A Deep Learning Based Discontinuous Galerkin Method for Hyperbolic Equations with Discontinuous Solutions and Random Uncertainties

Liyao Lyu

Department of Computational Mathematics, Science and Engineering, Michigan State University
lyuliyao@msu.edu

Joint work with Shi Jin (SJTU) and Jingrun Chen (Soochow University) (arXiv:2107.01127)

August 12, 2021

Hyperbolic equations with discontinuous solutions and random uncertainties

$$u_t + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}_\omega(u) = 0 \quad (t, \boldsymbol{x}, \boldsymbol{\omega}) \in [0, T] \times D \times \Omega$$

Difficulties

- Shock waves $\longleftarrow$ discontinuous Galerkin method[1]
- High dimensional random space $\longleftarrow$ stochastic Galerkin method[2]

Curse of dimensionality!!!

[1] **cockburn1989tvbo**.
[2] **xiu2010numerical**.

Machine-learning methods for PDEs

- Deep Ritz method[3]: variational formulation
- Deep Galerkin method[4]: least-squares formulation
- Physics-informed neural networks[5]: least-squares formulation in the discrete sense
- etc

[3] **weinan2018deep**.
[4] **sirignano2018dgm**.
[5] **raissi2019physics**.

Components

- Approximation: Approximate the PDE solution by neural network

$$u(x) \approx u(x;\theta)$$

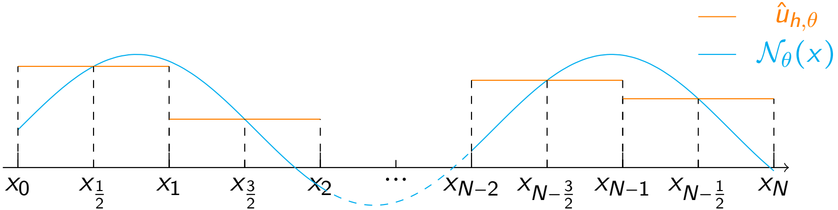
- Modeling: Build a loss function $\mathcal{J}(\theta)$ according to the PDE
- Optimization: Minimize the loss function

$$\theta^* = \arg\min_{\theta} \mathcal{J}(\theta)$$

Discontinuous element basis

The discontinuous element space is defined as

$$V_h^0 = \{v : v|_{I_i} \in P^0(I_i), \quad 0 \le i < N\}$$



This can be formalized in a way like the Galerkin formulation

$$u_{h,\theta}(x) = \sum_{i=0}^{N-1} \mathcal{N}_\theta(x_{i+\frac{1}{2}})\varphi_i(x), \quad \varphi_i(x) = \begin{cases} 1 & x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

Weak formulation

Find $u_h(t,\boldsymbol{x},\boldsymbol{\omega}) \in V_h^k$, such that, $\forall v_h \in V_h^k$ and all $0 \le i < N$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(u_h(t,\boldsymbol{x},\boldsymbol{\omega}), v_h(\boldsymbol{x})\right)_{I_i} - \left(\boldsymbol{f}(u_h(t,\boldsymbol{x},\boldsymbol{\omega})), \nabla v_h(\boldsymbol{x})\right)_{I_i} + \boldsymbol{f}(u_h(t,\boldsymbol{x},\boldsymbol{\omega}))\cdot \mathbf{n} v_h(\boldsymbol{x})\big|_{\partial I_i} = 0$$

$$\left(\frac{u_h(t_{n+1},x,\boldsymbol{\omega}) - u_h(t_n,x,\boldsymbol{\omega})}{\Delta t}, v_h(x)\right)_{I_i} - \left(f(u_h(t_n,x,\boldsymbol{\omega}))_{I_i}, v_h'(x)\right)_{I_i}$$
$$+ \hat{f}_{i+1} v_h(x_{i+1}^-) - \hat{f}_i v_h(x_i^+) = 0$$

- Upwind flux

$$\hat{f}^{\mathrm{upwind}}(u^-,u^+) = f(u^-)$$

- Godunov flux

$$\hat{f}^{\mathrm{God}}(u^-,u^+) = \begin{cases} \min_{u^- \le u \le u^+} f(u), & \text{if } u^- < u^+ \\ \max_{u^+ \le u \le u^-} f(u), & \text{if } u^+ < u^+ \end{cases}$$

