适用于多种数值格式的深度学习方法

常芸凡[1]
指导老师：杨顶辉教授[1]、贺茜君副教授[2]

[1] 清华大学数学科学系

[2] 北京工商大学数学科学系

2022 年 12 月 21 日

研究背景

目前有很多利用神经网络来快速求解偏微分方程的思想: 如 Deep Ritz 方法 [Weinan and Yu, 2017]、Physics-informed neural networks (PINN) [Raissi et al., 2019] 等，但是这些方法对于我们的地震波动方程来说效果并不是很理想。

借助于 [Chen et al., 2021] 的思路，我们提出了一种新的求解方法，将深度学习方法嫁接在我们传统的数值格式之中，从某个角度可以理解为，这种方法打破了传统数值格式中 CFL 条件的限制。

PINN

We define $f(t,x)$ to be given by the left-hand-side of equation (2); i.e.,

$$f := u_t + \mathcal{N}[u], \tag{3}$$

and proceed by approximating $u(t,x)$ by a deep neural network. This assumption along with equation (3) result in a *physics-informed neural network* $f(t,x)$. This network can be derived by applying the chain rule for differentiating compositions of functions using automatic differentiation [12], and has the same parameters as the network representing $u(t,x)$, albeit with different activation functions due to the action of the differential operator $\mathcal{N}$. The shared parameters between the neural networks $u(t,x)$ and $f(t,x)$ can be learned by minimizing the mean squared error loss

$$MSE = MSE_u + MSE_f, \tag{4}$$

where

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2,$$

and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2.$$

Here, $\{t_u^i, x_u^i, u^i\}_{i=1}^{N_u}$ denote the initial and boundary training data on $u(t,x)$ and $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ specify the collocations points for $f(t,x)$. The loss MSE_u corresponds to the initial and boundary data while MSE_f enforces the structure imposed by equation (2) at a finite set of collocation points. Although similar ideas for constraining neural networks using physical laws have been explored in previous studies [15,16], here we revisit them using modern computational tools, and apply them to more challenging dynamic problems described by time-dependent nonlinear partial differential equations.

PINN

其中 $MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f}|f(t_f^i, x_f^i)|^2$ 是 PINN 主要的内存消耗和计算量。

反向传播算法计算过程:

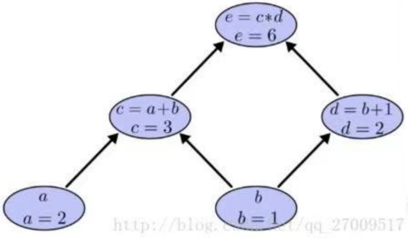

图 1: 计算图

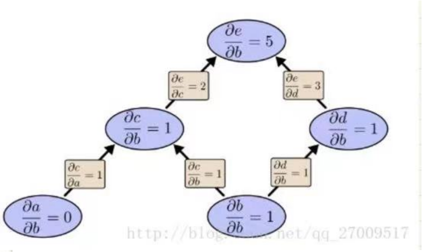

图 2: 反向传播计算 $\frac{\partial e}{\partial b}$ 流程

1 维声波方程

我们考虑如下形式的声波方程：

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f$$

初边值条件均为 0，点源

$$f(t,x) = -9.6 f_0 (0.6 f_0 t - 1) \exp(-8(0.6 f_0 t - 1)^2) \delta(x - x_0)$$

其中 $c = 2$ 表示波速，$f_0 = 20$ 表示频率，我们在下面的数值计算中选取 $t = 0$ 时刻作为初值，计算区域为 $0 \leq x \leq 4\mathrm{km}$。

LWC 方法

由泰勒展开，我们有

$$\begin{aligned} U_j^{n+1} = U_j^n + \Delta t \left(\frac{\partial U}{\partial t}\right)_j^n + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)_j^n \\ + \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_j^n + \frac{(\Delta t)^4}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_j^n, \end{aligned} \tag{1}$$

