偏微分方程数值解 上机作业 1

信计 81 韩恒锐 2160506010

一、问题分析

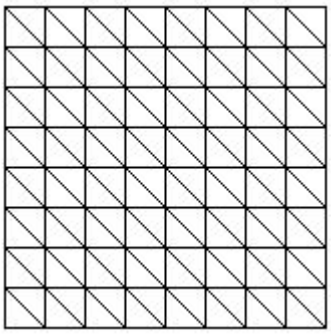
考虑矩形区域 $G \in (0,1)^2$ 上如下 Poisson 方程第一边值问题：

$$-\Delta u = f = (2+\pi^2 x(1-x))\sin \pi y, (x,y) \in G$$
$$u = \alpha = 0, (x,y) \in \Gamma = \partial G$$

该方程的解析解为

$$u = x(1-x)\sin \pi y$$

我们使用如图所示的三角形剖分：



间隔为 $h = 1/n$ 的网格将区域划分成 n^2 个小正方形，每个正方形又划分为左下、右上两个三角元。每个三角元的面积为 $S = \frac{h^2}{2}$。

在一次元中，网格中共有 $(n+1)^2$ 个节点，包括 $(n-1)^2$ 个内点和 $2n+1$ 个边界点。

在二次元中，网格中共有 $(2n+1)^2$ 个节点，包括 $(2n-1)^2$ 个内点和 $4n+1$ 个边界点。在内点中，又包括 $(n-1)^2$ 个顶点和 $n(3n-2)$ 个中点。

有限元方程的形式为：

$$\sum_{i=1}^{n_1} a(\varphi_i, \varphi_j) u_i = (f, \varphi_j) - \sum_{i=n_1+1}^{n_2} a(\varphi_i, \varphi_j)\alpha_i,$$
$$j = 1,2,\cdots,n_1$$

由边界条件有 $\alpha_i = 0$，上式化为

$$\sum_{i=1}^{n_1} a(\varphi_i,\varphi_j)u_i = (f,\varphi_j),$$
$$j = 1,2,\cdots,n_1$$

以下记

$$a(\varphi_i,\varphi_j) = a_{ij}, (f,\varphi_j) = b_j$$

经理论分析可得，在一次元中，若 i、j 为相同节点，$a_{ij} = 4$；若 i、j 为相邻节点，$a_{ij} = -1$；否则 $a_{ij} = 0$。在二次元中，若 i 为顶点：若 i、j 为相同节点，$a_{ij} = 4$；若 i、j 为相邻节点，$a_{ij} = -\frac{4}{3}$；若 i、j 相隔一个节点，$a_{ij} = \frac{1}{3}$；否则 $a_{ij} = 0$。若 i 为中点：若 i、j 为相同节点，$a_{ij} = \frac{16}{3}$；若 i、j 为相邻节点，$a_{ij} = -\frac{4}{3}$；否则 $a_{ij} = 0$。

利用三角形上的 Gauss 求积公式[1]可得：在一次元中，$b_j = 2S\overline{f_j}$，其中 $\overline{f_j}$ 是 f 在点 j 连接的 6 条网格线的中点处的平均值。在二次元中，若 j 为顶点，$b_j = 0$；若 j 为中点，$b_j = \frac{2}{3}Sf_j$。

利用以上结论即可通过解线性方程组得到原方程的数值解

二、计算结果

利用 Matlab 程序得到的计算结果如下：

	网格尺寸 h	$L^2(H^0)$ 误差	H^1 误差	程序运行时间[2]
一次元	0.1	0.0037636160	0.0916866634	0.008s
	0.05	0.0009484191	0.0460392160	0.016s
	0.01	0.0000380329	0.0092204729	3.602s
二次元	0.1	0.0000008952	0.0065557114	0.040s
	0.05	0.0000000556	0.0016397759	0.306s
	0.01	8.8671*10^-11	0.0000065602	213.170s

三、结果分析

从以上的结果中，我们可以计算误差与网格尺寸 h 关系的函数拟合：

一次元：
$$\log(\|u-u_h\|_0) = 1.9960\log h - 0.9844$$
$$\log(\|\nabla(u-u_h)\|_0) = 0.9978\log h - 0.0906$$

---

[1] 计算中所有三角元上的积分均采用数值积分近似，计算公式为：$\int f dxdy = S\overline{f}$，其中 S 是三角元的面积，$\overline{f}$ 是 f 在三条边中点上的平均值。

