**偏微分方程现代数值解法  
数值实验报告 (第一次/HW7)**

匡亚明学院 211240021 田铭扬

**一 问题**

编程用线性有限元方法求解条带上的二维Poisson方程周期边值问题：  
并讨论有限元解误差的收敛阶。

**二 数学推导与程序设计**

考虑到问题的形式，假设原PDE的解有变量分离的形式，  
进而易求得其真解为。

关于程序设计，大部分与讲义中的**程序3.1**相同，唯一需额外处理的是周期边值条件的控制。笔者的方法是使用**poimesh**函数生成等腰直角三角形单元（保证区域的左右两端有对应的节点）；并在生成单元刚度矩阵前“手动”修改记录编号的**it1 it2 it3**数组，将所有位于右侧边界上的节点均改为左侧边界的对应节点；计算完毕后，再将左侧节点的结果“复制”到右侧。

这样做，相当于把计算区域“卷”成了一个环面，从而实现了周期边值条件。留意，在处理Dirichlet边界时，除了要消去上、下边界外，还要消去已经“不存在”的左侧边界（否则刚度矩阵中会有全是0的行列）。

最后，后验收敛阶的计算遵循以下公式：  
 .

**三 实验结果与分析**

实验结果如下表所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **单元边长 (直角边)** | **单元 数量** | **节点 数量** | **L∞ 误差** | | **L2 误差** | |
|  | **收敛阶** |  | **收敛阶** |
| 0.25 | 32 | 25 | 1.39e-17 |  | 1.55e-17 |  |
| 0.125 | 128 | 81 | 9.71e-17 |  | 1.37e-16 |  |
| 0.0625 | 512 | 289 | 2.78e-16 | X | 5.07e-16 | X |
| 0.03125 | 2048 | 1089 | 7.36e-16 | X | 2.32e-15 | X |
| 0.015625 | 8192 | 4225 | 2.37e-15 | X | 1.17e-14 | X |
| 0.0078125 | 16384 | 16641 | 7.91e-15 | X | 6.05e-14 | X |

表1 实验数据

从表格中可以看到，线性有限元方法对于此问题有极好的收敛性：若只考虑数值解与真解在节点处的误差，收敛阶甚至是无法计算的——从第一次网格加密开始，舍入误差就占据了主导地位，导致误差随着网格加密反而增大。

但是若是考虑到非格点处，情形就不太一样了。由于数值解采用的是分片线性函数，而真解是二次函数，因此在非格点处会产生O(a) 量级的局部离散误差（a是单元的直角边边长）。从这个意义上来讲，算法理论上是1阶收敛的。

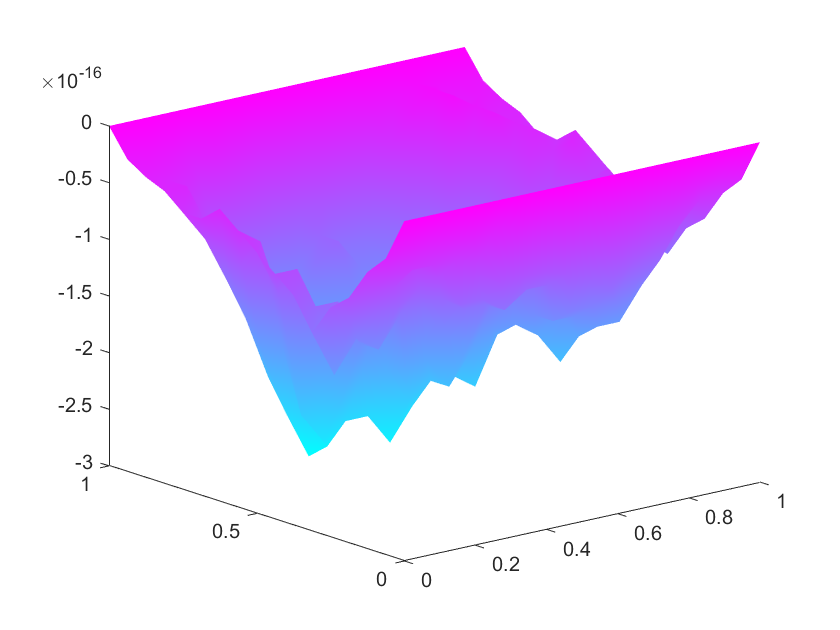
综合考虑，在实际应用时，应当根据场景需要的空间精度，在舍入误差的增加与离散误差的减小之间找到平衡点，从而达到最优的误差。

图1 误差图像（512单元/289节点）

**四 结论**

综上所述，线性有限元方法对此问题十分有效；但要根据实际场景的需要灵活地选择离散网格的精度，以在舍入误差与离散误差间达到平衡。

**附录：Matlab代码**

文件一：main.m

g=@square0101g;

[p,e,t]=poimesh(g);

[p,e,t]=refinemesh(g,p,e,t);

[p,e,t]=refinemesh(g,p,e,t); % start with a=0.25 (32 triangles)

