

# 课题组组会-练习 1

王程

2023 年 10 月 4 日

## 一 练习及结果

1. 对带源项的扩散方程  $u_t = u_{xx} + \pi^2 \sin(\pi x), x \in [0, 1], t \geq 0$  满足以下初始条件  $u(x, 0) = x^2 - x$ , 及边界条件  $u(0, t) = u(1, t) = 0$ 。

(1) 求该方程的解析稳态解。

(2) 使用 FHOS 引入辅助变量, 将上述方程改写成双曲方程组, 考虑均匀网格 (单元数: 8, 16, 32, 64, ……), 时间离散方法使用显式欧拉格式, 空间离散使用 DG(P0)+DGP(0) 格式, 求解稳态解, 并与 (1) 中的解析解进行对比, 测试原始变量  $u$  和它在  $x$  方向的导数的空间精度。

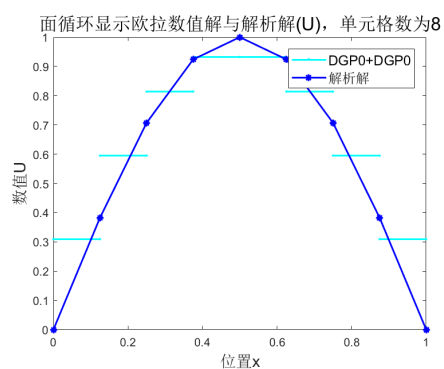
(3) 将时间离散格式改为 BDF1, 使用 Jacobi 迭代法重新对以上方法进行求解, 并与显式方法进行对比。

解:

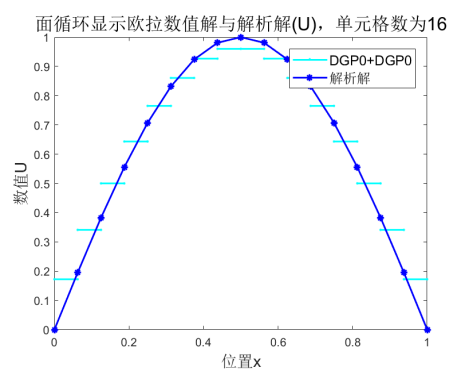
(1) 方程的解析稳态解为:  $u(x, t) = \sin(\pi x)$ 。

(2) 本题仅考虑物理时间达到稳态时的解, 时间离散方法使用显式欧拉格式, 空间离散使用 DG(P0)+DG(P0) 格式, 采用面循环计算 RHS, 最终得到变形后双曲方程组的稳态解。这里仅展示网格数为 8, 16, 32, 64, CFL=0.01 的稳态数值解与解析解的比较图, 并给出原始变量  $u$  和它在  $x$  方向的导数的空间精度比较图。