[2] 程序运行时间到将数值解计算出为止，后续的误差分析所占用时间不计算在内。

二次元：$\log(\|u-u_h\|_0)=4.0038\log h-4.7087$

$\log(\|\nabla(u-u_h)\|_0)=3.0756\log h-2.3613$

由此可得误差阶的估计：

一次元：$\|u-u_h\|_0=O(h^2),\|\nabla(u-u_h)\|_0=O(h)$

二次元：$\|u-u_h\|_0=O(h^4),\|\nabla(u-u_h)\|_0=O(h^3)$

这与教材中的结论：$\|u-u_h\|_1\le Ch\|u''\|,\|u-u_h\|\le c\alpha h^2\|u''\|$ 相符合。

四、附录：matlab 代码

PDE1_1.m

```
%一次 Lagrange 元求解偏微分方程
n=floor(1/input('请输入网格尺寸'));
h=1/n;%网格尺寸
g=h/2;
n=n-1;
A=zeros(n^2,n^2);
B=zeros(n^2,1);
s_mu=h^2/2;%三角形元的面积
for i=1:n^2
    for j=[i,i+1,i+n]
        if(j<=n^2)
            A(i,j)=a1(i,j,n);%计算方程系数
        end
    end
end
for i=2:n^2
    for j=[i-n,i-1]
        if(j>=1)
            A(i,j)=A(j,i);
        end
    end
end
for j=1:n^2
    B(j,1)=b1(j,h,n);%计算右端项系数
end
u=A\B;
uh=zeros(n+2,n+2);%将列向量 u 重新排布为插值近似 uh
for i=1:n+2
    for j=1:n+2
        if(i==1||i==n+2||j==1||j==n+2)
            uh(i,j)=0;
```

```
            else
                uh(i,j)=u((j-2)*n+i-1);
            end
        end
end
err0=0;%计算 L2(H0)度量下的误差
for i=1:n+1
        for j=1:n+1
            x=i*h;
            y=j*h;

err0=err0+(u0(x-g,y)-(uh(i+1,j+1)+uh(i,j+1))/2)^2+(u0(x,y-g)-(uh(i+1,j+1)+uh(i+1,j))/2)^2+2*(u0(x-
g,y-g)-(uh(i+1,j+1)+uh(i,j))/2)^2+(u0(x-g,y-h)-(uh(i+1,j)+uh(i,j))/2)^2+(u0(x-h,y-g)-(uh(i,j+1)+uh(i,j))
/2)^2;
        end
end
err0=sqrt(err0*s_mu/3);
err1=0;%计算 H1 度量下的误差
for i=1:n+1
        for j=1:n+1
            x=i*h;
            y=j*h;
            uh_grad_left=[uh(i+1,j)-uh(i,j),uh(i,j+1)-uh(i,j)]/h;
            uh_grad_right=[uh(i+1,j+1)-uh(i,j+1),uh(i+1,j+1)-uh(i+1,j)]/h;

err1=err1+sqnorm(u0_grad(x-h,y-g)-uh_grad_left)+sqnorm(u0_grad(x-g,y-h)-uh_grad_left)+sqno
rm(u0_grad(x-g,y-g)-uh_grad_left)+sqnorm(u0_grad(x,y-g)-uh_grad_right)+sqnorm(u0_grad(x-g,y)
-uh_grad_right)+sqnorm(u0_grad(x-g,y-g)-uh_grad_right);
        end
end
err1=sqrt(err1*s_mu/3);
disp(err0);
disp(err1);
```