和

$$\begin{aligned} U_j^{n-1} = U_j^n - \Delta t \left(\frac{\partial U}{\partial t}\right)_j^n \\ + \frac{(\Delta t)^2}{2}\left(\frac{\partial^2 U}{\partial t^2}\right)_j^n - \frac{(\Delta t)^3}{6}\left(\frac{\partial^3 U}{\partial t^3}\right)_j^n + \frac{(\Delta t)^4}{24}\left(\frac{\partial^4 U}{\partial t^4}\right)_j^n, \end{aligned} \tag{2}$$

两式相加可得:

$$\begin{aligned} u_j^{n+1} &= 2u_j^n - u_j^{n-1} + (\Delta t)^2\left(\frac{\partial^2 u}{\partial t^2}\right)_j^n + \frac{(\Delta t)^4}{12}\left(\frac{\partial^4 u}{\partial t^4}\right)_j^n \\ &= 2u_j^n - u_j^{n-1} + (\Delta t \cdot c)^2\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n + (\Delta t)^2 f(t,x)_j^n \quad , \\ &+ \frac{(\Delta t)^4}{12}\left(c^4\frac{\partial^4 u}{\partial x^4} + c^2 f_{xx} + f_{tt}\right)_j^n \end{aligned} \tag{3}$$

LWC 方法

利用 Lax–Wendroff 格式，Dablain 发展了如下的近似格式

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_j^n = \frac{1}{\Delta x^2}\left[w_0 u_j^n + \sum_{i=1}^{M} w_i \left(u_{j+i}^n + u_{j-i}^n\right)\right],$$

将上式代入 (3) 式中可得 LWC 格式。

神经网络方法

于是 LWC 的损失函数为：

$$L_{j,n} = (\mathcal{N}_\theta)_j^{n+1} - 2(\mathcal{N}_\theta)_j^n + (\mathcal{N}_\theta)_j^{n-1} - (\Delta t \cdot C)^2 \left(\frac{\partial^2 (\mathcal{N}_\theta)}{\partial x^2}\right)_j^n - (\Delta t)^2 f(t,x)_j^n - \frac{(\Delta t)^4}{12}\left(c_0^4 \frac{\partial^4 (\mathcal{N}_\theta)}{\partial x^4} + c^2 f_{xx} + f_{tt}\right)_j^n \tag{4}$$

1 研究背景

2 基于深度学习的传统数值方法
- 声波方程的有限差分方法
- 初边值条件的处理

3 数值算例

4 下一步工作安排

5 参考文献

初边值条件为:

$$\begin{aligned} u(0,x) &= g(x), & x &\in \Omega, \\ u(t,x) &= h(t,x), & x &\in \partial\Omega, \quad 0 \le t \le T, \end{aligned} \tag{5}$$

我们直接在损失函数中加入两个惩罚项:

$$\begin{aligned} \mathcal{L}_{u_i}(\theta) &= \|u_\theta(0,x) - g(x)\|_2 = \left(\int_\Omega (u_\theta(0,x) - g(x))^2 \mathrm{d}x\right)^{\frac{1}{2}} \\ \mathcal{L}_{u_b}(\theta) &= \|u_\theta(t,x) - h(t,x)\|_2 = \left(\int_{[0,T]\times\partial\Omega} (u_\theta(t,x) - h(t,x))^2 \mathrm{d}t\mathrm{d}x\right)^{\frac{1}{2}}. \end{aligned} \tag{6}$$