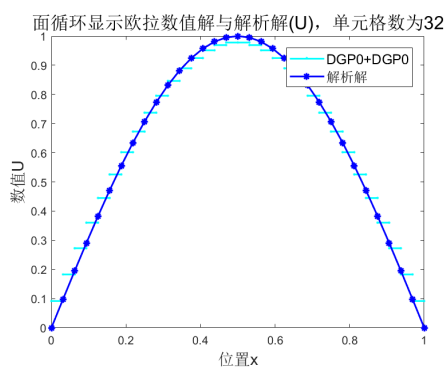
比较  $u$  稳态数值解和解析解：



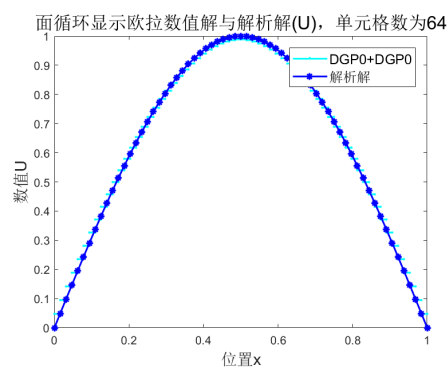
(a) 面循环 U8



(b) 面循环 U16



(c) 面循环 U32



(d) 面循环 U64

图 1: 显式欧拉面循环求得的  $u$  稳态解与解析解对比

$u$  的空间精度：

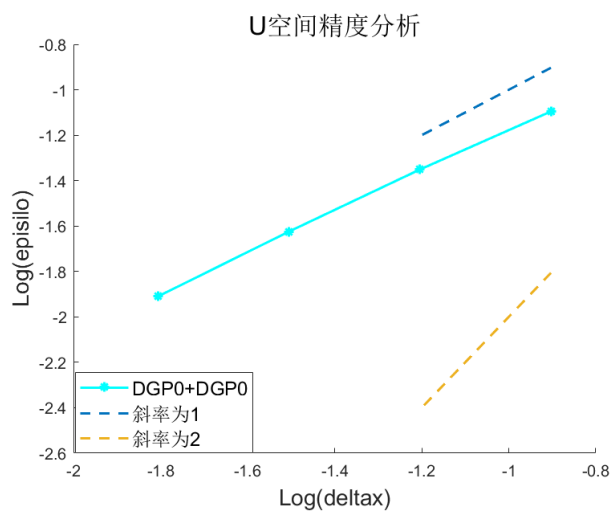
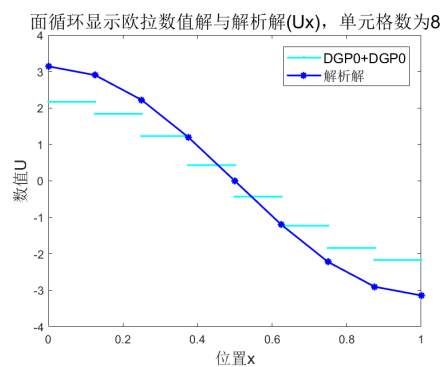
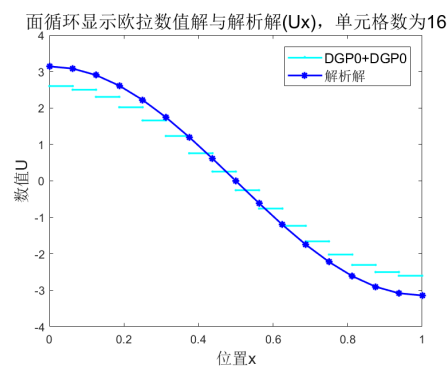


图 2: 面循环下  $u$  空间精度

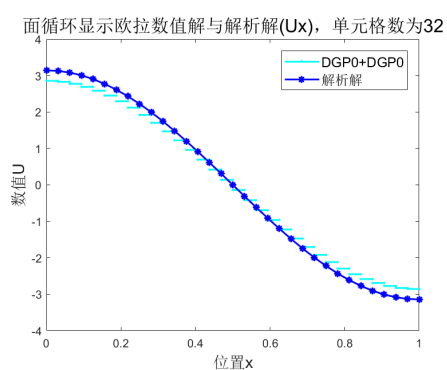
比较  $u_x$  稳态数值解和解析解：



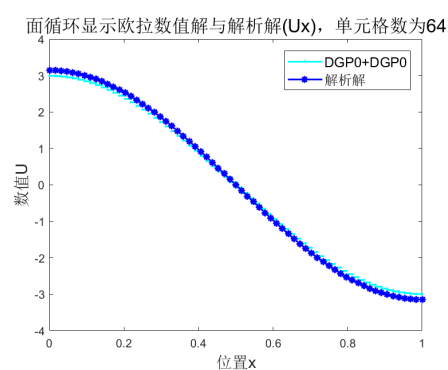
(a) 面循环  $U_x8$



(b) 面循环  $U_{x16}$



(c) 面循环  $U_{x32}$



(d) 面循环  $U_{x64}$

图 3: 显式欧拉面循环求得的  $u_x$  稳态解与解析解对比

$u_x$  的空间精度：

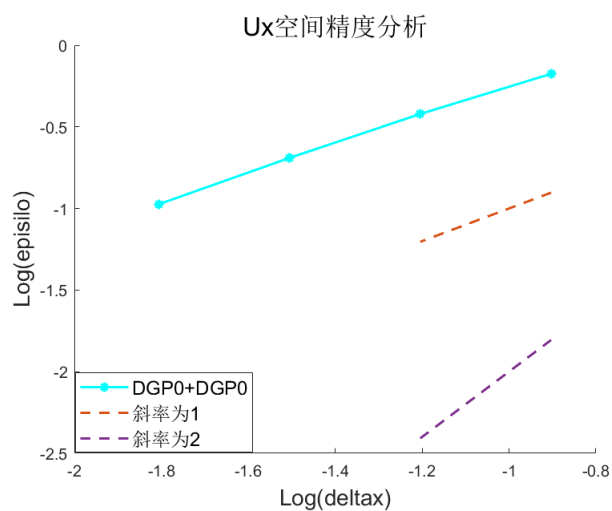


图 4: 面循环下  $u_x$  空间精度

(3) 将时间离散格式改为 BDF1，使用 Jacobi 迭代法重新对该方程的  $DG(P0)+DG(P0)$  格式进行求解，采用面循环计算 RHS，最终得到变形后双曲方程组的稳态解。这里仅展示网格数为 8,16,32,64,CFL=0.01 的稳态数值解与解析解的比较图，并给出原始变量  $u$  和  $u_x$  的空间精度图。

