**有限体积法模拟声波方程**

一、理论部分：

无源形式的一阶弹性波方程可写成如下形式：

为了数值计算形式简便，写成矩阵形式如下:

将**A**矩阵对角化，可写成如下形式：

代入可得到如下形式：

令, 有，

得到最终的关于速度和应力的解析形式：

解可以进而表示成如下形式：

问题的解是带有权重的的叠加结果，如此，我们有：

将上述的对角矩阵分解为如下形式：

对应的A矩阵为：

构建迎风格式：

联系到通量概念如下:

但由于这种格式的频散效应，实际中很少使用迎风格式，往往使用Lax-Wendroff形式，如下：

二维的弹性波方程Lax-Wendroff形式:

二维条件下的弹性波传播方程可写为如下形式：

写成矩阵形式为：

其中，

为：

1. 程序代码：

一维情况：

clear;clc;

tic

nx = 800; %number of x point

nt = 1000; % time step

c = 3000; %velocity

density = 2500; %define the density

shear\_modulu = c^2\*density;% to definite shear modulus

dt = 0.001;

dx = 10;

e = 200;%(Gauss)

x0 = dx\*nx/2; %source position

X = (1:nx)\*dx;

fmain = 10;

%% initialization

Q = zeros(2,nx);

Qnew = zeros(2,nx);

A = [0,-shear\_modulu;-1/density,0];

Q\_stress = zeros(nx,nt);

Q\_velocity = zeros(nx,nt);

% set the source term

t = ((1:nt)-30)\*dt;

ft = (2\*(pi\*fmain\*t).^2-1).\*exp(-(pi\*fmain\*t).^2);

%% main calculation procedure

for i=1:nt

Qnew(1,nx/2) = ft(i);

Qnew(2,nx/2) = ft(i);

Q = Qnew;

for j=2:nx-1

dQ1 = Q(:,j+1)-Q(:,j-1);

dQ2 = Q(:,j-1)-2\*Q(:,j)+Q(:,j+1);

Qnew(:,j) = Q(:,j) - dt/(2\*dx)\*A\*dQ1+dt^2/(2\*dx^2)\*(A\*A)\*dQ2;

% dQ1 = Q(:,j)-Q(:,j-1);

% dQ2 = Q(:,j+1)-Q(:,j);

% Qnew(:,j) = Q(:,j)-(dt/dx)\*A\*(dQ1);

end

%absorbed boundary

Qnew(:,1) = Qnew(:,2);

Qnew(:,nx) = Qnew(:,nx-1);

Q\_stress(:,i) = Qnew(1,:);

Q\_velocity(:,i) = Qnew(2,:);

end

pic\_num = 1;

filename = sprintf('FVM\_1D\_fmain=%d.gif',fmain);

for i=10:10:nt

subplot(211)

plot(X,Q\_stress(:,i)/10^6)

str = sprintf('FVM-1D-stress\ntime step=%d',i);

xlabel('X');

ylabel('Stress/MPa','FontWeight','bold');

title(str)

subplot(212)

plot(X,Q\_velocity(:,i))

str = sprintf('FVM-1D-velocity\ntime step=%d',i);

title(str)

xlabel('X');

ylabel('Velocity/(m/s)','FontWeight','bold');

F = getframe(gcf);

I = frame2im(F);

[I,map]=rgb2ind(I,256);

if pic\_num == 1

imwrite(I,map,filename,'gif','Loopcount',inf,'DelayTime',0.2);

else

imwrite(I,map,filename,'gif','WriteMode','append','DelayTime',0.2);

end

pic\_num = pic\_num + 1;

end

figure(2)

subplot(211)

plot(X,Q\_stress(:,500),'linewidth',1.5)

str = sprintf('FVM-1D-stress\ntime step=%d',500);

xlabel('X');

ylabel('Stress','FontWeight','bold');

title(str)

subplot(212)

plot(X,Q\_velocity(:,500),'linewidth',1.5)

str = sprintf('FVM-1D-velocity\ntime step=%d',500);

title(str)

xlabel('X');

ylabel('Velocity','FontWeight','bold');

toc

二维情况：

clear;clc;

tic

%% 初始化参数

nx = 154;

ny = 154;

dx = 10;

dy = 10;

nt = 2000;

dt = 0.001;

v = 3000;

fmain = 20;

e = 200;%(Gauss)

x0 = nx/2;

y0 = x0;

a = dt/dx;

X = (1:nx)\*dx;

Y = (1:ny)\*dy;

epoch = 50;

num\_save = nt/epoch;

count = 1;

t = ((1:nt)-30)\*dt;

ft = (2\*(pi\*fmain\*t).^2-1).\*exp(-(pi\*fmain\*t).^2);

Unew\_p = zeros(nx,ny);

Unew\_vx = zeros(nx,ny);

Unew\_vy = zeros(nx,ny);

U\_p = zeros(nx,ny);

U\_vx = zeros(nx,ny);

U\_vy = zeros(nx,ny);

U\_p\_save = zeros(nx,ny,num\_save);

U\_vx\_save = zeros(nx,ny,num\_save);

U\_vy\_save = zeros(nx,ny,num\_save);

for it=1:nt

U\_p(x0,y0) = ft(it);

U\_vx(x0,y0) = ft(it);

U\_vy(x0,y0) = ft(it);

for II=3:nx-2

for JJ=3:ny-2

dU1 = [U\_p(II,JJ+1);U\_vx(II,JJ+1);U\_vy(II,JJ+1)]+[U\_p(II,JJ-1);U\_vx(II,JJ-1);U\_vy(II,JJ-1)]...