With random variables

- Trial function $u_{h,\theta}(t,x,\omega) = \sum_{j'=0}^{K} \sum_{i'=1}^{N-1} \mathcal{N}_\theta^{j'}(t, x_{i'+\frac{1}{2}}, \omega)\varphi_{i'}^{j'}(x)$
- Test function $v_h(x) = \varphi_i^j(x)$

$$L_{i,j,n} \triangleq \frac{\mathcal{N}_\theta^j(t_{n+1}, x_{i+\frac{1}{2}}, \omega) - \mathcal{N}_\theta^j(t_n, x_{i+\frac{1}{2}}, \omega)}{\Delta t} - \left(f(u_{h,\theta}(t_n, x, \omega)), \frac{\mathrm{d}\varphi_i^j(x)}{\mathrm{d}x} \right)_{I_i}$$
$$+ \hat{f}_{i+1}\varphi_i^j(x_{i+1}^-) - \hat{f}_i\varphi_i^j(x_i^+) = 0$$

- Loss function

$$\mathcal{L}(\theta) = h\Delta t \sum_{i,j,n} L_{i,j,n}^2$$

- Monte-Carlo sampling over the indices

Boundary/Initial conditions

- Add a penalty term
- Build a DNN that satisfies the boundary condition exactly[6]
- For a grid-based method
  - Neumann

$$f_0 = f_1, \quad f_N = f_{N-1}$$

  - Periodic

$$f_0 = f_{N-1}, \quad f_N = f_1$$

[6] **lyu2020bc**.

Linear conservation law

Consider

$$\begin{cases} 2d\pi u_t - \sum_{k=1}^{d} u_{x_k} = 0 & x \in [0,1]^d \\ u(0,x) = \sin(2\pi \sum_{k=1}^{d} x_k) \end{cases}$$

with periodic boundary condition, and the exact solution $u(t,x) = \sin(t + 2\pi \sum_{k=1}^{d} x_k)$

First-order method

- Trial solution that satisfies the initial condition

$$u_{h,\theta}(t,\boldsymbol{x}) = \sum_{\boldsymbol{i}} [t\mathcal{N}_\theta(t, x_{\boldsymbol{i}+\frac{1}{2}}) + g(x_{\boldsymbol{i}+\frac{1}{2}})]\varphi_{\boldsymbol{i}}(\boldsymbol{x}) \quad \varphi_{\boldsymbol{i}}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in I_{\boldsymbol{i}} \\ 0 & \text{otherwise} \end{cases}$$

- 1D: $U_i(t) = t\mathcal{N}(t, x_{i+\frac{1}{2}}) + \sin(2\pi x_{i+\frac{1}{2}})$
- 2D: $U_{i_1,i_2}(t) = t\mathcal{N}(t, x^1_{i_1+\frac{1}{2}}, x^2_{i_2+\frac{1}{2}}) + \sin(2\pi(x^1_{i_1+\frac{1}{2}} + x^2_{i_2+\frac{1}{2}}))$
- 3D: $U_{i_1,i_2,i_3}(t) = t\mathcal{N}(t, x^1_{i_1+\frac{1}{2}},, x^2_{i_2+\frac{1}{2}}, x^3_{i_3+\frac{1}{2}}) + \sin(2\pi(x^1_{i_1+\frac{1}{2}} + x^2_{i_2+\frac{1}{2}} + x^3_{i_3+\frac{1}{2}}))$
- Boundary condition

$$U_0(t) = U_{N-1}(t), \quad U_N(t) = U_1(t)$$

Loss function

- Semi-discrete scheme

$$\mathcal{L}_{\mathrm{semi}}(\theta) = \sum_{i_i, i_2, i_3, j} \Bigg(6\pi \partial_t U_{i_1,i_2,i_3}(t_j) - \frac{U_{i_1+1,i_2,i_3}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} - \frac{U_{i_1,i_2+1,i_3}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} - \frac{U_{i_1,i_2,i_3+1}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} \Bigg)^2$$