比较  $u$  稳态数值解和解析解：

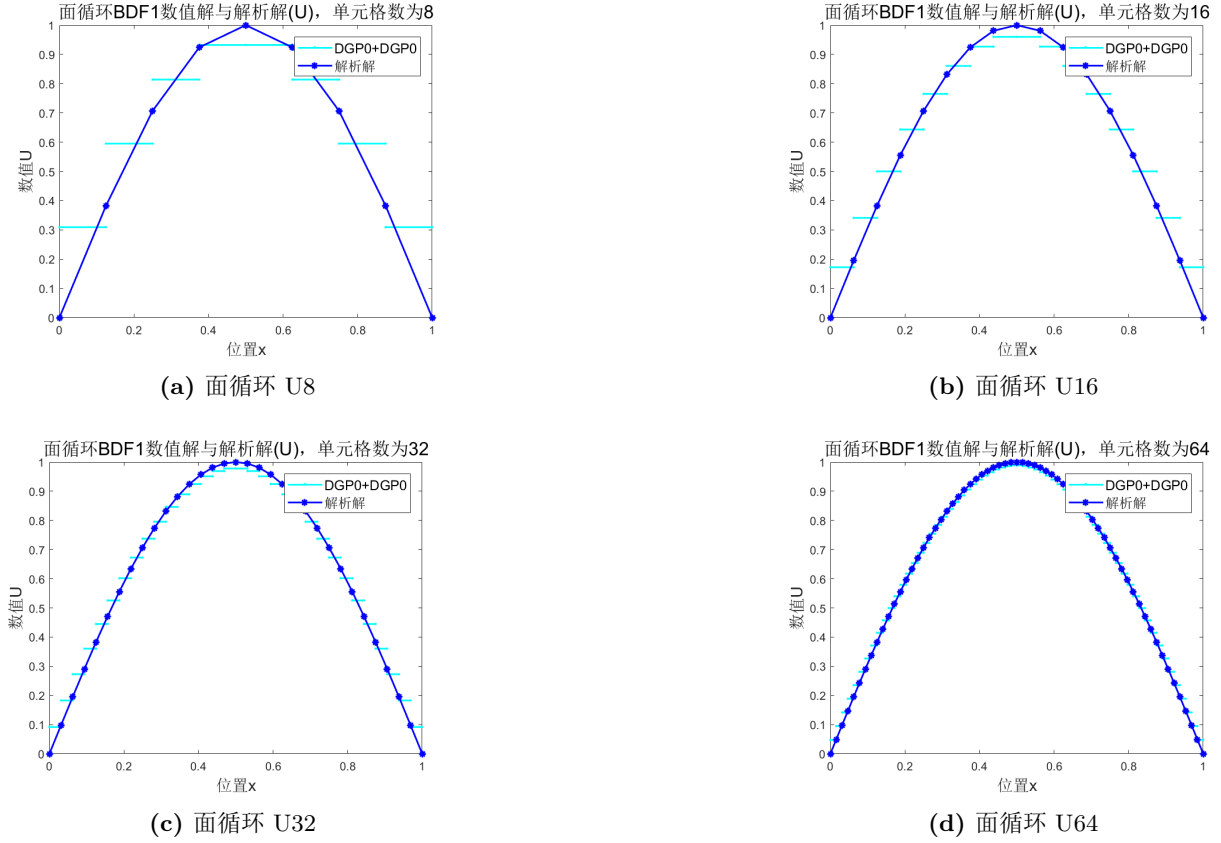


图 5: BDF1 面循环求得的  $u$  稳态解与解析解对比

$u$  的空间精度：

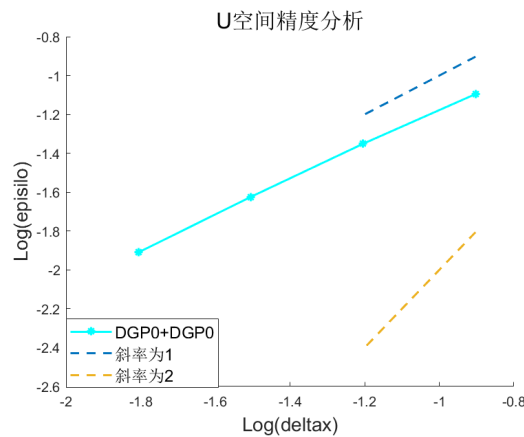
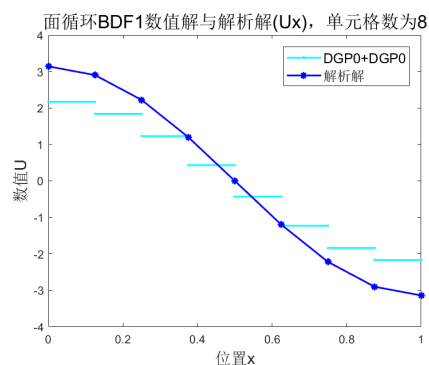
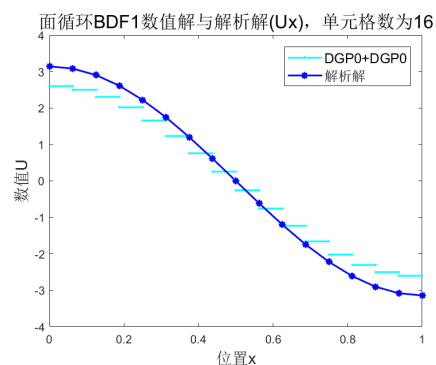


