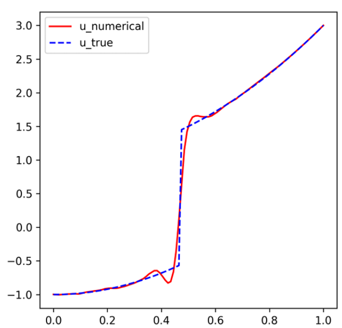
Exact solution

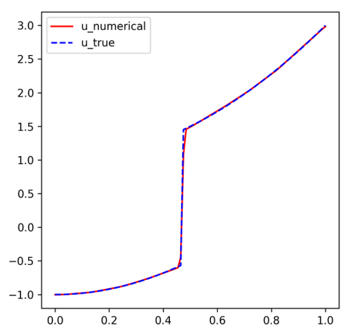


(2-4000-1)

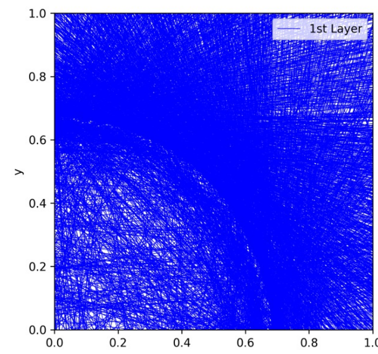


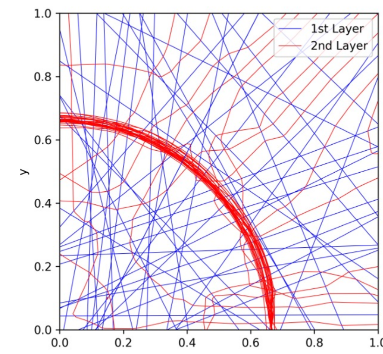
(2-65-65-1)





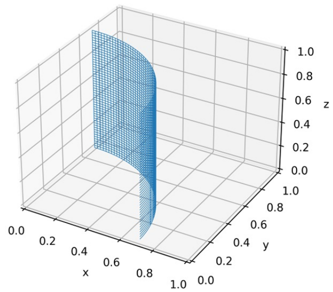
Trace on y=x



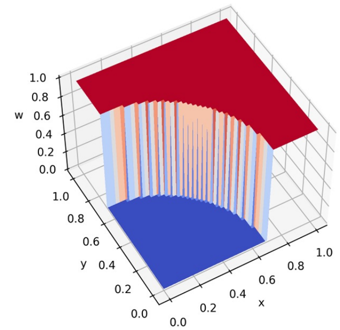


Physical partitions

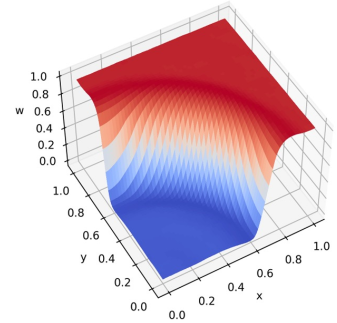
C.-Choi-Liu, LSNN method for linear advection-reaction equation: general discontinuous interface.



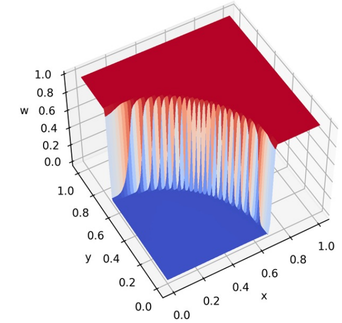
Surface interface



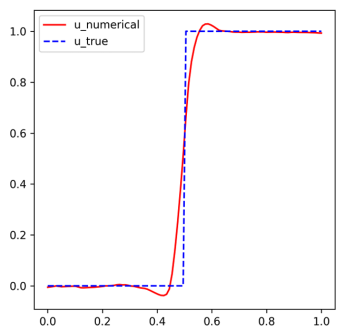
Exact solution at z=0.5

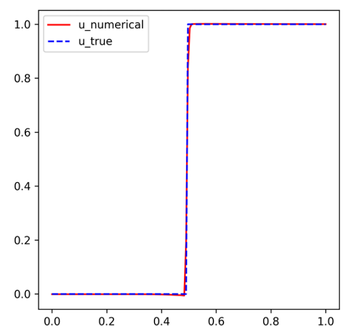


(2-1500-1)

