

偏微分方程数值解 + 第二次上机作业

2100012131 蒋鹏

2025 年 11 月 30 日

目录

1 问题描述	1
2 算法设计	1
2.1 网格离散	1
2.2 算子离散及边界条件处理	3
2.3 数值算法	4
3 数值结果	4
3.1 数值算例	4
3.2 数值实验及结果	4
3.3 简要分析	4
4 后续改进	5

1 问题描述

在指定区域上 Ω 求解 Robin 边值问题的 Laplace 方程

$$\begin{cases} -\Delta u = f, & x \in \Omega \\ \alpha u + \beta \frac{\partial u}{\partial n} = g, & x \in \partial\Omega \end{cases} \quad (1)$$

其中区域 Ω 如图1所示。当 $\alpha = 0$ 得到 Neumann 边值问题，当 $\beta = 0$ 得到 Dirichlet 边值问题，所以我们设计该方程的对应有限差分格式，同时满足三类问题的求解。

2 算法设计

2.1 网格离散

在 x 和 y 方向以相同尺度 h 进行均匀剖分。令 $n = 1/h$, 自下而上，从左至右，对 Domain 内的点及边界点进行编号， $0 \leq i \leq 4n, 0 \leq j \leq j_num[i]$ 。所有点可分为三类：内点，边界点及临边界点，角点。

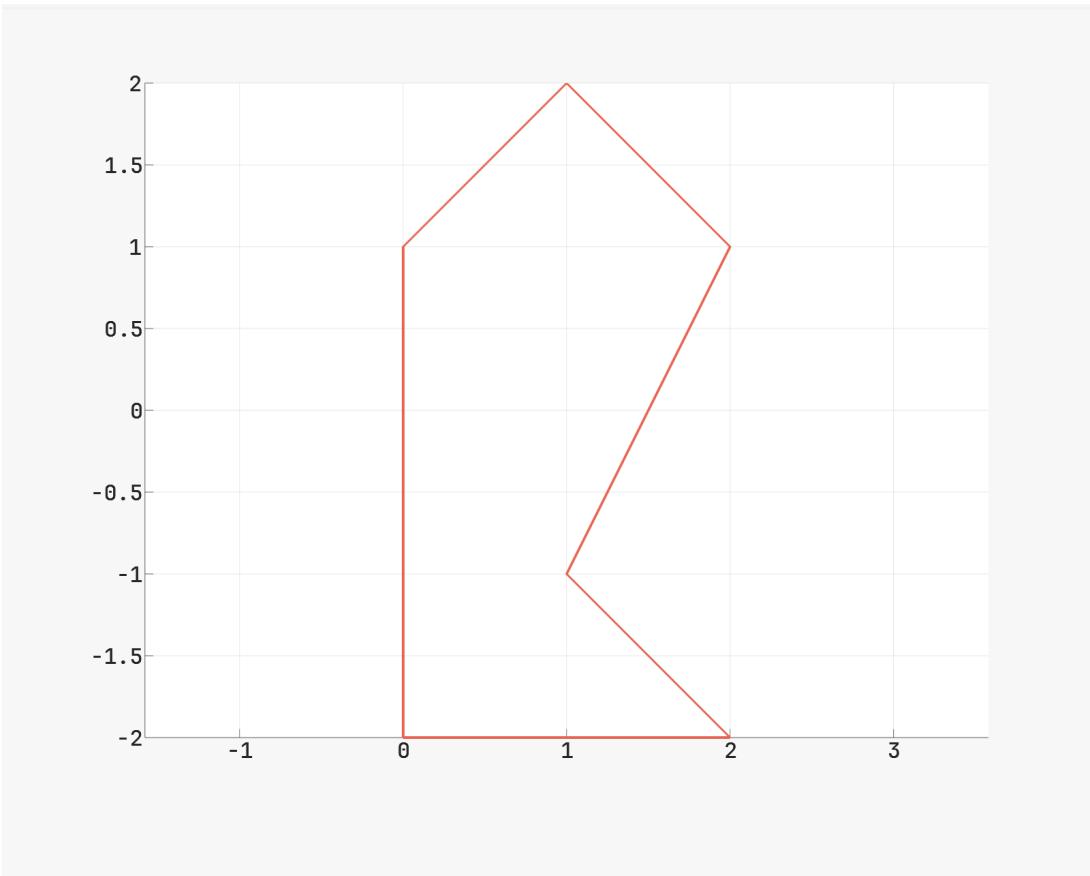


图 1: Domain

2.2 算子离散及边界条件处理

对于不同点进行不同格式离散：

对于内点，采取五点差分格式离散 Laplace 算子。即

$$AU_{i,j} = \frac{-U_{i+1,j} - U_{i-1,j} - U_{i,j+1} - U_{i,j-1} + 4U_{i,j}}{h^2} = b_{i,j} = f(x_i, y_j) \quad (2)$$