图 6: BDF1 面循环下  $u$  空间精度

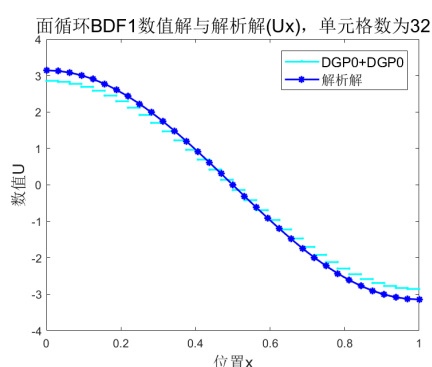
比较  $u_x$  稳态数值解和解析解：



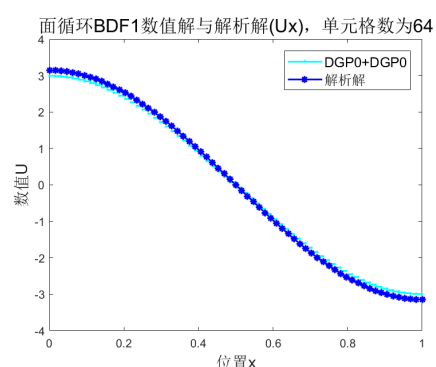
(a) 面循环  $U_{x8}$



(b) 面循环  $U_{x16}$



(c) 面循环  $U_{x32}$



(d) 面循环  $U_{x64}$

图 7: BDF1 面循环求得的  $u_x$  稳态解与解析解对比

$u_x$  的空间精度：

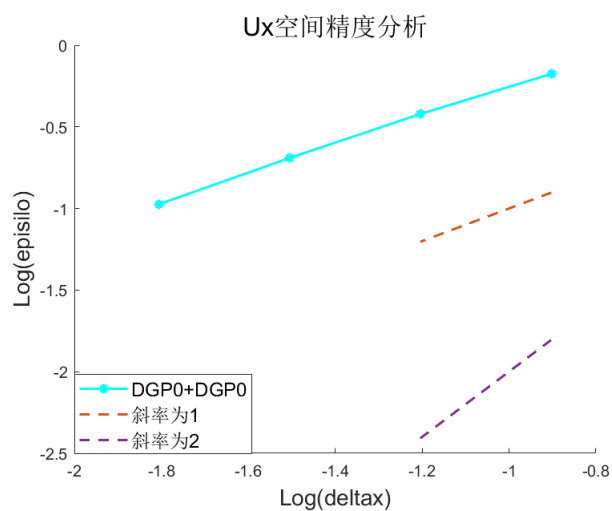


图 8: BDF1 面循环下  $u_x$  空间精度

## 二 附录 (代码)

### BDF1 main

```
1  clc
2  clear all
3  close all
4  %Unit=64;CFL=0.01;endtau=1;endx=1;deltax=endx/Unit;numberx=endx/deltax+1;
5  tol=10^(-10);
6  nu=1;Lr=1/(2*pi);Tr=Lr^2/nu;
7  abslambda=sqrt(nu/Tr);deltatau=CFL*deltax/abslambda;%伪时间变量
8  B1=1;
9  C=[B1,0;0,B1/deltax];Mtau=[deltax,0;0,1/deltax];%此为推导出的 U=CV 中的 C
10 A=[abslambda,0;0,abslambda];
11 Uexasolution=zeros(2,numberx);
12 %后处理预设的量
13 Unumsolution1=zeros(1,2);
14 Unumsolution2=zeros(2,numberx-1);
15 %计算空间精度所预设的量
16 Acc=zeros(3,4);a1=[1/8,1/16,1/32,1/64];a2=[1/8,1/16];
17 %x=0;
18 for k=1:numberx
19     Uexasolution(1,k)=sin(pi*x);
20     Uexasolution(2,k)=pi*cos(pi*x);
21     x=x+deltax;
22 end
23
24 [UDGP0plusDGP0,endtau0plus0]=subDGP0plusDGP0(Unit,CFL,endtau);
25
26 %figure
27 x=0*deltax:deltax:1*deltax;
28 Unumsolution1(1,1)=UDGP0plusDGP0(1,1);Unumsolution1(1,2)=UDGP0plusDGP0
    (1,1);
29 plot(x,Unumsolution1,'-c.','linewidth',1.5);hold on
30 H1=plot(x,Unumsolution1,'-c.','linewidth',1.5);hold on
31 for i=2:numberx-1
32     x=(i-1)*deltax:deltax:i*deltax;
33     Unumsolution1(1,1)=UDGP0plusDGP0(1,i);Unumsolution1(1,2)=UDGP0plusDGP0
        (1,i);
34     plot(x,Unumsolution1,'-c.','linewidth',1.5)
35 end
```

```

36
37 %plot the exact
38 y=0:deltax:endx;
39 plot(y,Uexasolution(1,:),'-b*','linewidth',1.5)
40 H2=plot(y,Uexasolution(1,:),'-b*','linewidth',1.5);hold on
41 lgd=legend([H1,H2],'DGP0+DGP0','解析解');
42 lgd.FontSize=12;
43 xlabel('位置x','fontsize',14)
44 ylabel('数值U','fontsize',14)
45 title('面循环BDF1数值解与解析解(U), 单元格数为64','fontsize',16)
46 hold off
47
48 %plot the ux
49 %DGP0plusDGP0
50 figure
51 x=0*deltax:deltax:1*deltax;
52 Unumsolution1(1,1)=UDGP0plusDGP0(2,1);Unumsolution1(1,2)=UDGP0plusDGP0
    (2,1);
53 plot(x,Unumsolution1,'-c.','linewidth',1.5);hold on
54 H1=plot(x,Unumsolution1,'-c.','linewidth',1.5);hold on
55 for i=2:numberx-1
56 x=(i-1)*deltax:deltax:i*deltax;
57 Unumsolution1(1,1)=UDGP0plusDGP0(2,i);Unumsolution1(1,2)=UDGP0plusDGP0
    (2,i);
58 plot(x,Unumsolution1,'-c.','linewidth',1.5)
59 end
60
61 %exact
62 y=0:deltax:endx;
63 plot(y,Uexasolution(2,:),'-b*','linewidth',1.5)
64 H2=plot(y,Uexasolution(2,:),'-b*','linewidth',1.5);hold on
65 lgd=legend([H1,H2],'DGP0+DGP0','解析解');
66 lgd.FontSize=12;
67 xlabel('位置x','fontsize',14)
68 ylabel('数值U','fontsize',14)
69 title('面循环BDF1数值解与解析解(Ux), 单元格数为64','fontsize',16)
70 hold off
71
72 %determine the accuracy of space U
73 Acc(1,1)=AccuracyU(8,subDGP0plusDGP0(8,CFL,endtau));