与 PINN 的区别

We define $f(t,x)$ to be given by the left-hand-side of equation (2); i.e.,

$$f := u_t + \mathcal{N}[u], \tag{3}$$

and proceed by approximating $u(t,x)$ by a deep neural network. This assumption along with equation (3) result in a *physics-informed neural network* $f(t,x)$. This network can be derived by applying the chain rule for differentiating compositions of functions using automatic differentiation [12], and has the same parameters as the network representing $u(t,x)$, albeit with different activation functions due to the action of the differential operator $\mathcal{N}$. The shared parameters between the neural networks $u(t,x)$ and $f(t,x)$ can be learned by minimizing the mean squared error loss

$$MSE = MSE_u + MSE_f, \tag{4}$$

where

$$MSE_u = \frac{1}{N_u}\sum_{i=1}^{N_u} |u(t_u^i, x_u^i) - u^i|^2,$$

and

$$MSE_f = \frac{1}{N_f}\sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2.$$

Here, $\{t_u^i, x_u^i, u^i\}_{i=1}^{N_u}$ denote the initial and boundary training data on $u(t,x)$ and $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ specify the collocations points for $f(t,x)$. The loss MSE_u corresponds to the initial and boundary data while MSE_f enforces the structure imposed by equation (2) at a finite set of collocation points. Although similar ideas for constraining neural networks using physical laws have been explored in previous studies [15,16], here we revisit them using modern computational tools, and apply them to more challenging dynamic problems described by time-dependent nonlinear partial differential equations.

我们考虑如下的 Burgers 方程:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{7}$$

初值条件为:

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

我们选取 Burgers 方程，一方面可以验证间断有限元方法对于具有间断性质问题的可行性，一方面也是验证间断有限元与深度学习结合后是否还能具有良好的效果。

方程 (7) 具有间断解析解:

$$u(t,x) = \begin{cases} 1 & ,x < t/2 \\ 0 & ,x > t/2 \end{cases}$$

我们将区间 $[-1,1]$ 均分成 N 份，也就是分为 N 个大小相同的单元 $I_1, I_2, \cdots, I_N$

数值结果

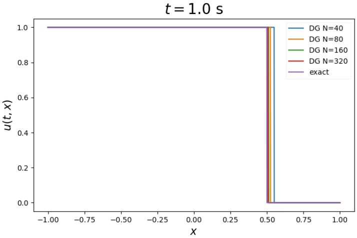

图 3: 1 阶间断有限元结果

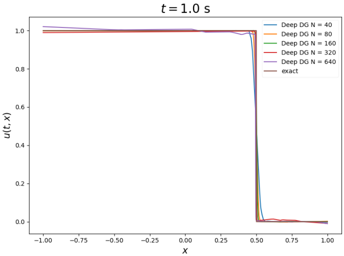

图 4: 1 阶深度神经网络 (间断有限元) 结果

误差下降曲线

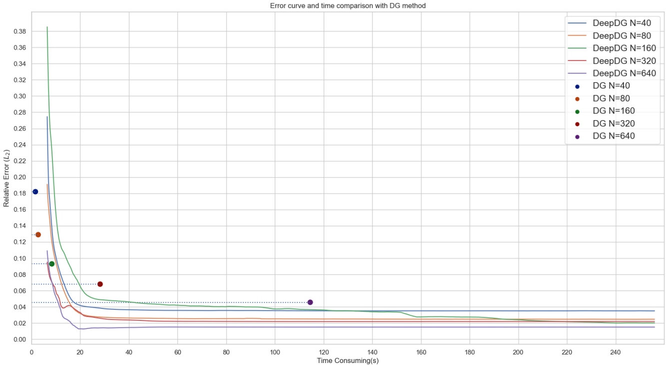

图 5: 深度学习误差下降曲线和传统方法的用时对比

PINN 误差下降曲线

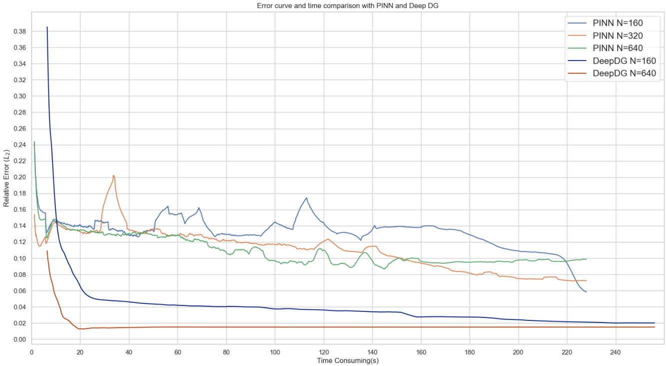

图 6: PINN 误差下降曲线

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=320)	28.2	6.8	1280
深度学习方法	误差达(6.8%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG(N=40)	13.3 (0.47)	3.5	14 (0.011)
DeepDG(N=80)	12.3 (0.43)	2.5	14 (0.011)
DeepDG(N=160)	19.4 (0.69)	2.0	14 (0.011)
DeepDG(N=320)	8.7 (0.30)	2.1	14 (0.011)
DeepDG(N=640)	8.6 (0.30)	1.2	14 (0.011)

图 7: burgers' 方程数据对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=640)	114.5	4.6	3840
深度学习方法	误差达(4.6%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG(N=40)	17.2 (0.15)	3.5	14 (0.0036)
DeepDG(N=80)	15.4 (0.13)	2.5	14 (0.0036)
DeepDG(N=160)	40.3 (0.35)	2.0	14 (0.0036)
DeepDG(N=320)	11.7 (0.10)	2.1	14 (0.0036)
DeepDG(N=640)	10.9 (0.09)	1.2	14 (0.0036)

图 8: burgers' 方程数据对比

1. 研究背景
2. 基于深度学习的传统数值方法
3. 数值算例
   - Burgers' 方程
   - 1 维声波方程
   - 点源一维声波方程
4. 下一步工作安排
5. 参考文献

我们考虑如下的一维声波方程

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \tag{8}$$

初值条件为

$$\begin{cases} u(x,0) = \cos\left(-\frac{2\pi f_0}{c}x\right) \\ \dfrac{\partial u(x,0)}{\partial t} = -2\pi f_0 \sin\left(-\frac{2\pi f_0}{c}x\right) \end{cases}$$

该方程有解析解:

$$u(t,x) = \cos\left(2\pi f_0 t - \frac{2\pi f_0}{c}x\right)$$

计算区域为 $0 \leq x \leq 4\text{km}, 0 \leq t \leq 0.5s$。

1 阶格式结果

我们采用 LLF 通量，并用二阶 TVD 时间离散格式得到结果如下：

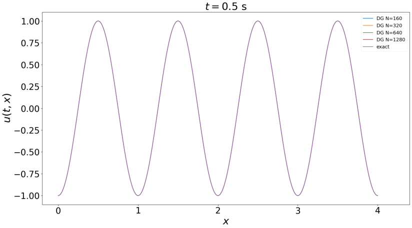

图 9: 传统 DG 方法 1 阶格式结果 $t = 0.5$

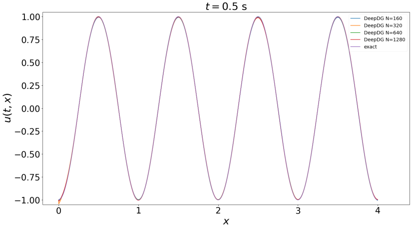

图 10: 深度学习 DG 方法 1 阶格式结果 $t = 0.5$

误差下降曲线

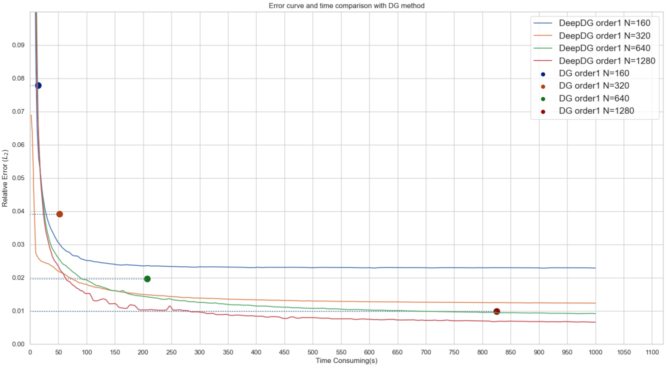

图 11: 深度学习误差下降曲线和传统方法的用时对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=640)	207.39	1.96	10240
深度学习方法	误差达(1.96%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG order1(N=320)	74.4 (0.36)	1.2	42 (0.0041)
DeepDG order1(N=640)	91.4 (0.44)	0.91	42 (0.0041)
DeepDG order1(N=1280)	63.4 (0.30)	0.65	42 (0.0041)