% pdemesh(p,e,t);

uexact=(0.5\*p(2,:).\*(1-p(2,:)))';

u=pdeasmpoi\_period(p,e,t);

% pdesurf(p,t,u-uexact);

size(p(1,:))

max(abs(u-uexact))

max(0.25\*(u-uexact).^2)

for i=1:5

[p,e,t]=refinemesh(g,p,e,t);

uexact=(0.5\*p(2,:).\*(1-p(2,:)))';

u=pdeasmpoi\_period(p,e,t);

size(p(1,:))

max(abs(u-uexact))

max(2^(-i-2)\*(u-uexact).^2)

end

文件二：square0101g.m

function [x,y]=square0101g(bs,s) % Modified from "squareg.m"

nbs=4;

if nargin==0

x=nbs; % number of boundary segments

return

end

d=[

0 0 0 0 % start parameter value

1 1 1 1 % end parameter value

0 0 0 0 % left hand region

1 1 1 1 % right hand region

];

bs1=bs(:)';

if find(bs1<1 | bs1>nbs)

error(message('pde:squareg:InvalidBs'))

end

if nargin==1

x=d(:,bs1);

return

end

x=zeros(size(s));

y=zeros(size(s));

[m,n]=size(bs);

if m==1 && n==1

bs=bs\*ones(size(s)); % expand bs

elseif m~=size(s,1) || n~=size(s,2)

error(message('pde:squareg:SizeBs'));

end

if ~isempty(s)

% boundary segment 1

ii=find(bs==1);

if length(ii)

x(ii)=interp1([d(1,1),d(2,1)],[0 1],s(ii));

y(ii)=interp1([d(1,1),d(2,1)],[1 1],s(ii));

end

% boundary segment 2

ii=find(bs==2);

if length(ii)

x(ii)=interp1([d(1,2),d(2,2)],[1 1],s(ii));

y(ii)=interp1([d(1,2),d(2,2)],[1 0],s(ii));

end

% boundary segment 3

ii=find(bs==3);

if length(ii)

x(ii)=interp1([d(1,3),d(2,3)],[1 0],s(ii));

y(ii)=interp1([d(1,3),d(2,3)],[0 0],s(ii));

end

% boundary segment 4

ii=find(bs==4);

if length(ii)

x(ii)=interp1([d(1,4),d(2,4)],[0 0],s(ii));

y(ii)=interp1([d(1,4),d(2,4)],[0 1],s(ii));

end

end

文件三：pdeasmpoi\_period.m

function u=pdeasmpoi\_period(p,e,t)

% Slightlt modified from "pdeasmpoi.m" from the handouts

it1=t(1,:); it2=t(2,:); it3=t(3,:);

np=size(p,2);

[ar,g1x,g1y,g2x,g2y,g3x,g3y]=pdetrg(p,t);

% Period-Boudary on Segment 2 & 4

p2=[[];[]]; p4=[[];[]];

for j=1:np

%if(abs(p(2,j))<1e-6 || abs(p(2,j)-1)<1e-6)

% continue

%end

if(abs(p(1,j)-1)<1e-6)

p2=[p2 [j;p(2,j)]];

end

if(abs(p(1,j))<1e-6)

p4=[p4 [j;p(2,j)]];

end

end

p2=(sortrows(p2',2))';

p4=(sortrows(p4',2))';

for j=1:size(p2,2)

it1(it1==p2(1,j))=p4(1,j);

it2(it2==p2(1,j))=p4(1,j);

it3(it3==p2(1,j))=p4(1,j);

end

c3=(g1x.\*g2x+g1y.\*g2y).\*ar;

c1=(g2x.\*g3x+g2y.\*g3y).\*ar;

c2=(g3x.\*g1x+g3y.\*g1y).\*ar;

A=sparse(it1,it2,c3,np,np);

A=A+sparse(it2,it3,c1,np,np);

A=A+sparse(it3,it1,c2,np,np);

A=A+A.';

A=A+sparse(it1,it1,-c2-c3,np,np);

A=A+sparse(it2,it2,-c3-c1,np,np);

A=A+sparse(it3,it3,-c1-c2,np,np);

f=ar/3;

F=sparse(it1,1,f,np,1);

F=F+sparse(it2,1,f,np,1);

F=F+sparse(it3,1,f,np,1);

% Dirichlet-0 on Segment 1 & 3

ie=zeros(1,np);

for j=1:size(e,2)

if(e(5,j)==1 || e(5,j)==3)

ie(e(1,j))=1;

ie(e(2,j))=1;

end

end

ie(p2(1,:))=1; % Important!!! Otherwise A will be singular.

ie=find(ie);

B=speye(np);

B(:,ie)=[];

A=B'\*A\*B;

F=B'\*F;

un=A\F;

u=B\*un; % Redo the Dirichlet-Boundary

u(p2(1,:))=u(p4(1,:)); % Redo the Period-Boudary modification\

u=full(u);