基于深度学习的间断有限元求解方法

常芸凡[1]

指导老师：杨顶辉教授[1]、贺茜君教授[2]

[1]清华大学数学科学系

[2]北京工商大学数学科学系

2021 年 10 月 17 日

1 进度汇报

1 进度汇报

方程

为了考察一维混合算法的精度，使用混合方法求解如下初值问题：

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - c^2 \dfrac{\partial^2 u}{\partial x^2} = 0 \\ u(x,0) = \cos\left(-\dfrac{2\pi f_0}{c} x\right) \\ \dfrac{\partial u(x,0)}{\partial t} = -2\pi f_0 \sin\left(-\dfrac{2\pi f_0}{c} x\right) \end{cases} \tag{5-15}$$

初值问题(5-15)具有如下精确解：

$$u(t,x) = \cos\left(2\pi f_0 t - \frac{2\pi f_0}{c} x\right) \tag{5-16}$$

其中 c 表示波速，f_0 表示频率，选取 t=0 时刻的值作为初值。计算区域 $0 \le x \le 4$ km，ONAD 方法与 WRKDG 方法的计算区域在 x=2 km 处断开。本例中取 c=4 km/s，频率取 10 Hz，5 Hz。同时为了精确分析空间精度的误差，选取时间步长 $\Delta t = 0.5$ ms 足够小，WRKDG 方法中的加权系数取 $\eta = 0.5$。在试验中发现 η 对误差的影响非常小，几乎可以忽略不计。对于 ONAD 方法，定义其计算区域上的离散范数为：

结果

编辑器 - D:\我的文件\机器学习DG方法\常芸凡\llf-acoustic\RKDG1.m

RKDG1.m | rhs1.m | legd.m | init2.m | eval.m

```
clc,clear
tic
N=10000;ff0=1;c0=1;tend=.5;
lx=0;rx=4;
dx=(rx-lx)/(N);dt=0.20*dx/c0;Nt=floor(tend/dt);
xx=0:dx:(N-1)*dx;

kk=2;
u1=zeros(kk+1,N);v1=u1;p1=v1;
tt=0;
[u1,p1]=init2(N,dx,lx,tt,c0,ff0);

%演化，不加限制器
k=0;
uu1=zeros(kk+1,N);pp1=uu1;
for iit=1:Nt
```

命令行窗口

```
k =

        6249

时间已过 10574.202098 秒。
```

图 1: DG方法空间网格数N=10000

结果

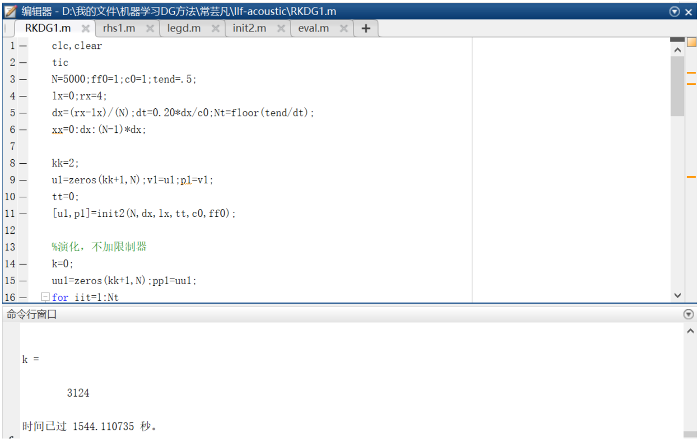


图 2: DG方法空间网格数N=5000

DNN结果

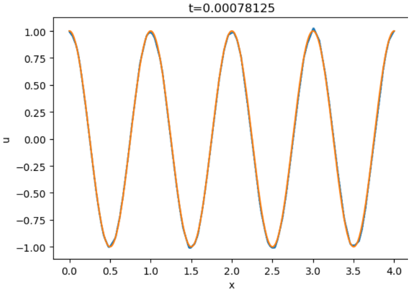
```
i=  1400
error = 0.11951502031692031
loss0 = tensor(0.0128, device='cuda:0')
loss1 = tensor(0.0349, device='cuda:0')
time cost 386.40352034568787 s
```