- Fully discrete scheme based on the forward Euler method

$$\mathcal{L}_{\mathrm{FE}}(\theta) = \sum_{i_i, i_2, i_3, j} \Bigg(6\pi \frac{U_{i_1,i_2,i_3}(t_{j+1}) - U_{i_1,i_2,i_3}(t_j)}{\Delta t} - \frac{U_{i_1+1,i_2,i_3}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} - \frac{U_{i_1,i_2+1,i_3}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} - \frac{U_{i_1,i_2,i_3+1}(t_j) - U_{i_1,i_2,i_3}(t_j)}{h} \Bigg)^2$$

d	$h = \Delta t$	Fully discrete		Semi-discrete	
		error	order	error	order
1	$1/20$	1.50 e-01	0.93	1.58 e-01	0.93
	$1/40$	7.73 e-02	0.94	8.05 e-02	0.98
	$1/80$	3.95 e-02	0.96	8.39 e-02	-0.05
	$1/160$	2.10 e-02	0.91	7.01 e-02	0.25
2	$1/20$	1.72 e-01	0.90	1.81 e-01	0.91
	$1/40$	8.90 e-02	0.95	8.89 e-02	1.03
	$1/80$	4.68 e-02	0.92	6.00 e-02	0.56
	$1/160$	2.57 e-02	0.86	5.40 e-02	0.15
3	$1/20$	1.92 e-01	0.90	2.02 e-01	0.89
	$1/40$	9.95 e-02	0.95	1.03 e-01	0.96
	$1/80$	5.20 e-02	0.93	6.94 e-02	0.57
	$1/160$	3.14 e-02	0.72	9.45 e-02	-0.44

Table: The averaged L^2 relative error and the convergence rate for the linear conservation law. Batchsize: 10000; Depth: 4 hidden layers and 2 shortcut connections. Width: 20 (1D), 40 (2D), 60 (3D)

$h = \Delta t$	error	order
$1/10$	3.64 e-01	
$1/20$	1.92 e-01	0.92
$1/40$	9.92 e-02	0.95
$1/80$	5.04 e-02	0.97
$1/160$	2.54 e-02	0.98
$1/320$	1.29 e-02	0.97
$1/640$	6.78 e-03	0.93
$1/1280$	4.24 e-03	0.67

Table: The averaged L^2 relative error and the convergence rate for the linear conservation law using fully discrete loss function and a wider (200 width) neural network with 883601 parameters in 3D.

Second-order method

- Trial solution

$$u_{h,\theta}(t,x) = \sum_i [U_i^0 \varphi_i^0(x) + U_i^1 \varphi_i^1(x)]$$

where

$$\varphi_i^0(x) = \begin{cases} 1 & x \in I_i \\ 0 & \text{otherwise} \end{cases},$$

$$\varphi_i^1(x) = \begin{cases} (x - x_{i+\frac{1}{2}}) & x \in I_i \\ 0 & \text{otherwise} \end{cases},$$

$$U_i^0(t) = t\mathcal{N}_{\theta_0}^0(t, x_{i+\frac{1}{2}}) + \sin(2\pi x_{i+\frac{1}{2}}),$$

$$U_i^1(t) = t\mathcal{N}_{\theta_1}^1(t, x_{i+\frac{1}{2}}) + 2\pi\cos(2\pi x_{i+\frac{1}{2}}).$$

Cont'd ...

- Loss function

$$\mathcal{L}_0(\theta_0) = \sum_{i,j} \left(2\pi \frac{U_i^0(t_{j+1}) - U_i^0(t_j)}{\Delta t} h + \hat{f}_{i+\frac{3}{2}} - \hat{f}_{i+\frac{1}{2}} \right)^2$$

$$\mathcal{L}_1(\theta_1) = \sum_{i,j} \left(2\pi \frac{U_i^1(t+\Delta t) - U_i^1(t)}{\Delta t} \frac{h^3}{12} + hu_0 + \frac{h}{2}\hat{f}_{i+\frac{3}{2}} + \frac{h}{2}\hat{f}_{i+\frac{1}{2}} \right)^2$$

- Optimization: ADMM[7]

