差分格式稳定性及数值效应比较实验

林元莘

2024 年 5 月 14 日

摘要

在几周的学习中，我们学习了如何各种差分格式求解对流方程。并且使用傅里叶变换，通过增长因子来讨论数值解的稳定性。本次大作业旨在利用 Python，对各个差分格式求解方法进行对比，通过更改流速 a，通过作图以及误差分析，理解不稳定的格式会对数值解带来灾难性的打击。最后，通过色散、耗散理论，来理解差分格式求解时出现的振荡以及爆破。

关键词：对流方程；差分格式；数值稳定性；色散耗散

1 实验目的

1、了解并求解对流方程的差分格式。

2、进行数值实验，比较各种差分方法在不同波速下的精度以及稳定性。

3、探究差分方法稳定性的内涵。

2 实验问题

考虑如下非光滑边界条件的对流方程：

$$\begin{cases} u_t + au_x = 0, \\ u(0,x) = f(x) = \begin{cases} 1, & x \le 0 \\ 0, & x > 0 \end{cases} \end{cases}$$

其中 $a = 1, 2, 4$，查看 $t = 4.0$ 时的数值结果。

2.1 差分格式及其稳定性

考虑对方程 x 方向与 t 方向做离散差分，时间间隔为 $\tau = 0.08$, 空间间隔为 $h = 0.1$, $\lambda = \frac{\tau}{h}$。可以得到如下差分格式：

2.1.1 迎风格式（Upwind scheme）

$$u_j^{n+1} = u_j^n - a\lambda(u_j^n - u_{j-1}^n)$$

其截断误差为：$O(\tau + h)$, 其增长因子为：

$$G(\tau, k) = (1 - a\lambda) + a\lambda e^{ikh}$$

稳定性条件为：

$$\lambda \leq \frac{1}{a}$$

2.1.2 Beam-Warming 格式

$$u_j^{n+1} = u_j^n - a\lambda(u_j^n - u_{j-1}^n) - \frac{a\lambda}{2}(1 - a\lambda)(u_j^n - 2u_{j-1}^n + u_{j-2}^n)$$

其截断误差为：$O(\tau^2 + h^2)$, 其增长因子为：

$$G(\tau, k) = (1 - a\lambda)(1 - \frac{a\lambda}{2}) + a\lambda(2 - a\lambda)e^{-ikh} - \frac{a\lambda}{2}(1 - a\lambda)e^{-2ikh}$$

稳定性条件为：

$$\lambda \leq \frac{2}{a}$$

2.1.3 Lax-Friedrichs 格式

$$u_j^{n+1} = \frac{1}{2}(u_{j+1}^n + u_{j-1}^n) - \frac{1}{2}a\lambda(u_{j+1}^n - u_{j-1}^n)$$

其截断误差为：$O(\tau + h)$, 其增长因子为：

$$G(\tau, k) = \frac{1 - a\lambda}{2}e^{ikh} + \frac{1 + a\lambda}{2}e^{-ikh}$$

稳定性条件为：

$$\lambda \leq \frac{1}{a}$$

2.1.4 Lax-Wendroff 格式

$$u_j^{n+1} = u_j^n - \frac{1}{2}a\lambda(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}a^2\lambda^2(u_{j+1}^n - 2u_j^n + u_{j-1}^n)$$

其截断误差为: $O(\tau^2 + h^2)$, 其增长因子为:

$$G(\tau, k) = 1 - 2a^2\lambda^2\sin^2(\frac{kh}{2}) - ia\lambda\sin(kh)$$

稳定性条件为:

$$\lambda \leq \frac{1}{a}$$

2.2 差分格式算法

Algorithm 1 求解对流方程的差分格式一般性算法

输入: 解定义域 $[d_1, d_2]$, 解时间 t, 时间步长 τ, 迭代数 $N = \frac{t}{\tau}$, 空间步长 h, 初始函数 $f(x)$.

