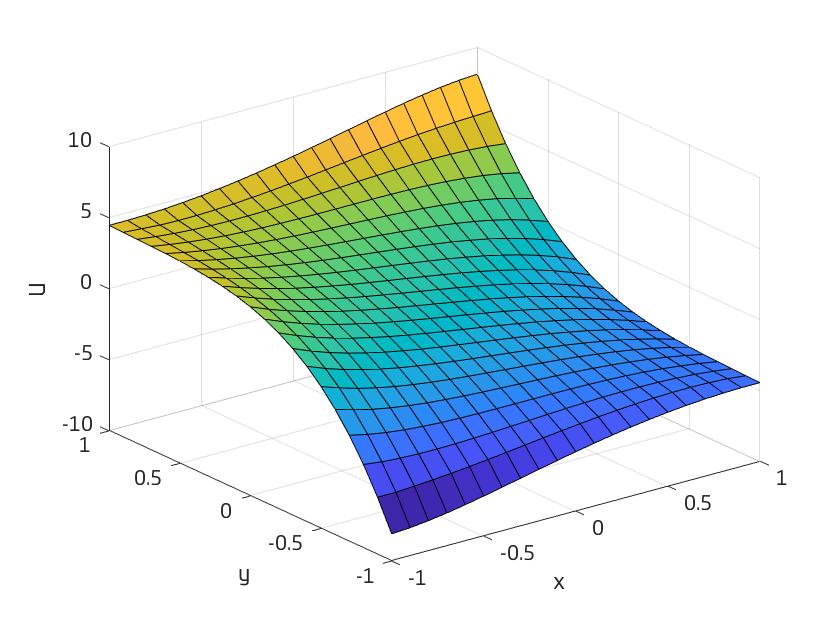
解方程在时间，其中初值条件和边界条件用精确解：

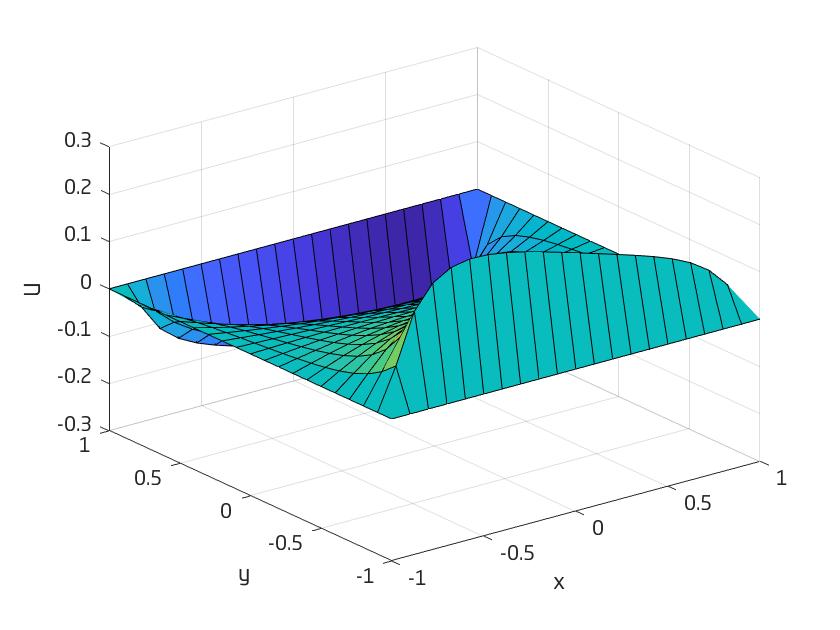
1. 用Peaceman-Rachford方法解
2. 用Mitchell-Fairweather方法解