```

```

74 Acc(1,2)=AccuracyU(16,subDGP0plusDGP0(16,CFL,endtau));
75 Acc(1,3)=AccuracyU(32,subDGP0plusDGP0(32,CFL,endtau));
76 Acc(1,4)=AccuracyU(64,subDGP0plusDGP0(64,CFL,endtau));
77
78 figure
79 hold on
80 plot(log10(a1),log10(Acc(1,:)),'-c*','linewidth',1.5)
81 H1=plot(log10(a1),log10(Acc(1,:)),'-c*','linewidth',1.5);
82
83 H2=plot(log10(a2),1*log10(a2),'--','linewidth',1.5);
84 plot(log10(a2),2*log10(a2),'--','linewidth',1.5)
85 H3=plot(log10(a2),2*log10(a2),'--','linewidth',1.5);
86 lgd=legend([H1,H2,H3],'DGP0+DGP0','斜率为1','斜率为2');
87 lgd.FontSize=12;
88 xlabel('Log(deltax)','fontsize',14)
89 ylabel('Log(episilo)','fontsize',14)
90 title('U空间精度分析','fontsize',16)
91
92 %determine the accuracy of space Ux
93 %DGP0
94 Acc(1,1)=AccuracyUx(8,subDGP0plusDGP0(8,CFL,endtau));
95 Acc(1,2)=AccuracyUx(16,subDGP0plusDGP0(16,CFL,endtau));
96 Acc(1,3)=AccuracyUx(32,subDGP0plusDGP0(32,CFL,endtau));
97 Acc(1,4)=AccuracyUx(64,subDGP0plusDGP0(64,CFL,endtau));
98
99 figure
100 hold on
101 plot(log10(a1),log10(Acc(1,:)),'-c*','linewidth',1.5)
102 H1=plot(log10(a1),log10(Acc(1,:)),'-c*','linewidth',1.5);
103
104 plot(log10(a2),1*log10(a2),'--','linewidth',1.5)
105 H2=plot(log10(a2),1*log10(a2),'--','linewidth',1.5);
106 plot(log10(a2),2*log10(a2),'--','linewidth',1.5)
107 H3=plot(log10(a2),2*log10(a2),'--','linewidth',1.5);
108 lgd=legend([H1,H2,H3],'DGP0+DGP0','斜率为1','斜率为2');
109 lgd.FontSize=12;
110 xlabel('Log(deltax)','fontsize',14)
111 ylabel('Log(episilo)','fontsize',14)
112 title('Ux空间精度分析','fontsize',16)

```



## BDF1 function DG(P0)+DG(P0)

```
1 function [ Unumsolution ,n]=subDGP0plusDGP0( Unit ,CFL, endtau)
2 %Some basic paramater
3 endx=1;deltax=endx/Unit ;numberx=endx/deltax+1;
4 tol=10^(-10);
5 nu=1;Lr=1/(2*pi);Tr=Lr^2/nu;
6 abslambda=sqrt(nu/Tr);deltatau=CFL*deltax/abslambda;%伪时间变量
7 B1=1;
8 C=[B1,0;0,B1/deltax];Mtau=[deltax,0;0,1/deltax];%此为推导出的 U=CV 中的 C
9 A=[abslambda,0;0,abslambda];
10 R=zeros(2*Unit,1);
11 Rd=zeros(2,numberx-1);
12 Rb=zeros(2,numberx-1);
13 Fn=zeros(2,numberx);
14
15 %构建 LHS
16 %Mtau/deltatau
17 LHS1=sparse(1:2:2*Unit-1,1:2:2*Unit-1,deltax/deltatau,2*Unit,2*Unit);
