

线性常系数双曲型方程上机题目

数 33 赵丰 *

April 6, 2017

1 题目

考虑如下对流问题

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \\ u(x, 0) = u_0(x), \end{cases} \quad (1)$$

分别取以下两种初值:

$$u_0(x) = \begin{cases} 1 - \cos 2\pi x, & x \in [0, 1], \\ 0 & x \notin [0, 1] \end{cases} \quad (2)$$

和

$$u_0(x) = \begin{cases} 1, & x \in [0.4, 0.6], \\ 0 & x \notin [0.4, 0.6] \end{cases} \quad (3)$$

进行计算, 算到 $T = 1$, 对于适中的网格比, 初值取 $x \in [0, 2]$ 进行计算。计算 x_i 时使用如下的积分平均公式:

$$x_i = \frac{1}{h} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_0(x) dx \quad (4)$$

2 解析解

$$u(t, x) = u_0(x - t) \quad (5)$$

*学号:2013012178

3 使用的差分格式

3.1 迎风格式

$$\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{u_j^n - u_{j-1}^n}{h} = 0 \quad (6)$$

3.2 Lax-Friedrichs 格式

$$\frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\tau} + \frac{u_{j+1}^n - u_{j-1}^n}{2h} = 0 \quad (7)$$

3.3 Lax-Wendroff 格式

$$u_j^{n+1} = u_j^n - \frac{a\lambda}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{a^2\lambda^2}{2}(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \quad (8)$$

其中 $\lambda = \frac{\tau}{h}$ 为网格比

3.4 Beam-Warming 格式

$$u_j^{n+1} = u_j^n - \frac{a\lambda}{2}(3u_j^n - 4u_{j-1}^n + u_{j-2}^n) + \frac{a^2\lambda^2}{2}(u_j^n - 2u_{j-1}^n + u_{j-2}^n) \quad (9)$$

3.5 不同差分格式理论性能比较

4 数值结果

用如上列出的数值格式分别求解给定的双曲问题，matlab 代码如下：

```
1 function g=hyperbolic(h,tau,method_name)
2 gl=floor(1/h);
3 gw=floor(0.5/tau);
4 g=zeros(gl,gw);
5 lambda=tau/h;
6 tvd_vector=zeros(gw,1);
7
8 for i=1:gl
9     g(i,1)=integral(@ini_f,h*(i-1),h*i)/h;
10    if(i>1)
11        tvd_vector(1)=tvd_vector(1)+abs((g(i,1)-g(i-1,1)));
```

```

12     end
13 end
14 function y=Riemann_Problem(uL,uR)
15     if(abs(uL-uR)<1e-7)
16         y=uL;
17     elseif(uL>uR)
18         y=uL;%sound speed >=0
19     elseif(uL<uR && uL>=0)
20         y=uL;
21     elseif(uL<uR && uR<=0)
22         y=uR;
23     elseif(uL<=0 && uR>=0)
24         y=0;
25     end
26 end
27 function y=get_a(j)
28     [q1,q2]=set_b1b2(j);
29     y=(q1+q2)/2;
30 end
31 function [b1,b2]=set_b1b2(j)
32     if(j==0)
33         b1=0;
34     else
35         b1=g(j,n-1);
36     end
37     if(j==g1)
38         b2=0;
39     else
40         b2=g(j+1,n-1);
41     end
42 end
43 function y=g_H(j)
44     a=get_a(j);
45     [m1,m2]=set_b1b2(j);
46     y=(f(m1)+f(m2))/2-lambda*a*(f(m2)-f(m1))/2;
47 end
48 function y=g_L(j)
49     [r1,r2]=set_b1b2(j);
50     y=f(r1);
51 end
52 function y=g_flux(j)
53     y=g_L(j)+phi(theta(j))*(g_H(j)-g_L(j));
54 end
55 function y=theta(j)
56     a=get_a(j);
57     [k1,k2]=set_b1b2(j);
58     if(abs(k1-k2)<1e-7)
59         y=0;
60     elseif(a>=0)
61         if(j==0)
62             y=0;
63         else
64             [b3,b4]=set_b1b2(j-1);
65             y=(b4-b3)/(k2-k1);
66         end
67     else
68         if(j==g1)
69             y=0;

```

```

70         else
71             [b3, b4]=set_b1b2(j+1);
72             y=(b4-b3)/(k2-k1);
73         end
74     end
75 end
76 f=@(u) u.^2/2;
77 phi=@(theta) (abs(theta)+theta)./(1+abs(theta));
78 for n=2:gw
79     if (string(method_name)==string('lax_wandroff'))
80         for j=1:gl
81             g(j,n)=g(j,n-1)-lambda*(g_H(j)-g_H(j-1));
82         end
83         elseif (string(method_name)==string('Godunov'))
84             g(1,n)=g(1,n-1)-lambda*(f(Riemann_Problem(g(1,n-1),g(2,n-1)))-f(
Riemann_Problem(0,g(1,n-1))));
85             g(gl,n)=g(gl,n-1)-lambda*(f(Riemann_Problem(g(gl,n-1),0))-f(
Riemann_Problem(g(gl-1,n-1),g(gl,n-1))));
86             for j=2:(gl-1)
87                 g(j,n)=g(j,n-1)-lambda*(f(Riemann_Problem(g(j,n-1),g(j+1,n-1)))-f(
Riemann_Problem(g(j-1,n-1),g(j,n-1))));
88             end
89             elseif (string(method_name)==string('flux_limiter'))
90                 for j=1:gl
91                     g(j,n)=g(j,n-1)-lambda*(g_flux(j)-g_flux(j-1));
92                 end
93                 elseif (string(method_name)==string('lax_friedrich'))
94                     for j=1:gl
95                         g(j,n)=g(j,n-1)-lambda*(g_L(j)-g_L(j-1));
96                     end
97                     elseif (string(method_name)==string('upwind'))
98                         for j=1:gl
99                             [b1, b2]=set_b1b2(j-1);
100                             g(j,n)=g(j,n-1)-lambda*(f(b2)-f(b1));
101                         end
102                     end
103                 end
104             end
105         end
106     end