Peaceman-Rachford方法

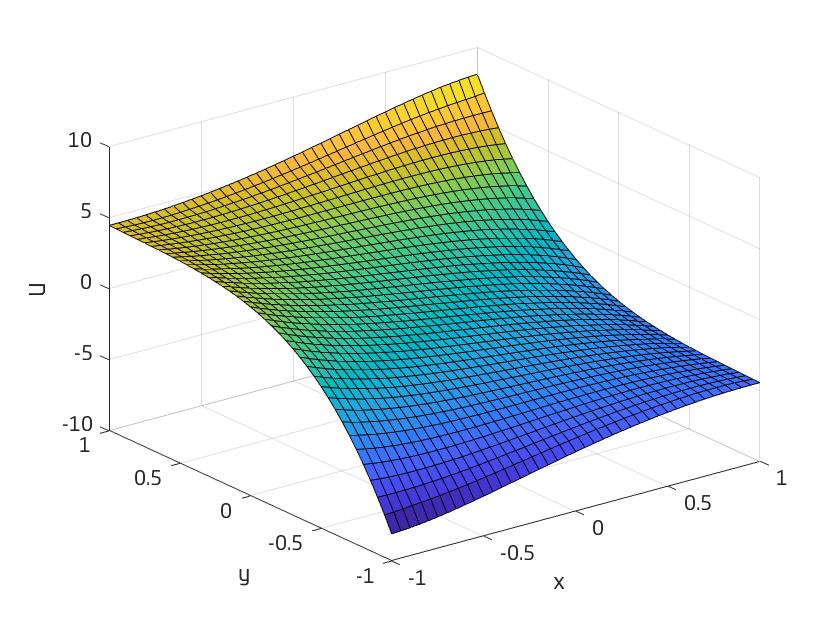
1.取



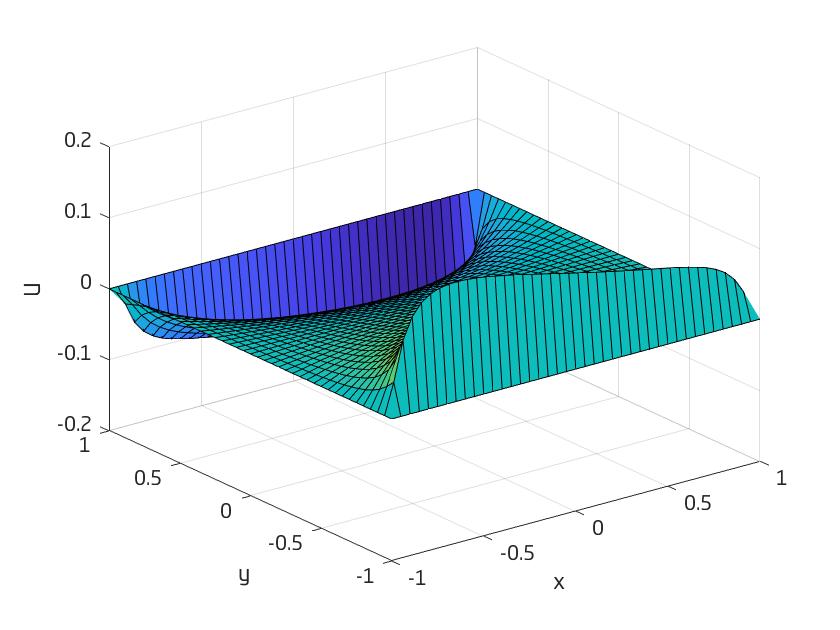
其误差为



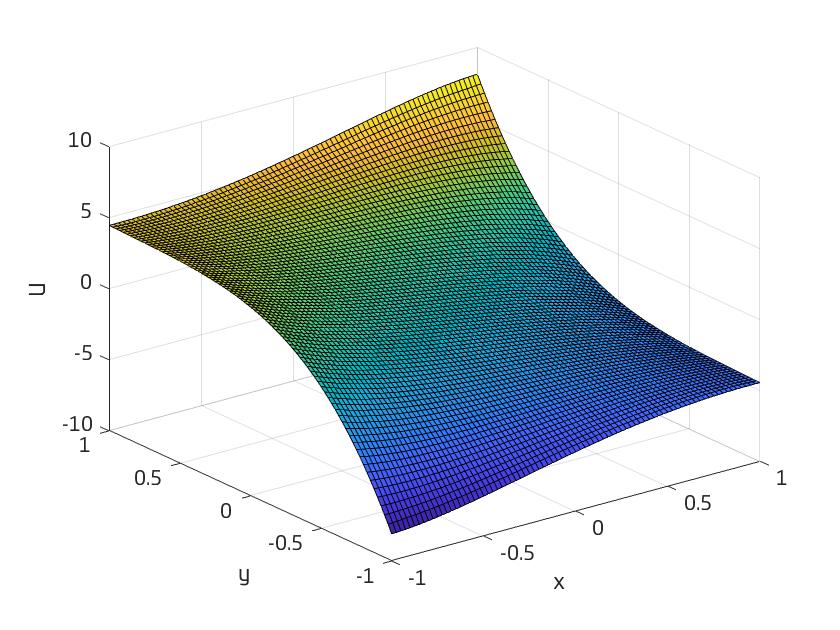
2.取



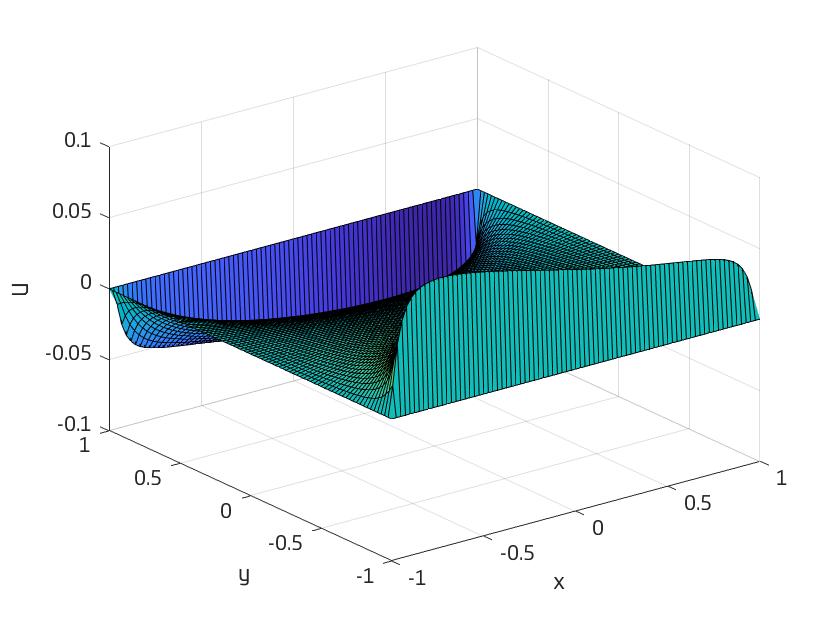
其误差为



3.取

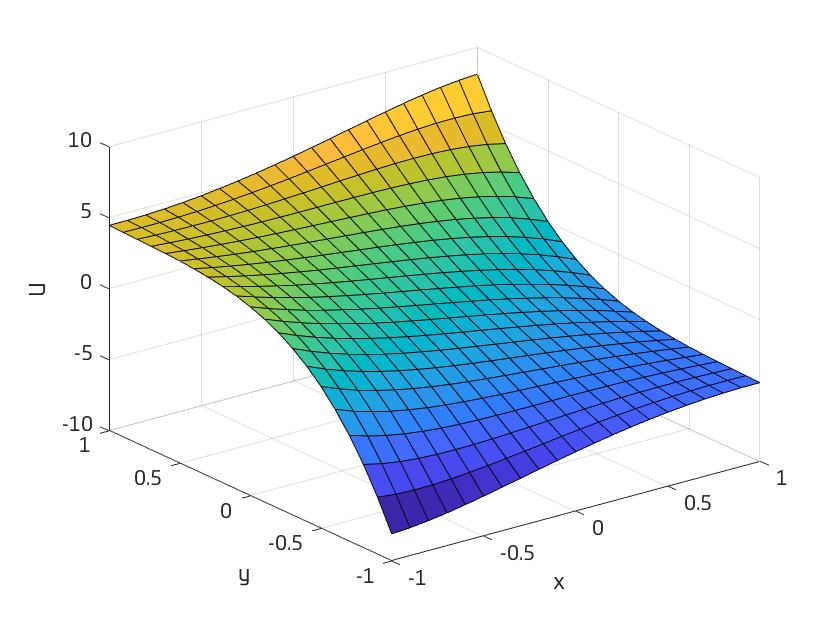


其误差为：

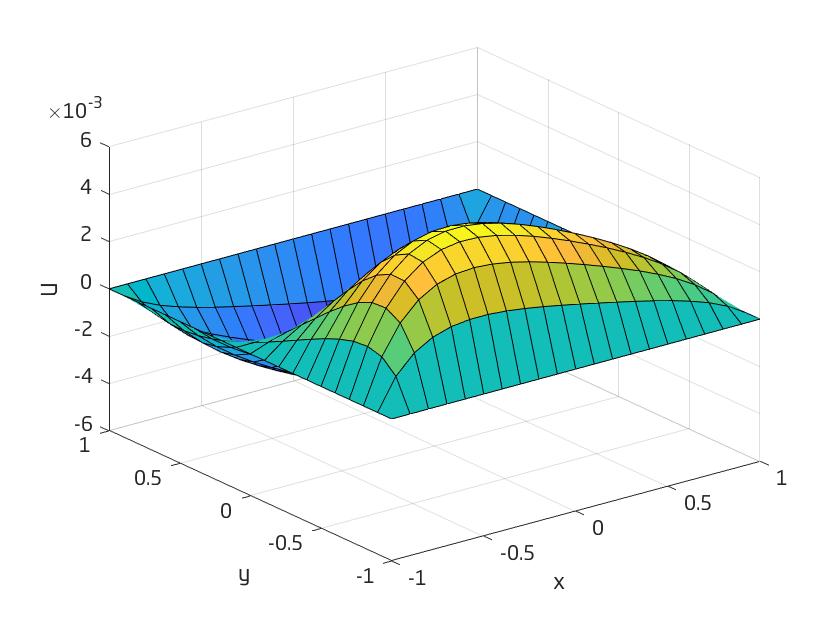


Mitchell-Fairweather方法

取，



其误差为：



通过误差的图像可以看出，随着网格的加细，解的误差在减小，Peaceman-Rachford方法的误差要比Mitchell-Fairweather方法的误差要小。

附录

%%追赶法 Thomas Algorithm 解三对角矩阵

function [ w] = TA( A,v,d )

n=length(d);%n存储将要求解未知量的个数m-1个

p=zeros(1,n+1);

q=zeros(1,n+1);

w=zeros(1,n+2);%w0放在第一个位置，wi放在第i+1个位置

w(1)=v(1);

w(n+2)=v(2);

q(1)=v(1);%为p1，q1赋初值

for i=1:1:n %先求出pq的值

p(i+1)=-1/(A(1)\*p(i)+A(2))\*A(3);

q(i+1)=1/(A(1)\*p(i)+A(2))\*(d(i)-A(1)\*q(i));

end

for j=n:-1:1

w(j+1)=p(j+1)\*w(j+2)+q(j+1);

end

end

Peaceman-Rachford方法的代码为

clear all;