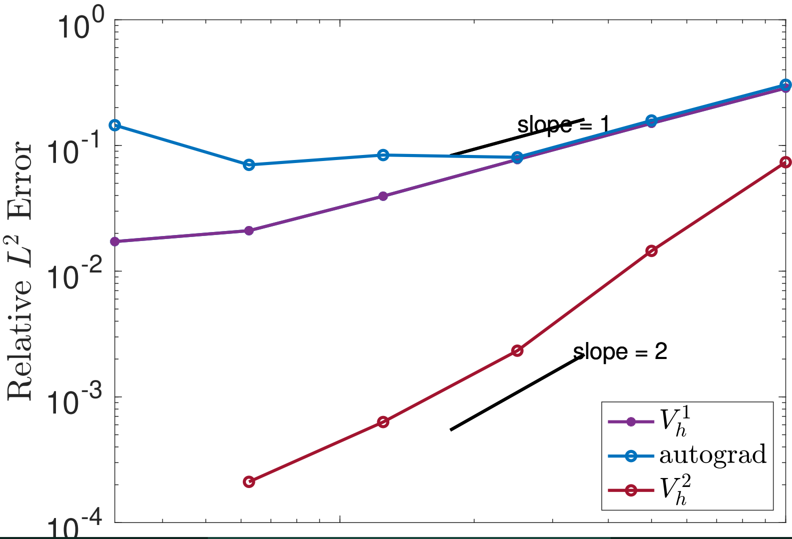
$$\arg\min_{\{\theta_0,\theta_1\}} \mathcal{L}_0(\theta_0) + \mathcal{L}_1(\theta_1)$$

[7] **boyd2011distributed**.

$h = \sqrt{\Delta t}$	error	order
$1/10$	7.36 e-02	
$1/20$	1.45 e-02	2.34
$1/40$	2.33 e-03	2.63
$1/80$	6.31 e-04	1.88
$1/160$	2.11 e-04	1.57

Table: The averaged L^2 relative error and the convergence rate for the 1D linear conservation law solved by the second-order scheme. Batchsize: 10000; Total number of parameters: 12951.

Comparison



10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
Relative L^2 Error
slope = 1
slope = 2
V_h^1
autograd
V_h^2

Burgers' Equation

$$u_t + (\frac{u^2}{2})_x = 0$$

with initial condition

$$u(0,x) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases}$$

and reflecting boundary condition

Numerical solution

$$u(t,x) = \begin{cases} \mathcal{N}_\theta(t, x_{i+\frac{1}{2}})\varphi_i(x) & t > 0 \; x \in (x_i, x_{i+1}) \\ u(0, x_{i+\frac{1}{2}}) & t = 0 \end{cases}$$

Loss function

Semi-discrete scheme

$$\mathcal{L}_{\mathrm{semi}}(\theta) = \sum_{i,j} \left(\frac{\partial u_\theta(t_j, x_{i+\frac{1}{2}})}{\partial t} - \hat{f}^{\mathrm{God}}(u_\theta(t_j, x_i^-), u_\theta(t_j, x_i^+)) \right.$$
$$\left. + \hat{f}^{\mathrm{God}}(u_\theta(t_j, x_{i+1}^-), u_\theta(t_j, x_{i+1}^+)) \right)^2$$

Fully-discrete scheme using the forward Euler method

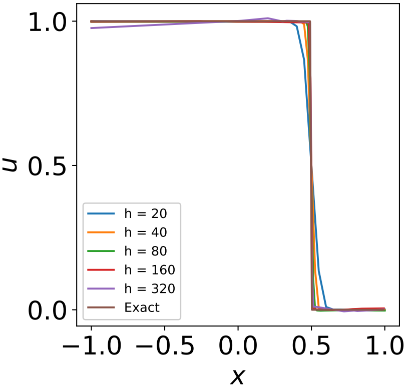
$$\mathcal{L}_{\mathrm{FE}}(\theta) = \sum_{i,j} \left(\frac{u_\theta(t_{j+1}, x_{i+\frac{1}{2}}) - u_\theta(t_j, x_{i+\frac{1}{2}})}{\Delta t} - \hat{f}^{\mathrm{God}}(u_\theta(t_j, x_i^-), u_\theta(t_j, x_i^+)) \right.$$
$$\left. + \hat{f}^{\mathrm{God}}(u_\theta(t_j, x_{i+1}^-), u_\theta(t_j, x_{i+1}^+)) \right)^2$$

Approximation error

$h = \Delta t$	Fully discrete	Semi-discrete
$1/10$	9.87 e-02	3.82 e-01
$1/20$	4.88 e-02	3.37 e-01
$1/40$	3.48 e-02	3.03 e-01
$1/80$	2.58 e-02	3.12 e-01
$1/160$	1.84 e-02	1.91 e-01
$1/320$	1.73 e-02	3.86 e-01

Table: The averaged L^2 relative error and the convergence rate for the Burgers' equation. Batchsize: 10000; Width 20, Depth:4, Total number of parameters: 1341.