图 12: 1 维声波方程方程 1 阶格式数据对比

2 阶格式结果

我们采用 LLF 通量，并用二阶 TVD 时间离散格式得到结果如下：

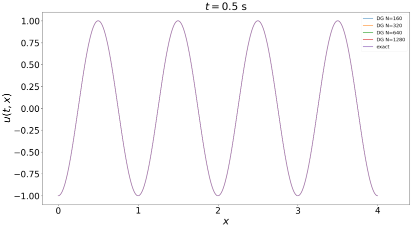

图 13: 传统 DG 方法 2 阶格式结果 $t = 0.5$

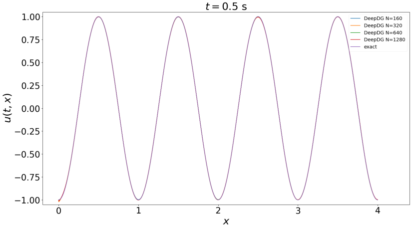

图 14: 深度学习 DG 方法 2 阶格式结果 $t = 0.5$

误差下降曲线

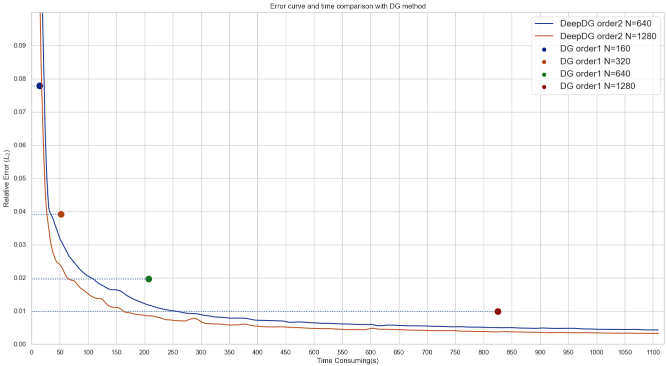

图 15: 深度学习误差下降曲线和传统方法的用时对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=640)	207.39	1.96	10240
深度学习方法	误差达(1.96%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG order2(N=640)	109.5 (0.52)	0.42	63 (0.0062)
DeepDG order2(N=1280)	65.8 (0.31)	0.32	63 (0.0062)

图 16: 1 维声波方程方程 1 阶格式数据对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=1280)	825.53	0.98	39680
深度学习方法	误差达(0.98%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG order2(N=640)	259.4 (0.31)	0.42	63 (0.0062)
DeepDG order2(N=1280)	163.5 (0.19)	0.32	63 (0.0062)

图 17: 1 维声波方程方程 1 阶格式数据对比

点源一维声波方程

我们考虑如下形式的声波方程:

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = f(t, x)$$

初边值条件均为 0，点源

$$f(t, x) = -9.6 f_0 (0.6 f_0 t - 1) \exp(-8(0.6 f_0 t - 1)^2) \delta(x - x_0)$$

其中 $c = 2$ 表示波速，$f_0 = 20$ 表示频率，我们在下面的数值计算中选取 $t = 0$ 时刻作为初值，计算区域为 $0 \leq x \leq 4$km。