```

5 结果分析

5.1 光滑初值情形

取 $h = 0.01, \tau = 0.011$ 发现结果完全不对。解释为 $\lambda = \frac{\tau}{h} > 1$ ，不满足 CFL 条件，以下的比较基于 $h = 0.1 \times 10^{-k}, \tau = 0.9 \times 10^{-k}$ 计算到 $T=1$ 的均方误差定义为

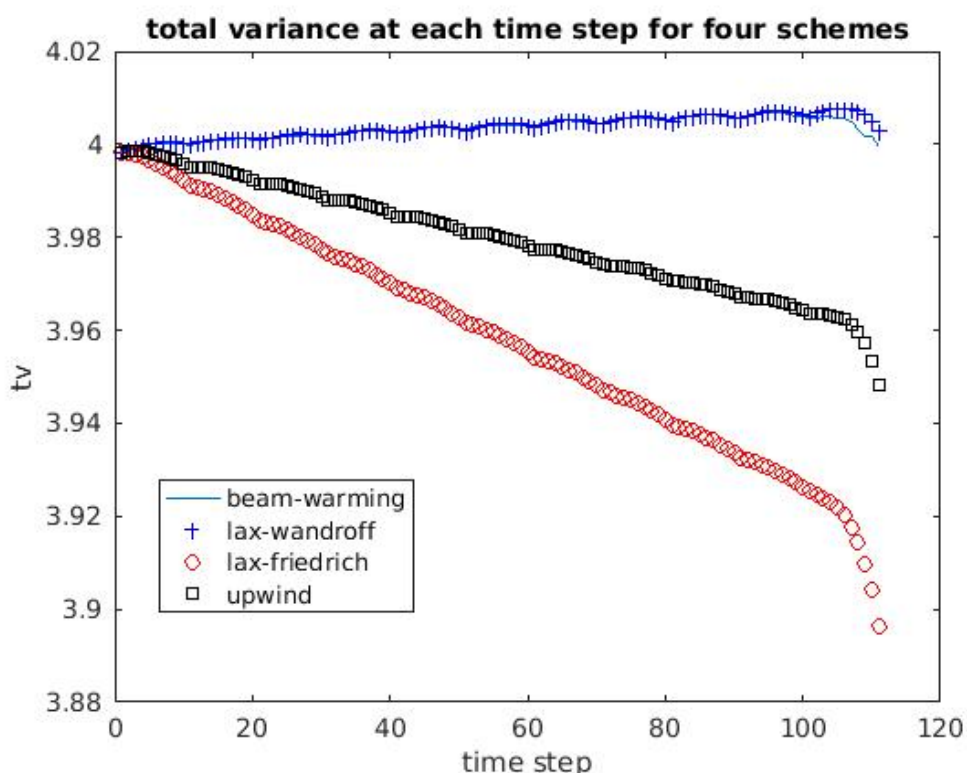
$$\text{MSE} = \sum_{j=2}^N |u_j - u_{j-1}|^2 \quad (10)$$

Table 1: 四种数值格式均方误差比较

h	upwind	lax_friedrich	lax_wandroff	beam_warming
0.01	0.038	0.099	0.0171	0.0151
0.001	0.022	0.028	0.0198	0.0197
1e-4	0.002	0.003	0.002	0.002

由上表可见，二阶格式在步长较大时精度明显高于一阶格式，但当步长逐渐缩小时，二阶格式和一阶格式的差别也在缩小。

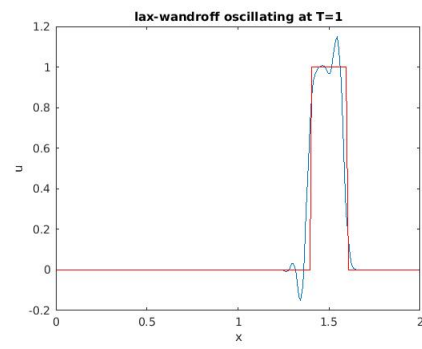
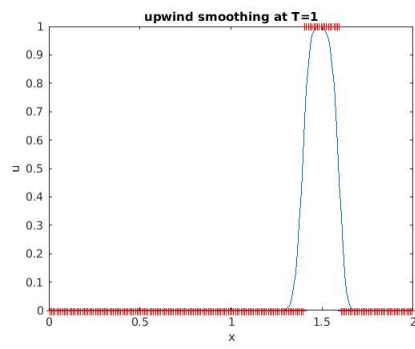
代码在求解过程中也计算了每一步的 TVD 的值，对 $h = 0.01$ 的情形作图得：



通过上图可以看出，两个一阶的格式确实是 TVD 的格式，而两个二阶的格式随时间步长的增大全变差增大，因此不是 TVD 的格式。

5.2 非光滑初值情形

求解由 (2) 给出的初值时，由于初值不连续，无论用什么格式 $T=1$ 时的解误差均比较大，一阶格式给出的解是将初值的不连续性磨光，如下面左图所示；二阶格式给出的解在初值不连续点出现振荡的现象，如下面的右图所示。



6 结论

通过这次上机实验熟悉了使用显式两层差分格式求解双曲问题的思路。