18 LHS1=LHS1+sparse(2:2:2*Unit,2:2:2*Unit,1/(deltax*deltatau),2*Unit,2*Unit
19 );
20 %Rdomain
21 LHS2=-sparse(2:2:2*Unit,2:2:2*Unit,-1/(Tr*deltax),2*Unit,2*Unit);
22 %Rboundary
23 LHS3=zeros(2*Unit,2*Unit);
24 for iface=2:numberx-1
25     ieL=iface-1;
26     ieR=iface;
27     %diag
28     LHS3(2*ieL-1:2*ieL,2*ieL-1:2*ieL)=LHS3(2*ieL-1:2*ieL,2*ieL-1:2*ieL)+C'*[
29         abslambda/2,-nu/(2*deltax);-1/(2*Tr),abslambda/(2*deltax)];
30     LHS3(2*ieR-1:2*ieR,2*ieR-1:2*ieR)=LHS3(2*ieR-1:2*ieR,2*ieR-1:2*ieR)-C
31         '*[-abslambda/2,-nu/(2*deltax);-1/(2*Tr),-abslambda/(2*deltax)];
32     %upper
33     LHS3(2*ieL-1:2*ieL,2*ieR-1:2*ieR)=LHS3(2*ieL-1:2*ieL,2*ieR-1:2*ieR)+C
34         '*[-abslambda/2,-nu/(2*deltax);-1/(2*Tr),-abslambda/(2*deltax)];
35     %lower
36     LHS3(2*ieR-1:2*ieR,2*ieL-1:2*ieL)=LHS3(2*ieR-1:2*ieR,2*ieL-1:2*ieL)-C'*[
37         abslambda/2,-nu/(2*deltax);-1/(2*Tr),abslambda/(2*deltax)];
38 end
```

```

34 LHS3(2*1-1:2*1,2*1-1:2*1)=LHS3(2*1-1:2*1,2*1-1:2*1)-C'*([abslambda/2,-nu
    /2;-1/(2*Tr),abslambda/2]*[0,0;0,1]+[-abslambda/2,-nu/2;-1/(2*Tr),-
    abslambda/2])*C;
35 LHS3(2*(numberx-1)-1:2*(numberx-1),2*(numberx-1)-1:2*(numberx-1))=LHS3
    (2*(numberx-1)-1:2*(numberx-1),2*(numberx-1)-1:2*(numberx-1))+C'*([
    abslambda/2,-nu/2;-1/(2*Tr),abslambda/2]+[-abslambda/2,-nu/2;-1/(2*Tr
    ),-abslambda/2]*[0,0;0,1])*C;
36
37 LHS=LHS1+LHS2+LHS3;
38
39 %取出我们所需要的 D
40 D=zeros(2*Unit,2*Unit);
41 for iface=2:numberx
42 ieL=iface-1;
43 D(2*ieL-1:2*ieL,2*ieL-1:2*ieL)=LHS(2*ieL-1:2*ieL,2*ieL-1:2*ieL);
44 end
45 %为循环所预设的一些量
46 Ucurrent=zeros(2,numberx-1);
47 Unext=zeros(2*Unit,1);
48 %initial condition set up
49 x=0;
50 for k=1:numberx-1
51 Ucurrent(1,k)=(x+deltax/2)^2-(x+deltax/2);
52 Ucurrent(2,k)=(2*(x+deltax/2)-1)*deltax;
53 x=x+deltax;
54 end
55
56 %Rdomain
57 x=0;
58 for k=1:numberx-1
59 Rd(1,k)=pi*(cos(pi*x)-cos(pi*(x+deltax)));
60 Rd(2,k)=-Ucurrent(2,k)/(Tr*deltax);
61 x=x+deltax;
62 end
63 %Rboundary
64 for iface=2:numberx-1
65 ieL=iface-1;
66 ieR=iface;
67 Fn(:,iface)=0.5*([-nu*Ucurrent(2,ieL)/deltax;-Ucurrent(1,ieL)/Tr]+[-nu*
    Ucurrent(2,ieR)/deltax;-Ucurrent(1,ieR)/Tr])-0.5*A*([Ucurrent(1,ieR);