DG 格式结果

我们采用 LLF 通量，并用二阶 TVD 时间离散格式得到结果如下：

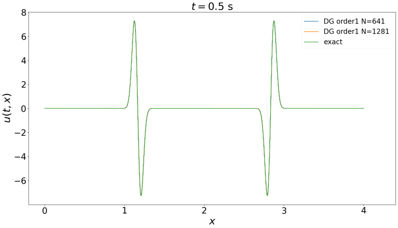

图 18: 传统 DG 方法 1 阶格式结果 $t = 0.5$

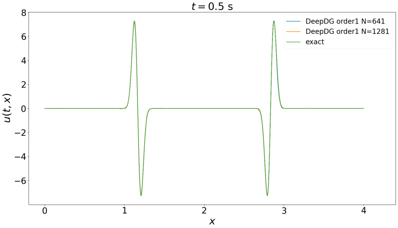

图 19: 深度学习 DG 方法 1 阶格式结果 $t = 0.5$

DG 格式结果

我们采用 LLF 通量，并用二阶 TVD 时间离散格式得到结果如下：

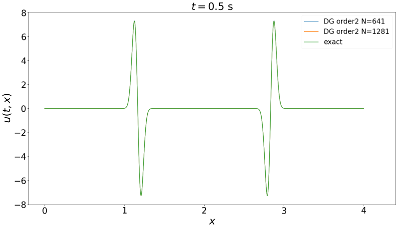

图 20: 传统 DG 方法 2 阶格式结果 $t = 0.5$

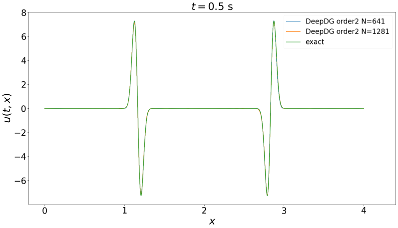

图 21: 深度学习 DG 方法 2 阶格式结果 $t = 0.5$

误差下降曲线

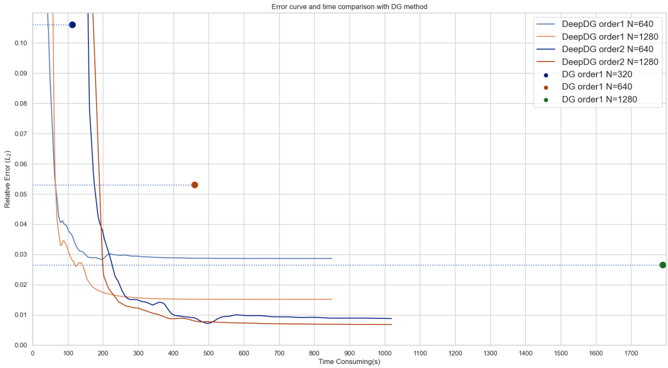

图 22: 深度学习误差下降曲线和传统方法的用时对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=640)	460.5	5.3	10240
深度学习方法	误差达(5.3%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG order1(N=640)	64.9 (0.14)	2.8	70 (0.0068)
DeepDG order1(N=1280)	64.9 (0.14)	1.5	70 (0.0068)
DeepDG order2(N=640)	175.8 (0.38)	0.71	70 (0.0068)
DeepDG order2(N=1280)	190.0 (0.41)	0.67	70 (0.0068)

图 23: 1 维点源声波方程方程 1 阶 DG 格式数据对比

数据对比

传统DG方法	用时(s)	相对误差(%)	波场存储空间(kb)
DG方法(N=1280)	1790.2	2.65	39680
深度学习方法	误差达(2.65%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepDG order1(N=1280)	120.2 (0.067)	1.5	70 (0.0017)
DeepDG order2(N=640)	225.8 (0.12)	0.71	70 (0.0017)
DeepDG order2(N=1280)	199.2 (0.11)	0.67	70 (0.0017)