输出: 解函数向量 $U^N = [u_j]^T$

按照定义域以及空间步长, 以及根据算法增加额外的格点划分 x, 形成迭代初值 $U^0 = f(x)$

For $i \leftarrow 1$ To N

$$\begin{aligned} U^{i+1}[d_1 - (N-i)h : d_2 + (N-i)h] = &\cdots + k_1 U^i[d_1 - (N-i+1)h : d_2 + (N-i+1)h] \\ &+ k_0 U^i[d_1 - (N-i)h : d_2 + (N-i)h] \\ &+ k_{-1} U^i[d_1 - (N-i-1) * h : d_2 + (N-i-1)h] + \ldots \end{aligned}$$

得到结果 U^N

3 实验结果

3.1 实验函数

我们考虑上述方程于 $t = 4.0$ 处的解, 即

$$u(x, 4) = \begin{cases} 1, & x \leq 4a \\ 0, & x > 4a \end{cases} \tag{1}$$

3.2 四种差分格式算法对比

3.2.1 $a = 1$

此时，$a\lambda = 0.8$，均通过稳定性条件，计算效果如图1。

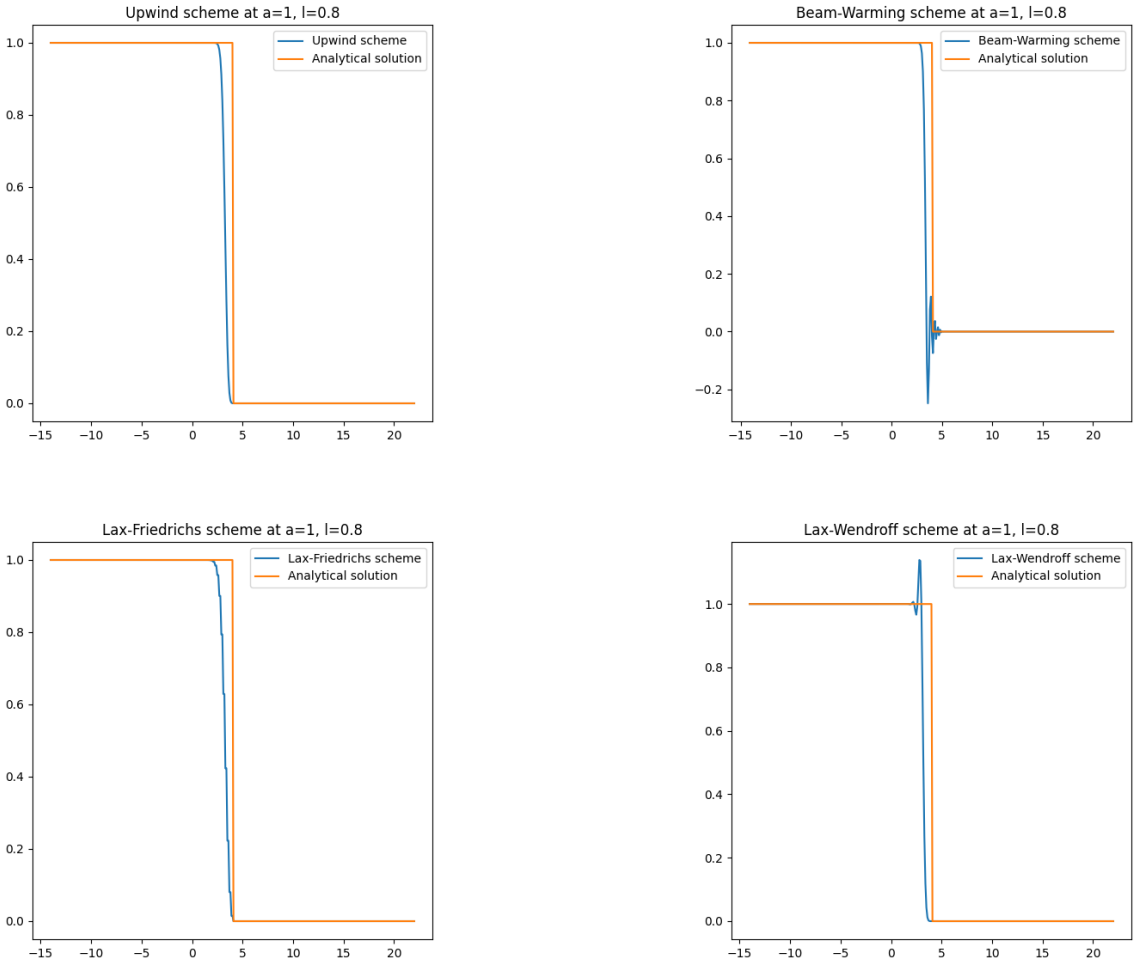


图 1: a=1 时四种差分格式计算效果

3.2.2 $a = 2$

此时，$a\lambda = 1.6$，Beam-Warming 格式通过稳定性条件，而其余三种格式均无法通过稳定性条件，计算效果如图2。

3.2.3 $a = 4$

此时，$a\lambda = 3.2$，均无法通过稳定性条件，计算效果如图??4。

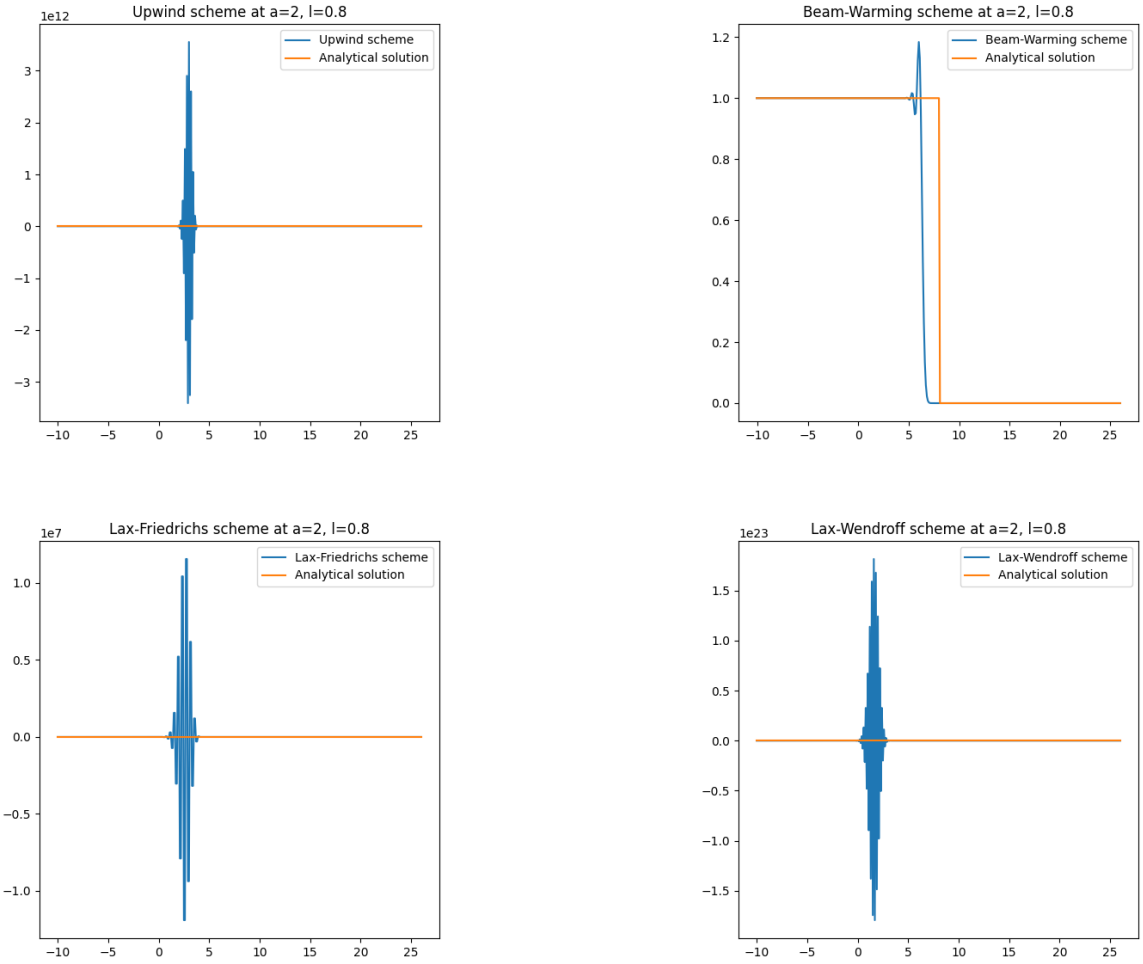
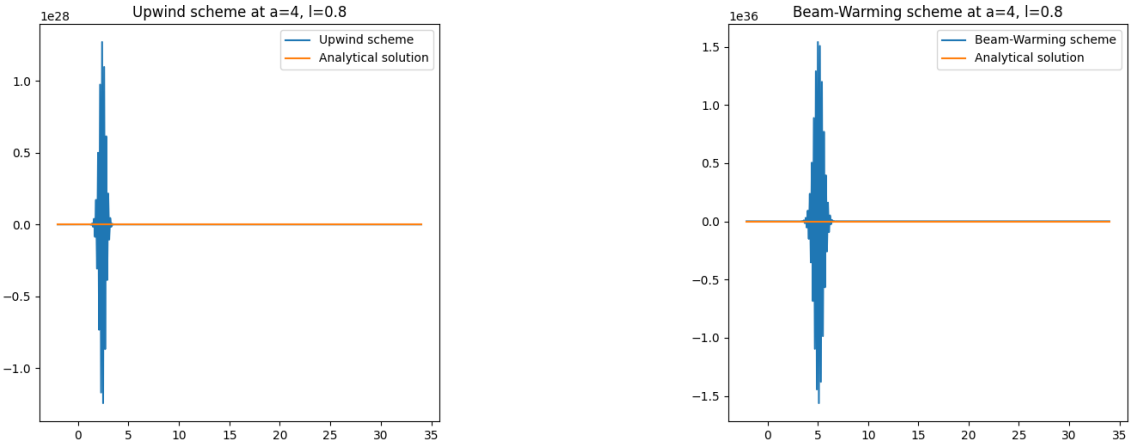