clc;

delt=1/10;%时间间隔

tal=1/10;%x和y的空间间隔

t=0:delt:1;

x=-1:tal:1;

y=-1:tal:1;

U=zeros(length(x),length(y));%初始条件

for i=1:length(x)

for j=1:length(y)

U(i,j)=sin(x(i)-0.5\*y(j))\*cosh(x(i)+y(j));

end

end

for it=2:length(t) %%%%开始循环一层一层的计算

U2=zeros(length(x),length(y));%%%存储n+1/2层的数值

U3=zeros(length(x),length(y));%%%存储n+1层的数

for j=1:length(y)%%%边界条件

U3(1,j)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(1)-0.5\*y(j))\*cosh(x(1)+y(j));

U3(length(x),j)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(length(x))-0.5\*y(j))\*cosh(x(length(x))+y(j));

end

for i=2:length(x)-1

U3(i,1)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(i)-0.5\*y(1))\*cosh(x(i)+y(1));

U3(i,length(y))=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(i)-0.5\*y(length(y)))\*cosh(x(i)+y(length(y)));

end

for j=2:length(y)-1%%%第n+1/2层的边界条件

for i=1:length(x)-1:length(x)

U2(i,j)=1/2\*(U(i,j)+U3(i,j))+1/2\*delt\*(U(i,j+1)-2\*U(i,j)+U(i,j-1)-U2(i,j+1)+2\*U2(i,j)-U2(i,j-1));

end

end

A=ones(1,3);

A(1)= -1/2\*delt; %存储a

A(2)= delt+tal\*tal; %存储b

A(3)=-1/2\*delt; %存储c

d=zeros(1,length(x)-2);

for j=2:1:(length(y)-1)

v=[U2(1,j),U2(length(x),j)]';

for i=1:length(d)

d(i)=tal\*tal\*U(i+1,j)+delt\*(U(i+1,j+1)-2\*U(i+1,j)+U(i+1,j-1));

end

[U2(:,j)]=TA(A,v,d); %%v，d在不同步骤中不同，需要分别求

end

A(1)= -delt; %存储a

A(2)= 2\*delt+tal\*tal; %存储b

A(3)=-delt; %存储c

d1=zeros(1,length(y)-2);

for i=2:1:(length(x)-1)

v=[U3(i,1),U3(i,length(y))]';

for j=1:length(d)

d1(j)=tal\*tal\*U2(i,j+1)+1/2\*delt\*(U2(i+1,j+1)-2\*U2(i,j+1)+U2(i-1,j+1));

end

[U3(i,:)]=TA(A,v,d1); %%v，d在不同步骤中不同，需要分别求

end

U=U3;

end

Ud=zeros(length(x),length(y));%初始条件

for i=1:length(x)

for j=1:length(y)