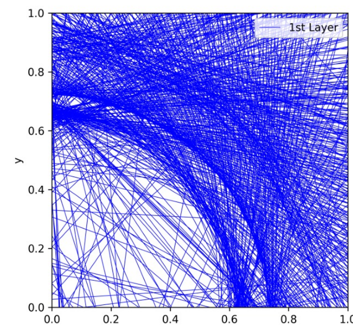


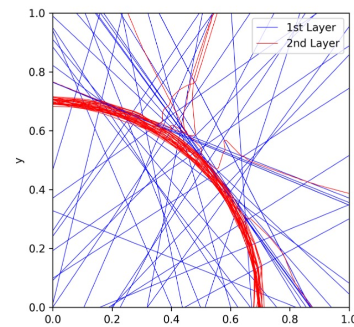
(2-50-50-1)





Trace on y=x





Physical partitions

C.-Choi-Liu, LSNN method for linear advection-reaction equation: general discontinuous interface.

LSNN method for nonlinear scalar HCLs

- **Least-squares formulation**

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

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

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

- **LSNN method** find $u_N \in \mathcal{M}(d,n) = \mathcal{M}(d, 1, \lceil \log_2(d+1) \rceil + 1, n)$ such that

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

- **Numerical Issues:** **discrete divergence operator**

C.-Chen-Liu, ANM (2022) and *J. Comput. Appl. Math. (2023)*

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 (2023), J. Comput. Appl. Math.)**

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.

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

Discrete Divergence Operator in 1D

- **Primitive form over K_{ij}**

$$\frac{1}{|K_{ij}|}\int_{\partial K_{ij}} \mathbf{F}(u)\cdot\mathbf{n}ds = \frac{1}{\delta}\int_{t_j}^{t_{j+1}} \sigma(x_i, x_{i+1}; t)\, dt + \frac{1}{h}\int_{x_i}^{x_{i+1}} u(x; t_j, t_{j+1})dx$$

$$\approx \frac{1}{\delta}Q(\sigma(x_i, x_{i+1}; t); t_j, t_{j+1}, \hat{n}) + \frac{1}{h}Q(u(x; t_j, t_{j+1}); x_i, x_{i+1}, \hat{m}) = \mathrm{div}_{\mathcal{T}}\mathbf{F}\big(u_{ij}\big)$$

- **Error estimate**

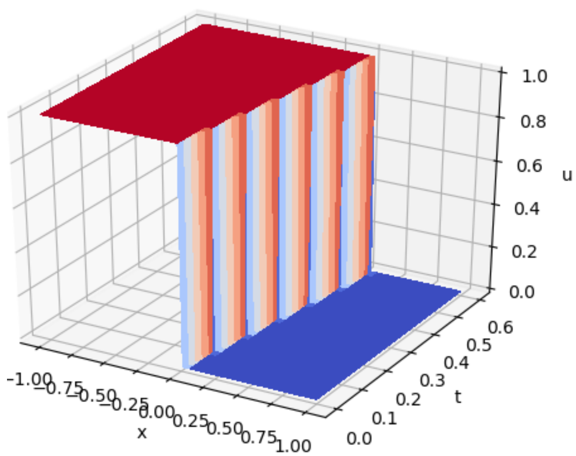
LEMMA 4.3. *Assume that u is a C^2 function of t and a piece-wise C^2 function of x on two vertical and two horizontal edges of K_{ij}, respectively. Moreover, u has only one discontinuous point on each horizontal edge. Then there exists a constant $C > 0$ such that*