```

```

    Ucurrent(2,ieR)/deltax]-[Ucurrent(1,ieL);Ucurrent(2,ieL)/deltax]);
68 Rb(:,ieL)=Rb(:,ieL)-C'*Fn(:,iface);
69 Rb(:,ieR)=Rb(:,ieR)+C'*Fn(:,iface);
70 end
71 Fn(:,1)=0.5*([-nu*Ucurrent(2,1)/deltax;0]+[-nu*Ucurrent(2,1)/deltax;-
    Ucurrent(1,1)/Tr])-0.5*A*([Ucurrent(1,1);Ucurrent(2,1)/deltax]-[0;
    Ucurrent(2,1)/deltax]);
72 Fn(:,numberx)=0.5*([-nu*Ucurrent(2,numberx-1)/deltax;-Ucurrent(1,numberx
    -1)/Tr]+[-nu*Ucurrent(2,numberx-1)/deltax;0])-0.5*A*([0;Ucurrent(2,
    numberx-1)/deltax]-[Ucurrent(1,numberx-1);Ucurrent(2,numberx-1)/
    deltax]);
73 Rb(:,1)=Rb(:,1)+C'*Fn(:,1);
74 Rb(:,numberx-1)=Rb(:,numberx-1)-C'*Fn(:,numberx);
75
76 %R 组装
77 for k=1:numberx-1
78 R(2*k-1:2*k,1)=Rd(:,k)+Rb(:,k);
79 end
80 %进行必要的向量等价转变
81 for k=1:numberx-1
82 Unext(2*k-1:2*k,1)=Ucurrent(:,k);
83 end
84 %循环迭代
85 for n=deltatau:deltatau:endtau
86 X=D\R;
87 if max(X)<tol
88 break
89 end
90 Unext=Unext+X;
91 Rd=zeros(2,numberx-1);
92 Rb=zeros(2,numberx-1);
93 for k=1:numberx-1
94 Ucurrent(:,k)=Unext(2*k-1:2*k,1);
95 end
96 %Rdomain
97 x=0;
98 for k=1:numberx-1
99 Rd(1,k)=pi*(cos(pi*x)-cos(pi*(x+deltax)));
100 Rd(2,k)=-Ucurrent(2,k)/(Tr*deltax);
101 x=x+deltax;

