偏微分方程数值解上机作业 4

段俊明*

2015 年 12 月 17 日

1 问题重述

求解正方形区域 $(0,1)\times(0,1)$ 上的 Poisson 方程混合边值问题

$$\begin{cases} -\Delta u = f, \\ u = u_0, \quad \boldsymbol{x} \in \partial\Omega_0. \\ \dfrac{\partial u}{\partial \nu} + \beta u = g, \quad \boldsymbol{x} \in \partial\Omega_1. \end{cases} \tag{1}$$

其中，$\partial\Omega_0 = \{0\}\times[0,1], \beta > 0$.

用 C^0 类型 (1) 矩形和型 (1) 三角形有限元构造的有限元函数空间上数值求解该问题.

2 理论分析

将原问题化为变分形式:

$$\begin{cases} a(u,v) = f(v), \\ a(u,v) = \int_\Omega \nabla u \cdot \nabla v \mathrm{d}\boldsymbol{x} + \beta \int_{\Omega_1} uv\mathrm{d}s, \\ f(v) = \int_\Omega fv\mathrm{d}\boldsymbol{x} + \int_{\Omega_1} gv\mathrm{d}s, \\ u \in \mathbb{V}_1 = \{\mathbb{H}^1(\Omega), u|_{\partial\Omega_0 = u_0}\}, \\ v \in \mathbb{V}_2 = \{\mathbb{H}^1(\Omega), v|_{\partial\Omega_0 = 0}\}. \end{cases} \tag{2}$$

取 xy 方向同样为 N 的网格数，将以上变分问题离散化, 等价与求解线性方程组:

$$\sum_{i=1}^{N_h} a(\phi_j,\phi_i)u_j = (f,\phi_i), \quad i = 1,2,\ldots,N_h. \tag{3}$$

其中 N_h 为总节点数，ϕ_i 为一组基函数, 在第 i 个节点上为 1, 其它节点处为 0. u_i 为第 i 个节点上的数值解.

*北京大学数学科学学院，科学与工程计算系，邮箱：duanjm@pku.edu.cn

将上述方程组写作:

$$\boldsymbol{K}\boldsymbol{u}_h = \boldsymbol{f},$$

并称 $\boldsymbol{K}$ 为刚度矩阵,$\boldsymbol{u}_h$ 为位移向量,$\boldsymbol{f}$ 为载荷向量.

下面需要确定的就是 $\boldsymbol{K}$ 和 $\boldsymbol{f}$ 的元素.

用 e 来表示单元序号，T_e 表示单元，$a^e(u,v) = \int_{T_e} \nabla u \cdot \nabla v \mathrm{d}\boldsymbol{x}$, 定义单元刚度矩阵 $\boldsymbol{K}^e$ 和单元载荷向量 $\boldsymbol{f}^e$, 则

$$k_{ij} = a(\phi_j,\phi_i) = \sum_{e=1}^{M} a^e(\phi_j,\phi_i) = \sum_{e=1}^{M} k_{ij}^e. \tag{4}$$

其中 k_{ij}^e 表示第 e 个单元的单元刚度矩阵的 (i,j) 元素. 对于型 (1) 三角形单元，书上 P_{213} 已经给出单元刚度矩阵 $\boldsymbol{K}^e$ 的结果; 对于型 (1) 矩形单元, 下面给出单元刚度矩阵:

取边长为 1 的正方形, 有四个节点 $(0,0),(1,0),(1,1),(0,1)$, 给出它们的一组基:

$$(1-x)(1-y), x(1-y), xy, (1-x)y.$$

利用与书上类似的仿射变换，将任意正方形上的单元刚度矩阵统一到标准正方形上来处理, 最终得到:

$$\boldsymbol{K}^e = \frac{h^2}{\det A_e}\begin{bmatrix} \frac{2}{3} & -\frac{1}{6} & -\frac{1}{3} & -\frac{1}{6} \\ -\frac{1}{6} & \frac{2}{3} & -\frac{1}{6} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{6} & \frac{2}{3} & -\frac{1}{6} \\ -\frac{1}{6} & -\frac{1}{3} & -\frac{1}{6} & \frac{2}{3} \end{bmatrix}$$

当 f 为常数时，与书上类似得到:

$$\boldsymbol{f}^e = \frac{1}{4} f(T_e)|T_e|(1,1,1)^{\mathrm{T}}.$$

对于边界项积分，采用数值积分公式, 并将其加入到刚度矩阵和载荷向量中; 对于 Diriclet 边界上的节点, 将其从生成的线性方程组中去除.