-2\*[U\_p(II,JJ);U\_vx(II,JJ);U\_vy(II,JJ)];

dU2 = [U\_p(II+1,JJ);U\_vx(II+1,JJ);U\_vy(II+1,JJ)]+[U\_p(II-1,JJ);U\_vx(II-1,JJ);U\_vy(II-1,JJ)]...

-2\*[U\_p(II,JJ);U\_vx(II,JJ);U\_vy(II,JJ)];

coeff = -a\*(-v/2-a/2\*v^2);

Um = [U\_p(II,JJ);U\_vx(II,JJ);U\_vy(II,JJ)]+coeff\*dU1+coeff\*dU2;

Unew\_p(II,JJ) = Um(1,1);

Unew\_vx(II,JJ) = Um(2,1);

Unew\_vy(II,JJ) = Um(3,1);

end

%边界条件

i = 1:nx;

j = 1:ny;

%左边界

Unew\_p(1,j) = Unew\_p(3,j);

Unew\_p(2,j) = Unew\_p(3,j);

Unew\_vx(1,j) = Unew\_vx(3,j);

Unew\_vx(2,j) = Unew\_vx(3,j);

Unew\_vy(1,j) = Unew\_vy(3,j);

Unew\_vy(2,j) = Unew\_vy(3,j);

%右边界

Unew\_p(nx,j) = Unew\_p(nx-2,j);

Unew\_p(nx-1,j) = Unew\_p(nx-2,j);

Unew\_vx(nx,j) = Unew\_vx(nx-2,j);

Unew\_vx(nx-1,j) = Unew\_vx(nx-2,j);

Unew\_vy(nx,j) = Unew\_vy(nx-2,j);

Unew\_vy(nx-1,j) = Unew\_vy(nx-2,j);

%上边界

Unew\_p(i,1) = Unew\_p(i,3);

Unew\_p(i,2) = Unew\_p(i,3);

Unew\_vx(i,1) = Unew\_vx(i,3);

Unew\_vx(i,2) = Unew\_vx(i,3);

Unew\_vy(i,1) = Unew\_vy(i,3);

Unew\_vy(i,2) = Unew\_vy(i,3);

%下边界

Unew\_p(i,ny) = Unew\_p(i,ny-2);

Unew\_p(i,ny-1) = Unew\_p(i,ny-2);

Unew\_vx(i,ny) = Unew\_vx(i,ny-2);

Unew\_vx(i,ny-1) = Unew\_vx(i,ny-2);

Unew\_vy(i,ny) = Unew\_vy(i,ny-2);

Unew\_vy(i,ny-1) = Unew\_vy(i,ny-2);

end

if(mod(it,100)==0)

fprintf('time step=%d total=%d\n',it,nt);

end

if(mod(it,epoch)==0)

U\_p\_save(:,:,count) = Unew\_p;

U\_vx\_save(:,:,count) = Unew\_p;

U\_vy\_save(:,:,count) = Unew\_p;

count = count+1;

end

U\_p = Unew\_p;

U\_vx = Unew\_vx;

U\_vy = Unew\_vy;

end

pic\_num = 1;

filename = 'FVM\_2D.gif';

for i=1:num\_save

pcolor(X,Y,U\_p\_save(:,:,i))

str = sprintf('FVM-2D\ntime=%.3f s',i\*50\*dt);

xlabel('X');

ylabel('Y');

title(str);

legend;

shading interp;

axis tight;

axis equal;

F = getframe(gcf);

I = frame2im(F);

[I,map]=rgb2ind(I,256);

if pic\_num == 1

imwrite(I,map,filename,'gif','Loopcount',inf,'DelayTime',0.2);

else

imwrite(I,map,filename,'gif','WriteMode','append','DelayTime',0.2);

end

pic\_num = pic\_num + 1;

end

%压力场p

figure(2)

pcolor(X,Y,U\_p\_save(:,:,num\_save))

str = sprintf('FVM-2D\ntime=%.3f s',num\_save\*50\*dt);

xlabel('X');

ylabel('Y');

title(str);

legend;

shading interp

axis tight;

axis equal;

xlabel('X');

ylabel('Y');

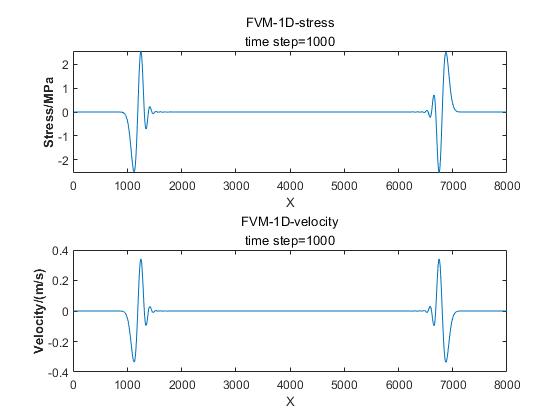
title(str)

toc

1. 结果及问题分析

对于Lax-Wendroff格式，其运算结果相比于迎风格式更加稳定，因为他的公式中含有二阶项，抵消掉了迎风格式具有的误差。

同时，迎风格式在编程过程中步骤繁琐，容易出错。



在模拟二维情况时，利用Lax-Wendroff格式成图效果较好，未使用迎风格式，没有比较两种方法的效果。下图为成像效果。