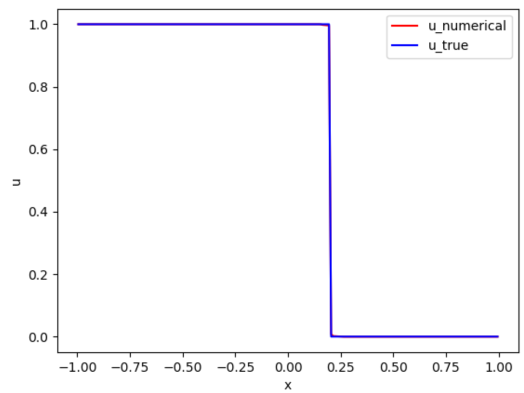
$$\|\mathbf{div}_{\mathcal{T}}\mathbf{f}(u) - \mathrm{avg}_{\mathcal{T}}\mathbf{div}\,\mathbf{f}(u)\|_{L^p(K_{ij})}$$

$$\leq C\left(\frac{h^{1/p}\delta^2}{\hat{n}^2} + \frac{h^2\delta^{1/p}}{\hat{m}^2} + \frac{h\delta^{1/p}}{\hat{m}^{1+1/q}}\right) + \frac{(h\delta)^{1/p}}{\hat{m}}\sum_{l=j}^{j+1}[\![u_{ij}]\!]_{t_l}. \tag{4.7}$$

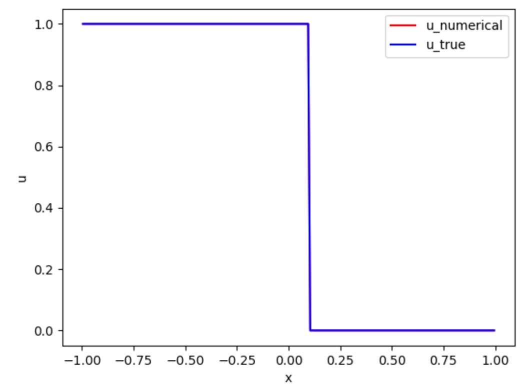
Inviscid Burger Equation $f(u) = \frac{1}{2}u^2$