对于边界点，采取前向或者后向差商格式离散梯度，进而离散方向导数。具体而言，就是把边界上的点也纳入数值解，边界条件也视为线性算子在数值解向量上的作用。以下边界上的点为例，通过数值离散，得到

$$g(x_i, y_j) = \alpha u(x_i, y_j) + \beta \frac{\partial u}{\partial \vec{n}}(x_i, y_j) \quad (3)$$

$$\approx \alpha U_{i,j} + \beta \frac{U_{i,j} - U_{i+1,j}}{h} \quad (4)$$

这就是此边界点所对应的线性方程，可以认为是矩阵-向量乘法的形式，用于 GMRES 的矩阵-向量乘法更新：

$$AU_{i,j} = \alpha U_{i,j} + \beta \frac{U_{i,j} - U_{i+1,j}}{h} = b_{i,j} = g(x_i, y_j) \quad (5)$$

对于临边界点，即该点 (x, y) 非内点且不位于边界，则在边界上最近点为 (x_i^*, y_j^*) ，以点 (x, y) 的差商近似 (x_i^*, y_j^*) 的梯度，进而离散方向导数。具体和边界点类似，以右边界 $y = 2x - 1 (1 \leq x \leq 2)$ 附近的临边界点为例， $(x_i^*, y_j^*) = (x + 0.4h, y - 0.2h)$ ，通过数值离散和近似，得到

$$g(x_i^*, y_j^*) = \alpha u(x^*, y^*) + \beta \frac{\partial u}{\partial \vec{n}}(x^*, y^*) \quad (6)$$

$$\approx \alpha u(x_i, y_j) + \beta \frac{\partial u}{\partial \vec{n}}(x_i, y_j) \quad (7)$$

$$\approx \alpha U_{i,j} + \beta \left(\frac{U_{i,j} - U_{i-1,j}}{h}, \frac{U_{i+1,j} - U_{i,j}}{h} \right) \cdot (2/\sqrt{5}, -1/\sqrt{5}) \quad (8)$$

这就是此边界点所对应的线性方程，可以认为是矩阵-向量乘法的形式，用于 GMRES 的矩阵-向量乘法更新：

$$AU_{i,j} = \alpha U_{i,j} + \beta \left(\frac{U_{i,j} - U_{i-1,j}}{h}, \frac{U_{i+1,j} - U_{i,j}}{h} \right) \cdot (2/\sqrt{5}, -1/\sqrt{5}) = b_{i,j} = g(x_i^*, y_j^*) \quad (9)$$

对于角点，赋予 Dirichlet 边界条件，即规定角点处取值为真实解。

$$AU_{i,j} = b_{i,j} = u(x_i, y_j) \quad (10)$$

这样做的原因是为了保持 Neumann 边界条件解的唯一性，从求解线性方程组的角度上讲，是为了保持矩阵可逆。同时由于在角点，并没有良定义的法向导数，所以无法使用 Neumann 或 Robin 边界条件设置离散的线性方程。如果在角点并未给出所谓真实解，也可以直接选择其他方式，如固定某内点的值为确定值，或者取在区域上积分为 0 为补充的线性方程，我们这种方式只是一种示例。

这样，我们在离散后的网格上的每个点都通过数值离散得到了一个线性方程，组合在一起的整体就是一个关于矩阵向量的线性方程组。注意到此方程组的构造形式基本适用于所有边界条件，所以我们可以进行统一求解，只需要根据问题设定 α, β 即可。

边界条件	参数取值	h	GMRES 迭代次数	$\ e_h\ _\infty$
Dirichlet	alpha=1,beta=0	0.125	16	3.1e-3
		0.0625	48	1.6e-3
		0.03125	172	8.0e-4
		0.015625	619	4.0e-4
		0.0078125	2367	2.0e-4
Neumann	alpha=0,beta=1	0.125	101	3.7e-2
		0.0625	502	1.9e-2
		0.03125	1169	9.7e-3
		0.015625	5983	5.0e-3
		0.0078125	31606	2.6e-3
Robin	alpha=1,beta=1	0.125	42	1.7e-2
		0.0625	104	8.1e-3
		0.03125	321	4.0e-3
		0.015625	1831	2.0e-3
		0.0078125	8553	1.0e-3

表 1: 数值结果展示

2.3 数值算法

此线性方程组较为复杂，我们采取 GMRES(m) 迭代法进行求解。为了保证收敛性，先进行 Gauss-Seidel 迭代约 100 步，找到较为理想的初值，随后进行 GMRES 求解。

3 数值结果

3.1 数值算例

采取给定的数值算例，真实解 $u(x, y) = \frac{\sin(\pi x) \cos(2\pi y)}{5\pi^2}$ ，进而源项 $f(x, y) = \sin(\pi x) \cos(2\pi y)$ 。

3.2 数值实验及结果

对 Dirichlet、Neumann、Robin 三类边界条件均进行数值实验，所得结果位于文档 `ouput.txt`，列表如1所示。

3.3 简要分析

- 求解耗时、迭代次数: Dirichlet < Robin < Neumann. 原因是不同边值条件导致离散矩阵的条件数差异较大。
- 从表中可以清晰看出误差的下降趋势，且该数值格式具有一阶精度，即

$$\|e_h\|_\infty = \mathcal{O}(h) \quad (11)$$

- 修改单次迭代上限 m , 可以减少 GMRES 迭代次数, 但单次迭代耗时增加, 总 CPU 时间应当基本不变。

4 后续改进

为了加速求解速率, 改进版将增加不完全 LU 分解 ILU 做预条件子, 实现 ILU-GMRES(m) 求解器。