图 2: a=2 时四种差分格式计算效果



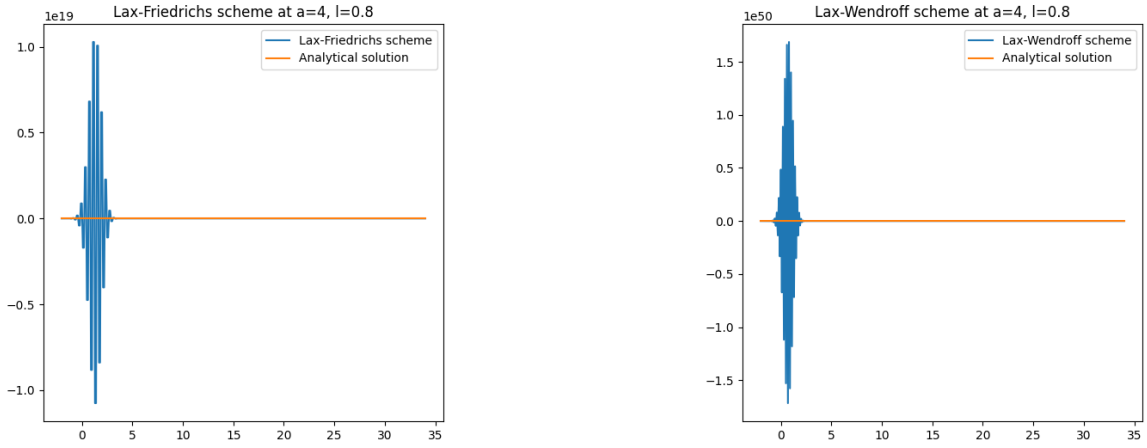


图 4: a=4 时四种差分格式计算效果

3.3 结果稳定性分析

3.3.1 作图结果分析

- $a\lambda = 0.8$ 时，逼近效果还可以，其中迎风格式最为光滑，Lax-Friedrichs 格式较为有锯齿状，Beam-Warming 与 Lax-Wendorff 格式都出现了不同程度的波的震荡。
- $a\lambda = 1.6$ 时，其中迎风格式，Lax-Friedrichs 格式与 Lax-Wendorff 格式出现了高频爆破的现象，而 Beam-Warming 逼近效果仍保持较好情形。
- $a\lambda = 3.2$ 时，其中迎风格式，Beam-Warming,Lax-Friedrichs 格式与 Lax-Wendorff 格式都出现了高频爆破的现象. 其中 Lax-Wendorff 最甚。