Solution profiles



Stochastic linear conservation law

$$2d\pi u_t - (1 + \exp((-\sum_{j=1}^{s} \omega_j)^2)) \sum_i u_{x_i} = 0$$

with periodic boundary condition and initial condition $u(0, \boldsymbol{x}, \boldsymbol{\omega}) = g(\boldsymbol{x}) = \sin\left(2\pi \sum_{i=1}^{d} x_i\right)$

Exact solution $u(t, \boldsymbol{x}, \boldsymbol{\omega}) = \sin\left((1 + \exp((-\sum_{j=1}^{s} \omega_j)^2))t + 2\pi \sum_{i=1}^{d} x_i\right)$

Numerical solution

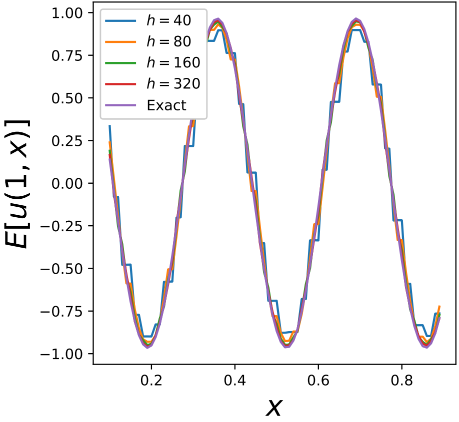
$$u(t, \boldsymbol{x}, \boldsymbol{\omega}) = \sum_{\boldsymbol{i}} [t\mathcal{N}_\theta(t, \boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}}, \boldsymbol{\omega}) + g(\boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}})]\varphi_{\boldsymbol{i}}(\boldsymbol{x})$$

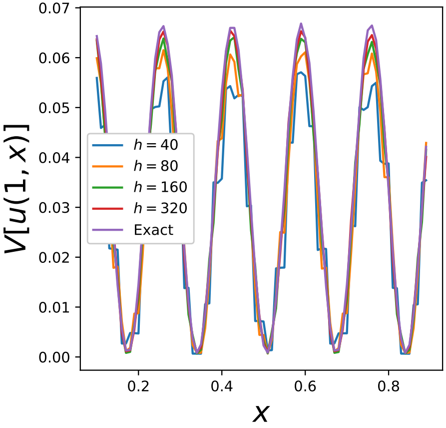
Loss function

$$\mathcal{L}_{\mathrm{FE}}(\theta) = \sum_{i,j} \Bigg(2\pi \frac{u_\theta(t_{j+1}, \boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}}, \boldsymbol{\omega}) - u_\theta(t_j, \boldsymbol{x}_{\boldsymbol{i}+\frac{1}{2}}, \boldsymbol{\omega})}{\Delta t} - (1 + \exp(-\sum_{j=1}^{s} \omega_j)^2) \frac{u_\theta(t_j, \boldsymbol{x}_{\boldsymbol{i}+1}, \omega) - u_\theta(t_j, \boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{\omega})}{h} \Bigg)^2$$

s	$h = \Delta t$	Expectation	Order	Variance	Order
100	$1/40$	1.53 e-01		2.07 e-01	
100	$1/80$	7.83 e-02	0.97	1.12 e-01	0.88
100	$1/160$	3.93 e-02	0.99	5.82 e-02	0.95
100	$1/320$	2.01 e-02	0.96	2.93 e-02	0.98

Solution profiles





Stochastic Burgers' equation

$$u_t + (\frac{u^2}{2})_x = 0$$

with initial condition

$$u(0, x, \boldsymbol{\omega}) = \begin{cases} 1 + \epsilon \sum_{i=1}^{s} \omega_i & x < 0 \\ 0 & x > 0 \end{cases}$$