3 算法

基于以上分析，有如下算法:

Algorithm 1: 生成刚度矩阵和载荷向量

$\boldsymbol{K} = (k(i,j)) = 0; \boldsymbol{f} = (f(i)) = 0;$

for $e = 1 : M$ **do**

  计算单元刚度矩阵$\boldsymbol{K}^e = (k^e(\alpha,\beta))$;

  计算单元载荷向量$\boldsymbol{f}^e = (f^e(\alpha))$;

  $k^e(en(\alpha,e), en(\beta,e)) += k^e(\alpha,\beta)$;

  $f^e(en(\alpha,e)) += f^e(\alpha)$;

end

其中 $en(\alpha,e)$ 是取值为整体节点序数的数组，α 为局部节点序数. 还需要保存单元坐标，$cd(i,nd)$ 是取值为空间坐标的数组，表示第 nd 个整体节点的空间坐标的第 i 个分量.

4 程序说明

main.cpp 是调用 fem.h 的主函数，space.h 是计算 cd,en 两个数组的头文件, fem.cpp 实现了该算例生成线性方程组并求解的过程. 线性方程组部分调用了 Eigen 的稀疏矩阵 LU 求解器.

5 数值结果

5.1 算例 1

取 $\beta = 1$, 精确解为 $x^3 + y^3$, 得到相应的初边值.

N	err($\mathbb{L}^1$)	err($\mathbb{L}^2$)	err($\mathbb{L}^\infty$)
10	1.416766e-03	2.031798e-03	1.116735e-02
20	3.591712e-04	4.899270e-04	3.612197e-03
40	9.024779e-05	1.193004e-04	1.109726e-03
80	2.262573e-05	2.938159e-05	3.292769e-04
160	5.664737e-06	7.289209e-06	9.528994e-05
10	1.455257e-03	1.972807e-03	7.418216e-03
20	3.717288e-04	4.885795e-04	2.193739e-03
40	9.408361e-05	1.215063e-04	6.277910e-04
80	2.367781e-05	3.030359e-05	1.758027e-04
160	5.939958e-06	7.568123e-06	4.850220e-05

表 1: 算例 1 各范数误差

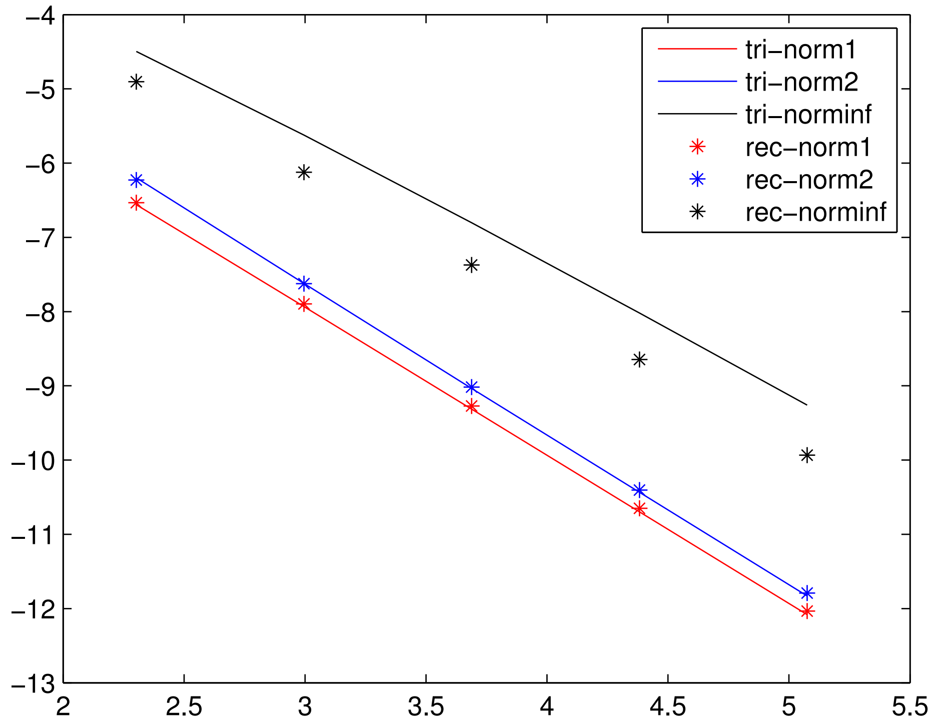

图 1: 算例 1 两种单元三种范数下的误差 -N 双对数坐标图

5.2 算例 2

取 $\beta = 1$, 精确解为 $e^x \sin(2\pi y)$, 得到相应的初边值.