3.3.2 数值计算结果

参数 a	迎风格式	Beam-Warming 格式	Lax-Friedrichs 格式	Lax-Wendorff 格式
a=1	0.999867	1.248354	0.985219	0.999998
a=2	3.555137e12	0.99999999	1.190655e7	1.816033e+23
a=4	1.271062e+28	1.562656e+36	1.075263e+19	1.716124e+50

表 1: 四种差分格式的最大绝对误差-a 大小对比表

3.3.3 理论分析

考虑对流方程的色散与耗散现象。假设其解为 $u(x,t) = \hat{u}\exp(i\omega t)\exp(i\xi t)$ 将该形式解代入原方程，得到色散关系，即 $\omega(\xi) = -a\xi$，振幅不随时间衰减且波按照相同速度传播，即方程的解是非耗散的。

考虑差分格式的色散耗散效应，利用 x-MPDE 余项。

迎风格式：

$$\frac{1}{2}a(1-a\lambda)hu_{xx} - \frac{1}{6}a(a\lambda-1)(2a\lambda-1)h^2u_{xxx} + \dots$$

Beam-Warming 格式：

$$\frac{a}{6}(a\lambda-2)(a\lambda-1)h^2u_{xxx} + \dots$$

Lax-Friedrichs 格式：

$$\frac{1}{2\lambda}(1-(a\lambda)^2)hu_{xx} + \frac{1}{3}a(1-(a\lambda)^2)h^2u_{xxx} + \dots$$

Lax-Wendorff 格式：

$$\frac{1}{6}a((a\lambda)^2-1)h^2u_{xxx} + \frac{1}{8}a(a\lambda)((a\lambda)^2-1)h^3u_{xxxx} + \dots$$

由耗散格式、色散格式理论知，迎风格式与 Lax-Friedrichs 格式为一阶耗散为主型，Beam-Warming 与 Lax-Wendorff 格式为一阶色散为主型，

当 $a\lambda = 0.8$ 时，迎风格式与 Lax-Friedrichs 格式为一阶耗散格式与一阶正色散格式，Beam-Warming 是一阶正色散，二阶耗散，波的位移偏右，Lax-Wendorff 格式时一阶逆色散，二阶耗散，波的位移偏左。且通过优势判断知道，色散优势由以下排序递增：迎风格式、Lax-Friedrichs 格式、Lax-Wendorff 格式、Beam-Warming 格式。与图中解的光滑性相符。

当 $a\lambda = 1.6$ 时，迎风格式、Lax-Friedrichs、Lax-Wendorff 格式均为逆耗散格式，格式不稳定，产生爆破！而 Beam-Warming 格式仍为耗散格式，格式稳定。且一阶逆色散，所以此时该格式与 $a\lambda = 0.8$ 时 Lax-Wendorff 格式振荡波的位移相近。

当 $a\lambda = 3.2$ 时，所有格式均为逆耗散格式，格式不稳定！且所有格式均为逆色散格式，所以波均向左移动，振荡均产生于非光滑点左侧。

4 结论

- 对流差分格式的解法多种多样，但稳定性条件各不相同。

- 在稳定性条件不满足时，不光滑趋于会产生很强的爆破现象，导致求解失败。
- 差分格式解的性态与格式的色散、耗散主型有关，关注差分格式的耗散与色散，能更好的理解数值解出现的病态情形。