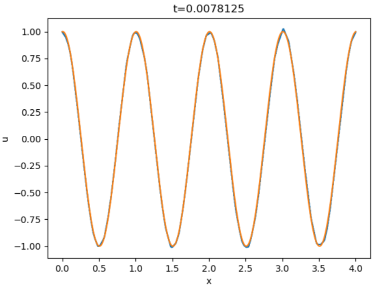
图 3: 0阶DNN方法空间网格数N=5000

```
i=  650
error = 0.1344114712371771
lossu0 = tensor(0.1506, device='cuda:0')
lossu1 = tensor(1.8588, device='cuda:0')
lossp0 = tensor(0.2637, device='cuda:0')
lossp1 = tensor(1.8298, device='cuda:0')
time cost 859.5154640674591 s
```

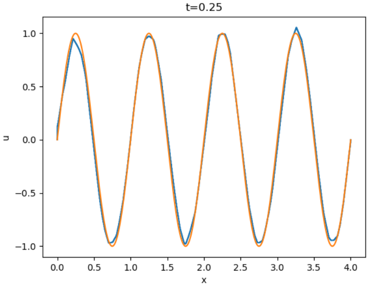
图 4: 1阶DNN方法空间网格数N=5000

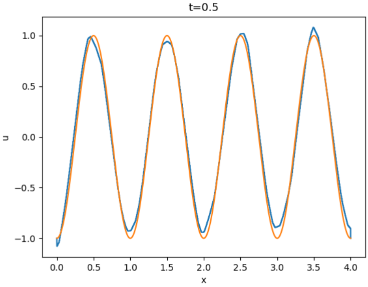
DNN结果0阶


t=0.00078125
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
u
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

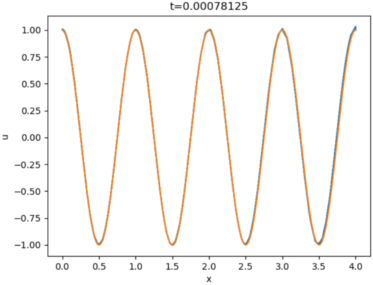

t=0.0078125
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
u
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

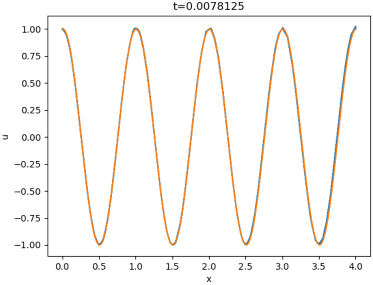
DNN结果0阶


t=0.25
1.0
0.5
0.0
−0.5
−1.0
u
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
x

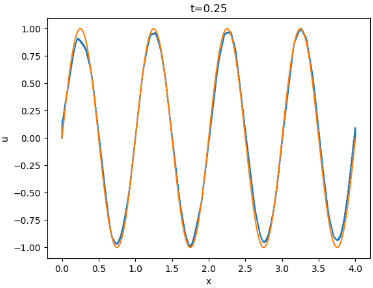

t=0.5
1.0
0.5
0.0
−0.5
−1.0
u
0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0
x

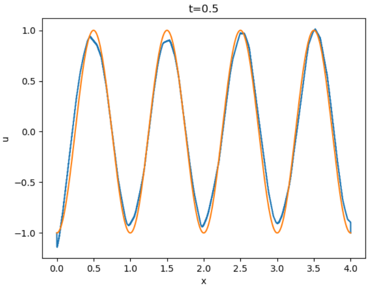
DNN结果1阶


t=0.00078125
u
x
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


t=0.0078125
u
x
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

DNN结果1阶


t=0.25
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
u
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x


t=0.5
1.0
0.5
0.0
−0.5
−1.0
u
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
x

简要的分析和疑问

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

简单的分析和疑问

$$C^{(0)} = C^{(n)}$$

$$C^{(1)} = C^{(0)} + \Delta t L(C^{(0)})$$

$$C^{(2)} = \frac{3}{4} C^{(0)} + \frac{1}{4} C^{(1)} + \frac{1}{4} \Delta t L(C^{(1)})$$

$$C^{(2)} = \frac{1}{3} C^{(0)} + \frac{2}{3} C^{(2)} + \frac{2}{3} \Delta t L(C^{(2)})$$

$$C^{(n+1)} = C^{(3)}$$

Thanks!