Riemann Problem Shock formation: exact solution

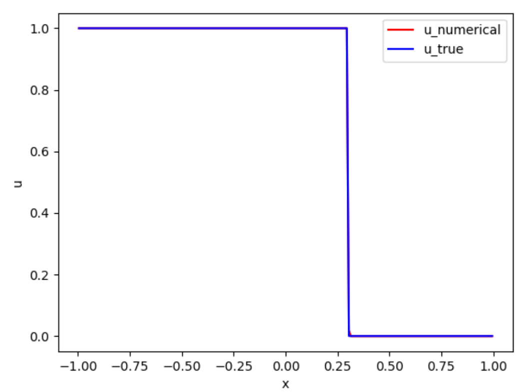




t=0.2



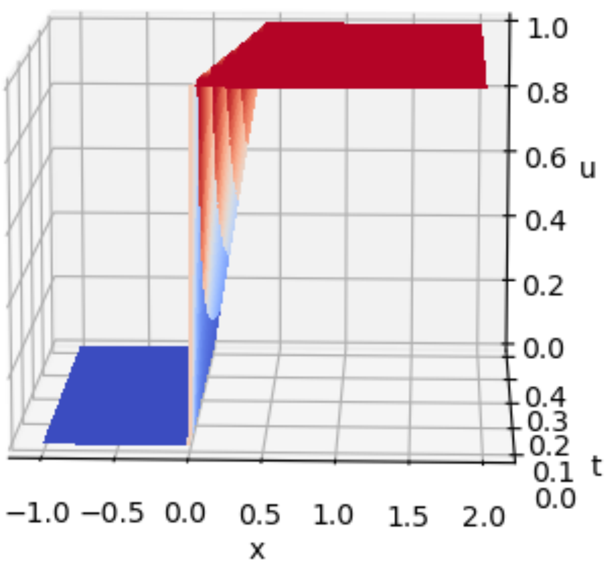
t=0.4

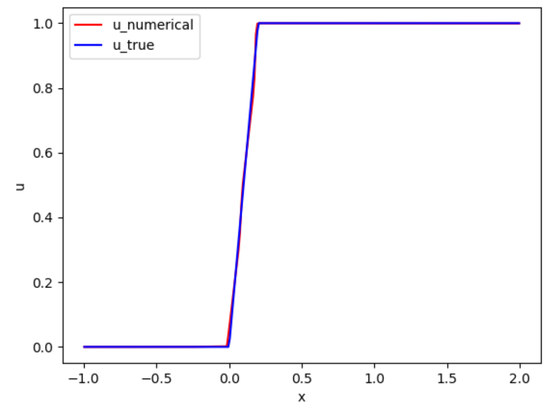


t=0.6

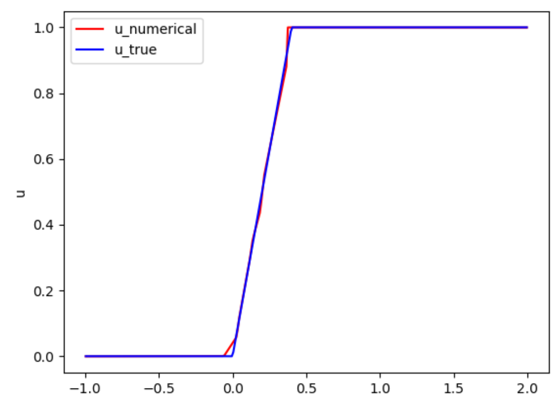
(2-10-10-1)

Riemann Problem
Rarefaction wave: exact solution





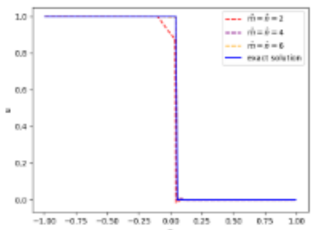
t=0.2



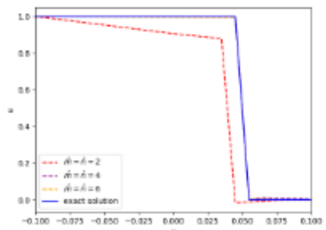
t=0.4

(2-10-10-1)

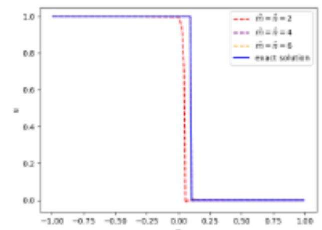
Riemann Problem with Higher order flux $f(u) = \frac{1}{4}u^4$



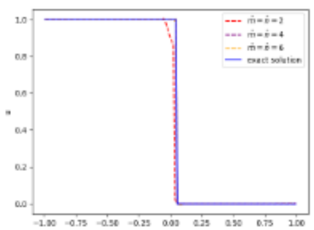
(a) Traces of exact and numerical solutions $u_{1,\mathcal{T}}$ using the trapezoidal rule on the plane $t = 0.2$



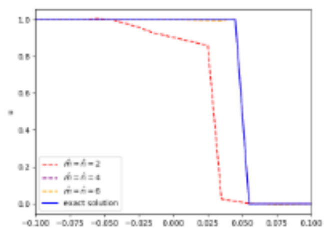
(b) Zoom-in plot near the discontinuous interface of sub-figure (a)



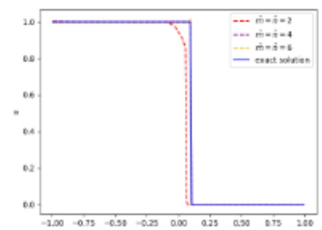
(c) Traces of exact and numerical solutions $u_{2,\mathcal{T}}$ using the trapezoidal rule on the plane $t = 0.4$



(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)

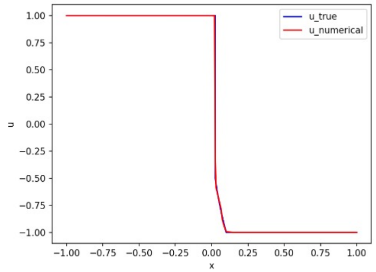


(f) Traces of exact and numerical solutions $u_{2,\mathcal{T}}$ using the mid-point rule on the plane $t = 0.4$

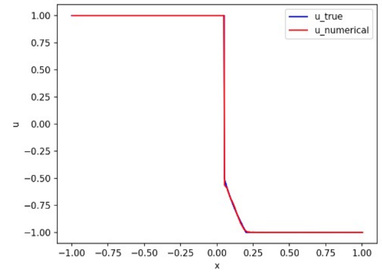
(2-10-10-1)

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

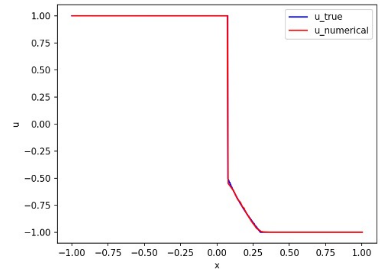
Riemann Problem with Non-convex flux $f(u) = \frac{1}{3}u^3$