N	$\mathrm{err}(\mathbb{L}^1)$	$\mathrm{err}(\mathbb{L}^2)$	$\mathrm{err}(\mathbb{L}^\infty)$
10	.604314e-02	2.684293e-02	1.329546e-01
20	3.606006e-03	6.008061e-03	3.564902e-02
40	8.486610e-04	1.400736e-03	9.365661e-03
80	2.052805e-04	3.368726e-04	2.444344e-03
160	5.045758e-05	8.252397e-05	6.360895e-04
10	1.659168e-02	2.622751e-02	7.467132e-02
20	3.744808e-03	5.911147e-03	1.850623e-02
40	8.878184e-04	1.393688e-03	4.617107e-03
80	2.151909e-04	3.375518e-04	1.154275e-03
160	5.291719e-05	8.300403e-05	2.886370e-04

表 2: 算例 2 各范数误差

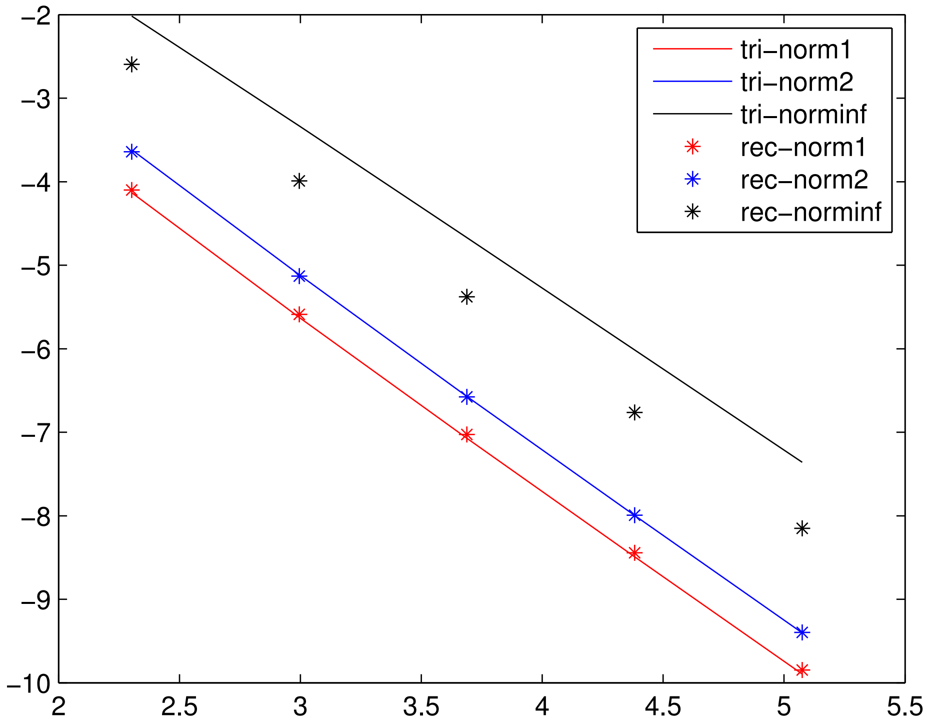

图 2: 算例 2 两种单元三种范数下的误差 -N 双对数坐标图

5.3 结果分析

可以看到，两个算例均收敛到了精确解. 在三角形单元和矩形单元的两种情况下它们的收敛阶是相同的, 在三种范数下的收敛阶也都相同.

算例 1 的数值解误差收敛阶约为 2, 算例 2 的数值解误差收敛阶略大于 2,

虽然在两种计算单元下误差收敛阶相同，但在 $\mathbb{L}^\infty$ 范数下, 矩形单元下的误差明显小于三角形单元. 这是因为矩形单元有 4 个自由度，使得计算结果优于三角形.

6 总结

有限元方法求得的数值解的误差在不同范数意义下基本相同, 数值结果显示出矩形单元的误差在 $\mathbb{L}^\infty$ 范数下要优于三角形单元.