5 附录

A 附录 A：Python 代码

在这个附录中，我们提供了完成该大作业所用的代码。

nmpde.ipynb

```
import numpy as np
import matplotlib.pyplot as plt
def f(x):
    return (x<=0).astype(int)

def scheme1(d1,d2,a,l):
    N = int((d2-d1)/0.1)
    x = np.linspace(d1-40*0.1,d2,N+40+1)
    A = f(x).astype(float)
    for i in range(40):
        A[i+1:] = (1-a*l)*A[i+1:]+a*l*A[i:-1]
    plt.figure(figsize=(6,6))

    plt.plot(x[40:],A[40:],label='Upwind scheme')
    plt.plot(x[40:],f(x[40:]-4*a),label='Analytical solution')
    plt.legend()
    plt.title(f'Upwind scheme at a={a}, l={l}')
    plt.savefig(f'upwind_{a}.png')
    plt.show()
    print(max(abs(A[40:]-f(x[40:]-4*a))))
    return A[40:]
```

```
def scheme2(d1,d2,a,l):
    N = int((d2-d1)/0.1)
    x = np.linspace(d1-41*0.1,d2,N+40+2)
    A = f(x).astype(float)
    for i in range(40):
        A[i+2:] = (1-3/2*a*l+1/2*(a*l)**2)*A[i+2:]+\
        (2*a*l-(a*l)**2)*A[i+1:-1]- \
        1/2*a*l*(1-a*l)*A[i:-2]
    plt.figure(figsize=(6,6))
    plt.plot(x[40:],A[40:],label = 'Beam-Warming scheme')
    plt.plot(x[40:],f(x[40:]-4*a),label='Analytical solution')
    plt.legend()
    plt.title(f'Beam-Warming scheme at a={a}, l={l}')
    plt.savefig(f'beam_{a}.png')
    plt.show()
    print(max(abs(A[40:]-f(x[40:]-4*a))))
    return A[41:]

def scheme3(d1,d2,a,l):
    N = int((d2-d1)/0.1)
    x = np.linspace(d1-40*0.1,d2+40*0.1,N+80+1)
    A = f(x).astype(float)
    for i in range(40):
        if i == 0:
            A[i+1:-i-1] = (1/2-1/2*a*l)*A[i+2:] \
            + (1/2+1/2*a*l)*A[i:-i-2]
        else:
            A[i+1:-i-1] = (1/2-1/2*a*l)*A[i+2:-i] \
            + (1/2+1/2*a*l)*A[i:-i-2]
    plt.figure(figsize=(6,6))
```

```
    plt.plot(x[40:-40],A[40:-40],label = 'Lax-Friedrichs scheme')
    plt.plot(x[40:-40],f(x[40:-40]-4*a),label='Analytical solution')
    plt.legend()
    plt.title(f'Lax-Friedrichs scheme at a={a}, l={l}')
    plt.savefig(f'laxf_{a}.png')
    plt.show()
    print(max(abs(A[40:-40]-f(x[40:-40]-4*a))))
    return A[40:-40]

def scheme4(d1,d2,a,l):
    N = int((d2-d1)/0.1)
    x = np.linspace(d1-40*0.1,d2+40*0.1,N+80+1)
    A = f(x).astype(float)
    for i in range(40):
        if i == 0:
            A[i+1:-i-1] = (1-(a*l)**2)*A[i+1:-i-1] \
            + (1/2*(a*l)**2-1/2*a*l)*A[i+2:] \
            + (1/2*a*l + 1/2*(a*l)**2)*A[i:-i-2]
        else:
            A[i+1:-i-1] = (1-(a*l)**2)*A[i+1:-i-1] \
            + (1/2*(a*l)**2-1/2*a*l)*A[i+2:-i] \
            + (1/2*a*l + 1/2*(a*l)**2)*A[i:-i-2]
    plt.figure(figsize=(6,6))
    plt.plot(x[40:-40],A[40:-40],label = 'Lax-Wendroff scheme')
    plt.plot(x[40:-40],f(x[40:-40]-4*a),label='Analytical solution')
    plt.legend()
    plt.title(f'Lax-Wendroff scheme at a={a}, l={l}')
    plt.savefig(f'laxw_{a}.png')
    plt.show()
    print(max(abs(A[40:-40]-f(x[40:-40]-4*a))))
    return A[40:-40]
```

```
for a in [1,2,4]:
    node = 4*a
    scheme1(node-18,node+18,a,0.8)
    scheme2(node-18,node+18,a,0.8)
    scheme3(node-18,node+18,a,0.8)
    scheme4(node-18,node+18,a,0.8)
```