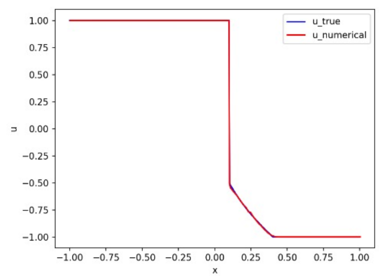
(a) Traces at $t = 0.1$



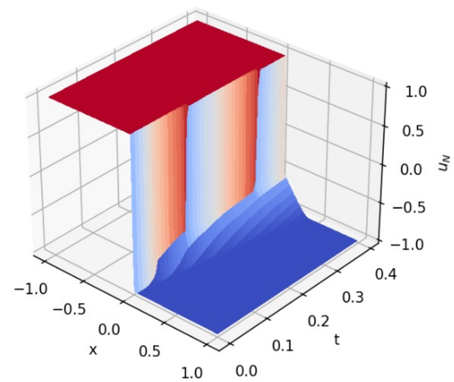
(b) Traces at $t = 0.2$



(c) Traces at $t = 0.3$



(d) Traces at $t = 0.4$

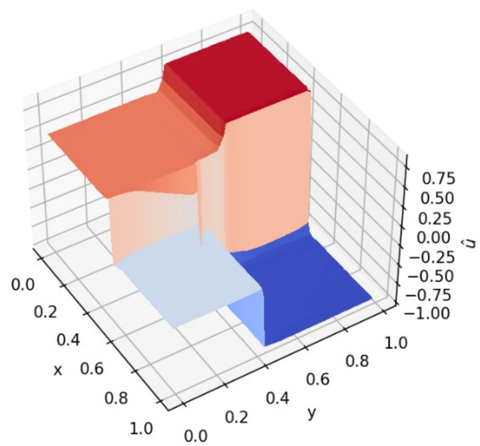


(e) Numerical Solution u_N on Ω

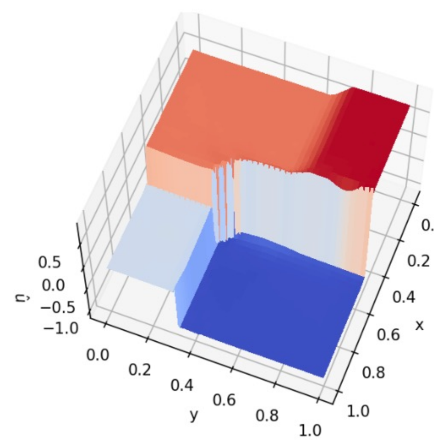
2-64-64-64-1

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

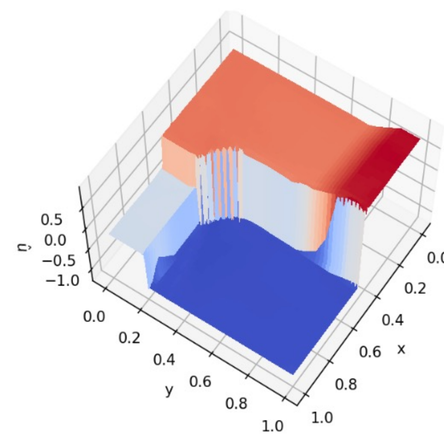
Network structure	Block	$\frac{\|u^k - u^k_{\mathcal{T}}\|_0}{\|u^k\|_0}$
3-48-48-48-1	$\Omega_{0,1}$	0.093679
	$\Omega_{1,2}$	0.121375
	$\Omega_{2,3}$	0.163755
	$\Omega_{3,4}$	0.190460
	$\Omega_{4,5}$	0.213013



(a) $t = 0.1$



(b) $t = 0.3$



(c) $t = 0.5$

Summary

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

  implicit moving "mesh" vs fixed quasi-uniform and adaptive meshes

- **Scalar hyperbolic conservation laws**

  NN is possibly a better class of approximating functions for scalar HCLs than existing ones.

- **Non-convex optimization**

  Bottleneck, the method of continuation, ...

THANK YOU