```

```

102 end
103 %Rboundary
104 for iface=2:numberx-1
105 ieL=iface-1;
106 ieR=iface;
107 Fn(:, iface)=0.5*([-nu*Ucurrent(2, ieL)/deltax;-Ucurrent(1, ieL)/Tr]+[-nu*
    Ucurrent(2, ieR)/deltax;-Ucurrent(1, ieR)/Tr])-0.5*A*([Ucurrent(1, ieR);
    Ucurrent(2, ieR)/deltax]-[Ucurrent(1, ieL); Ucurrent(2, ieL)/deltax]);
108 Rb(:, ieL)=Rb(:, ieL)-C'*Fn(:, iface);
109 Rb(:, ieR)=Rb(:, ieR)+C'*Fn(:, iface);
110 end
111 Fn(:, 1)=0.5*([-nu*Ucurrent(2, 1)/deltax;0]+[-nu*Ucurrent(2, 1)/deltax;-
    Ucurrent(1, 1)/Tr])-0.5*A*([Ucurrent(1, 1); Ucurrent(2, 1)/deltax]-[0;
    Ucurrent(2, 1)/deltax]);
112 Fn(:, numberx)=0.5*([-nu*Ucurrent(2, numberx-1)/deltax;-Ucurrent(1, numberx
    -1)/Tr]+[-nu*Ucurrent(2, numberx-1)/deltax;0])-0.5*A*([0; Ucurrent(2,
    numberx-1)/deltax]-[Ucurrent(1, numberx-1); Ucurrent(2, numberx-1)/
    deltax]);
113 Rb(:, 1)=Rb(:, 1)+C'*Fn(:, 1);
114 Rb(:, numberx-1)=Rb(:, numberx-1)-C'*Fn(:, numberx);
115
116 %R 组装
117 for k=1:numberx-1
118 R(2*k-1:2*k, 1)=Rd(:, k)+Rb(:, k);
119 end
120
121 for k=1:numberx-1
122 Unext(2*k-1:2*k, 1)=Ucurrent(:, k);
123 end
124 end
125 Unumsolution(1, :)=Ucurrent(1, :); Unumsolution(2, :)=Ucurrent(2, :)/deltax;
126 end

```

## $u$ 空间精度分析

```
1 function A=AccuracyU ( Unit , Unumsolution )
2 %deltx=1/Unit;endx=1;
3 numberx=endx/deltx+1;
4 %calculate the accuracy of space DGp0+DGP0
5 I1=0;t=[-1/sqrt(5) ,0 ,1/sqrt(5) ];W=[5/9 ,8/9 ,5/9];
6 k=1;%determine the correctness of the program
7 for x=0:deltx:endx-deltx
8     for i=1:3
9         xi=deltx/2*t(i)+0.5*(2*x+deltx);
10        for m=1:numberx-1
11            if xi>(m-1)*deltx&&xi<m*deltx
12                fi=(sin(pi*xi)-Unumsolution(1,m))^2;k=k+1;
13            end
14        end
15        I1=I1+W(i)*fi;
16    end
17 end
18 I1=I1*0.5*deltx;
19 A=sqrt(I1);
20 end
```

## $u_x$ 空间精度分析

```
1 function A1=AccuracyUx( Unit , Unumsolution)
2 %deltx=1/Unit;endx=1;
3 numberx=endx/deltx+1;
4 %calculate the accuracy of space
5 I2=0;t=[-1/sqrt(5) ,0 ,1/sqrt(5) ];W=[5/9 ,8/9 ,5/9];
6 k=1;%determine the correctness of the program
7 for x=0:deltx:endx-deltx
8     for i=1:3
9         xi=deltx/2*t(i)+0.5*(2*x+deltx);
10        for m=1:numberx-1
11            if xi>(m-1)*deltx&&xi<m*deltx
12                fi=(pi*cos(pi*xi)-Unumsolution(2,m))^2;k=k+1;
13            end
14        end
15        I2=I2+W(i)*fi;
16    end
17 end
18 I2=I2*0.5*deltx;
19 A1=sqrt(I2);
20 end
```