图 24: 1 维点源声波方程方程 1 阶 DG 格式数据对比

LWC 格式结果

我们采用 2 阶 LWC 格式，得到结果如下：

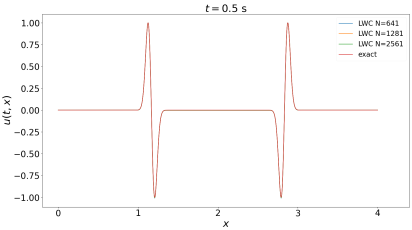

图 25: 传统 LWC 格式结果 $t = 0.5$

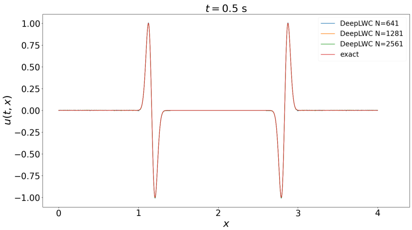

图 26: 深度学习 DG 方法 2 阶格式结果 $t = 0.5$

误差下降曲线

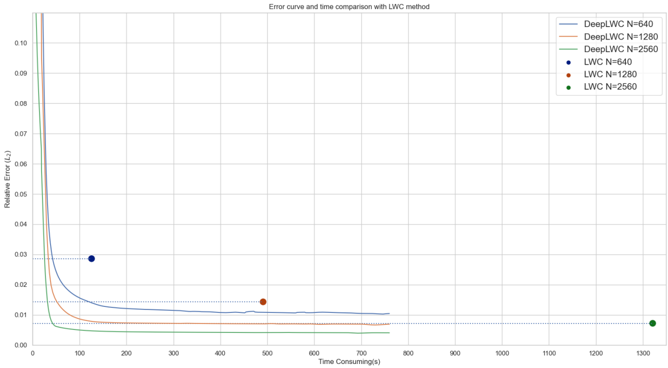

图 27: 深度学习误差下降曲线和传统方法的用时对比

数据对比

传统LWC方法	用时(s)	相对误差(%)	波场存储空间(kb)
LWC方法(N=640)	125.5	2.86	3840
深度学习方法	误差达(2.86%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepLWC (N=640)	42.56 (0.33)	1.0	83 (0.021)
DeepLWC (N=1280)	33.82 (0.26)	0.66	83 (0.021)
DeepLWC (N=2560)	25.46 (0.20)	0.39	83 (0.021)

图 28: 1 维点源声波方程方程 1 阶 LWC 格式数据对比

数据对比

传统LWC方法	用时(s)	相对误差(%)	波场存储空间(kb)
LWC方法(N=1280)	491	1.43	14080
深度学习方法	误差达(1.43%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepLWC (N=640)	120.46 (0.24)	1.0	83 (0.005)
DeepLWC (N=1280)	52.06 (0.10)	0.66	83 (0.005)
DeepLWC (N=2560)	31.92 (0.065)	0.39	83 (0.005)

图 29: 1 维点源声波方程方程 1 阶 LWC 格式数据对比

数据对比

传统LWC方法	用时(s)	相对误差(%)	波场存储空间(kb)
LWC方法(N=2560)	1321	0.72	53760
深度学习方法	误差达(0.72%)用时(s)	收敛时相对误差(%)	波场存储空间(kb)
DeepLWC (N=1280)	270.94 (0.20)	0.66	83 (0.0015)
DeepLWC (N=2560)	42.56 (0.032)	0.39	83 (0.0015)

图 30: 1 维点源声波方程方程 1 阶 LWC 格式数据对比

下一步工作安排

1. 向二维、三维空间尝试

参考文献 I

[Chen et al., 2021] Chen, J., Jin, S., and Lyu, L. (2021).
A deep learning based discontinuous galerkin method for hyperbolic equations with discontinuous solutions and random uncertainties.
arXiv preprint arXiv:2107.01127.

[Raissi et al., 2019] Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019).
Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations.
Journal of Computational physics, 378:686–707.

[Weinan and Yu, 2017] Weinan, E. and Yu, T. (2017).
The deep ritz method: A deep learning-based numerical algorithm for solving variational problems.
Communications in Mathematics and Statistics, 6:1–12.

Thanks!