PDE1_2.m

```
%二次 Lagrange 元求解偏微分方程
n=floor(1/input('请输入网格尺寸'));
h=1/n;%网格尺寸
g=h/2;
n=2*n-1;
A=zeros(n^2,n^2);
B=zeros(n^2,1);
s_mu=h^2/2;%三角形元的面积
for i=1:n^2
```

```
    for j=[i,i+1,i+2,i+n,i+2*n]
        if(j<=n^2)
            A(i,j)=a2(i,j,n);%计算方程系数
        end
    end
end
for i=2:n^2
    for j=[i-2*n,i-n,i-2,i-1]
        if(j>=1)
            A(i,j)=A(j,i);
        end
    end
end
for j=1:n^2
    B(j,1)=b2(j,h,n);%计算右端项系数
end
u=A\B;
uh=zeros(n+2,n+2);%将列向量 u 重新排布为插值近似 uh
for i=1:n+2
    for j=1:n+2
        if(i==1||i==n+2||j==1||j==n+2)
            uh(i,j)=0;
        else
            uh(i,j)=u((j-2)*n+i-1);
        end
    end
end
err0=0;%计算 L2(H0)度量下的误差
for i=2:2:n+1
    for j=2:2:n+1
        x=i*g;
        y=j*g;
        err0=err0+(u0(x-g,y)-uh(i,j+1))^2+(u0(x,y-g)-uh(i+1,j))^2+2*(u0(x-g,y-g)-uh(i,j))^2+(u0(x
    -g,y-h)-uh(i,j-1))^2+(u0(x-h,y-g)-uh(i-1,j))^2;
    end
end
err0=sqrt(err0*s_mu/3);
err1=0;%计算 H1 度量下的误差
for i=2:2:n+1
    for j=2:2:n+1
        x=i*g;
        y=j*g;

    uh_grad_1=[-uh(i-1,j-1)-uh(i+1,j-1)+2*(uh(i,j)-uh(i-1,j)+uh(i,j-1)),-uh(i-1,j-1)+uh(i-1,j+1)]/h;
```

```
    uh_grad_2=[-uh(i-1,j-1)+uh(i+1,j-1),-uh(i-1,j-1)-uh(i-1,j+1)+2*(uh(i,j)-uh(i,j-1)+uh(i-1,j))]/h;
    uh_grad_3=[uh(i-1,j-1)+uh(i+1,j-1)+2*(uh(i,j)-uh(i-1,j)-uh(i,j-1)),uh(i-1,j-1)+uh(i-1,j+1)+2*(uh(
i,j)-uh(i-1,j)-uh(i,j-1))]/h;
    uh_grad_4=-[-uh(i+1,j+1)-uh(i-1,j+1)+2*(uh(i,j)-uh(i+1,j)+uh(i,j+1)),-uh(i+1,j+1)+uh(i+1,j-1)]/
h;
    uh_grad_5=-[-uh(i+1,j+1)+uh(i-1,j+1),-uh(i+1,j+1)-uh(i+1,j-1)+2*(uh(i,j)-uh(i,j+1)+uh(i+1,j))]/
h;
    uh_grad_6=-[uh(i+1,j+1)+uh(i-1,j+1)+2*(uh(i,j)-uh(i+1,j)-uh(i,j+1)),uh(i+1,j+1)+uh(i+1,j-1)+2*(
uh(i,j)-uh(i+1,j)-uh(i,j+1))]/h;
err1=err1+sqnorm(u0_grad(x-h,y-g)-uh_grad_1)+sqnorm(u0_grad(x-g,y-h)-uh_grad_2)+sqnorm(u
0_grad(x-g,y-g)-uh_grad_3)+sqnorm(u0_grad(x,y-g)-uh_grad_4)+sqnorm(u0_grad(x-g,y)-uh_grad
_5)+sqnorm(u0_grad(x-g,y-g)-uh_grad_6);
    end
end
err1=sqrt(err1*s_mu/3);
disp(err0);
disp(err1);
```

a1.m

```
function a=a1(i,j,n)%一次有限元方程系数计算
a=0;
ix=mod(i-1,n)+1;
iy=ceil(i/n);
jx=mod(j-1,n)+1;
jy=ceil(j/n);
if(ix==jx && iy==jy)
    a=4;
elseif(abs(jx-ix)==1 && iy==jy)||(abs(jy-iy)==1 && ix==jx)
    a=-1;
end
end
```

a2.m

```
function a=a2(i,j,n)%二次有限元方程系数计算
a=0;
ix=mod(i-1,n)+1;
iy=ceil(i/n);
jx=mod(j-1,n)+1;
jy=ceil(j/n);
if(mod(ix,2)==0 && mod(iy,2)==0)%点 i 是顶点
    if (ix==jx && iy==jy)%i 与 j 是同一点
        a=4;
    elseif((ix==jx && abs(iy-jy)==1)||(iy==jy && abs(ix-jx)==1))%i 与 j 是相邻点
        a=-4/3;
```

```
    elseif((ix==jx && abs(iy-jy)==2)||(iy==jy && abs(ix-jx)==2))%i 与 j 相隔一点
        a=1/3;
    end
else%点 i 是中点
    if (ix==jx && iy==jy)%i 与 j 是同一点
        a=16/3;
    elseif(ix==jx && abs(iy-jy)==1)||(iy==jy && abs(ix-jx)==1)%i 与 j 是相邻点
        a=-4/3;
    end
end
```

b1.m

```
function b=b1(j,h,n)
jx=mod(j-1,n)+1;
jy=ceil(j/n);
x=jx*h;
y=jy*h;
g=h/2;
s_mu=h^2/2;
b=s_mu*(f(x+g,y)+f(x,y+g)+f(x-g,y+g)+f(x-g,y)+f(x,y-g)+f(x+g,y-g))/3;%积分的数值计算
```

b2.m

```
function b=b2(j,h,n)
jx=mod(j-1,n)+1;
jy=ceil(j/n);
x=jx*h/2;
y=jy*h/2;
s_mu=h^2/2;
%积分的数值计算
if ~(mod(jx,2)==0 && mod(jy,2)==0)%点 i 是中点
    b=s_mu*f(x,y)*2/3;
else
    b=0;
end
```

f.m

```
function f=f(x0,y0)
f=(2+pi^2*x0*(1-x0))*sin(pi*y0);
end
```

u0.m

```
function u0=u0(x0,y0)
u0=x0*(1-x0)*sin(pi*y0);
end
```

u0_grad.m

```
function du=u0_grad(x0,y0)
    du=[(1-2*x0)*sin(pi*y0),x0*(1-x0)*pi*cos(pi*y0)];
end
```

sqnorm.m

```
function s=sqnorm(a)
    s=sum(a.^2);
end
```