Exact solution

$$u(t, x, \boldsymbol{\omega}) = \begin{cases} z & x < \frac{z}{2} \\ 0 & x > \frac{z}{2} \end{cases}$$

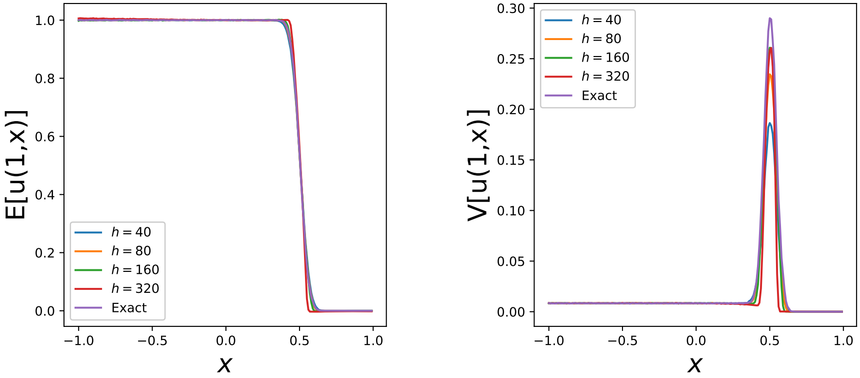
where $z = 1 + \epsilon \sum_{i=1}^{s} \omega_i$

DNN solution

$$u(t,x,\boldsymbol{\omega}) = \begin{cases} \mathcal{N}_\theta(t, x_{i+\frac{1}{2}}, \boldsymbol{\omega})\varphi_i(x) & t > 0 \ x \in (x_i, x_{i+1}) \\ u(0, x_{i+\frac{1}{2}}, \boldsymbol{\omega}) & t = 0 \end{cases}$$

ϵ	s	h	L^2 error		L^1 error	
			Expectation	Variance	Expectation	Variance
0.05	10	1/40	6.79 e-3	3.37 e-1	2.99 e-03	2.40 e-01
0.05	10	1/80	2.25 e-3	1.86 e-1	1.13 e-03	1.45 e-01
0.05	10	1/160	4.68 e-3	1.27 e-1	2.28 e-03	1.17 e-01
0.05	10	1/320	2.01 e-2	3.36 e-1	8.94 e-03	2.74 e-01
0.01	50	1/40	1.80 e-2	6.42 e-1	5.36 e-03	5.01 e-01
0.01	50	1/80	5.74 e-3	4.04 e-1	1.67 e-03	3.32 e-01
0.01	50	1/160	3.09 e-3	2.69 e-1	1.18 e-03	2.68 e-01
0.01	50	1/320	4.40 e-2	9.12 e-1	1.70 e-02	8.06 e-01
0.005	100	1/40	2.58 e-2	7.53 e-1	6.54 e-03	5.01 e-01
0.005	100	1/80	8.64 e-3	5.25 e-1	2.09 e-03	3.32 e-01
0.005	100	1/160	1.95 e-3	3.76 e-1	7.22 e-04	2.68 e-01
0.005	100	1/320	3.07 e-2	9.47 e-1	5.65 e-03	8.06 e-01
0.0025	200	1/40	2.58 e-2	7.53 e-1	7.56 e-03	7.52 e-01
0.0025	200	1/80	8.64 e-3	5.25 e-1	2.51 e-03	5.68 e-01
0.0025	200	1/160	1.95 e-3	3.76 e-1	8.30 e-04	5.51 e-01
0.0025	200	1/320	3.07 e-2	9.47 e-1	3.52 e-03	9.09 e-01

Table: Expectation and variance errors of the proposed method for the stochastic Burgers' equation when the MC method is used with the batchsize 50000.


E[u(1,x)]
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
x
h = 40
h = 80
h = 160
h = 320
Exact
V[u(1,x)]
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−1.0
−0.5
0.0
0.5
1.0
x
h = 40
h = 80
h = 160
h = 320
Exact

Conclusion

Discretize-then-learn

- A stable and convergent scheme in the classical sense
- A neural network representation of the numerical solution
- A loss function in the least-squares sense
- Monte-Carlo sampling

Thank you for your attention!