Ud(i,j)=exp(1.5)\*sin(x(i)-0.5\*y(j))\*cosh(x(i)+y(j));

end

end

Ue=U-Ud;

surf(x,y,Ue);

xlabel('x');

ylabel('y');

zlabel('U');

Mitchell-Fairweather方法的代码为：

clear all;

clc;

delt=1/10;%时间间隔

tal=1/10;%x和y的空间间隔

t=0:delt:1;

x=-1:tal:1;

y=-1:tal:1;

mu=tal/delt/delt;

U=zeros(length(x),length(y));%初始条件

for i=1:length(x)

for j=1:length(y)

U(i,j)=sin(x(i)-0.5\*y(j))\*cosh(x(i)+y(j));

end

end

for it=2:length(t) %%%%开始循环一层一层的计算

U2=zeros(length(x),length(y));%%%存储n+1/2层的数值

U3=zeros(length(x),length(y));%%%存储n+1层的数

for j=1:length(y)%%%边界条件

U3(1,j)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(1)-0.5\*y(j))\*cosh(x(1)+y(j));

U3(length(x),j)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(length(x))-0.5\*y(j))\*cosh(x(length(x))+y(j));

end

for i=2:length(x)-1

U3(i,1)=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(i)-0.5\*y(1))\*cosh(x(i)+y(1));

U3(i,length(y))=exp(1.5\*(t(1)+(it-1)\*delt))\*sin(x(i)-0.5\*y(length(y)))\*cosh(x(i)+y(length(y)));

end

for j=2:length(y)-1%%%第n+1/2层的边界条件

for i=1:length(x)-1:length(x)

a=(mu+1/6)\*( U(i,j)+1/2\*(2\*mu+1/6)\*(U(i,j+1)-2\*U(i,j)+U(i,j-1)) );

b=(mu-1/6)\*( U3(i,j)-1/2\*(2\*mu-1/6)\*(U3(i,j+1)-2\*U3(i,j)+U3(i,j-1)) );

U2(i,j)=1/(2\*mu)\*(a+b);

end

end

A=ones(1,3);

A(1)= -1/2\*(mu-1/6); %存储a

A(2)= 1+(mu-1/6); %存储b

A(3)=-1/2\*(mu-1/6); %存储c

d=zeros(1,length(x)-2);

for j=2:1:(length(y)-1)

v=[U2(1,j),U2(length(x),j)]';

for i=1:length(d)

d(i)=U(i+1,j)+1/2\*(2\*mu+1/6)\*( U(i+1,j-1)-2\*U(i+1,j)+U(i+1,j+1) );

end

[U2(:,j)]=TA(A,v,d); %%v，d在不同步骤中不同，需要分别求

end

A(1)= -1/2\*(2\*mu-1/6); %存储a

A(2)= 1+(2\*mu-1/6); %存储b

A(3)=-1/2\*(2\*mu-1/6); %存储c

d1=zeros(1,length(y)-2);

for i=2:1:(length(x)-1)

v=[U3(i,1),U3(i,length(y))]';

for j=1:length(d)

d1(j)=U2(i,j+1)+1/2\*(mu+1/6)\*( U2(i+1,j+1)-2\*U2(i,j+1)+U2(i-1,j+1) );

end

[U3(i,:)]=TA(A,v,d1); %%v，d在不同步骤中不同，需要分别求

end

U=U3;

end

Ud=zeros(length(x),length(y));%初始条件

for i=1:length(x)

for j=1:length(y)

Ud(i,j)=exp(1.5)\*sin(x(i)-0.5\*y(j))\*cosh(x(i)+y(j));

end

end

Ue=U-Ud;

surf(x,y,Ue);

xlabel('x');

ylabel('y');

zlabel('U');