摘要

本次编程作业实现了二维 Poission 方程的求解，报告将对结果进行展示并进行误差分析。

1 程序设计思路及实现原理

- **规则区域、Dirichlet 边界条件：**对于规则区域 Ω，由于 Dirichlet 边界条件下，边界上的函数值已知，我们直接采用讲义里给出的五点差分法离散拉普拉斯算子 Δ。
- **规则区域、Neumann 边界条件：**此时边界上的函数值未知，假设格点大小 $h=\frac{1}{N}$，对于直线 $x=h$, $x=1-h$, $y=h$, $y=1-h$ 上的点，此时无法直接用五点差分离散拉普拉斯算子 Δ，这里没有采用讲义上的 ghost cell 方法，而是直接用点 U_A, U_B, U_C 进行插值得到插值函数 p，用 p 函数的一阶导数近似边界上原函数 f 的一阶导数。具体见图片 1

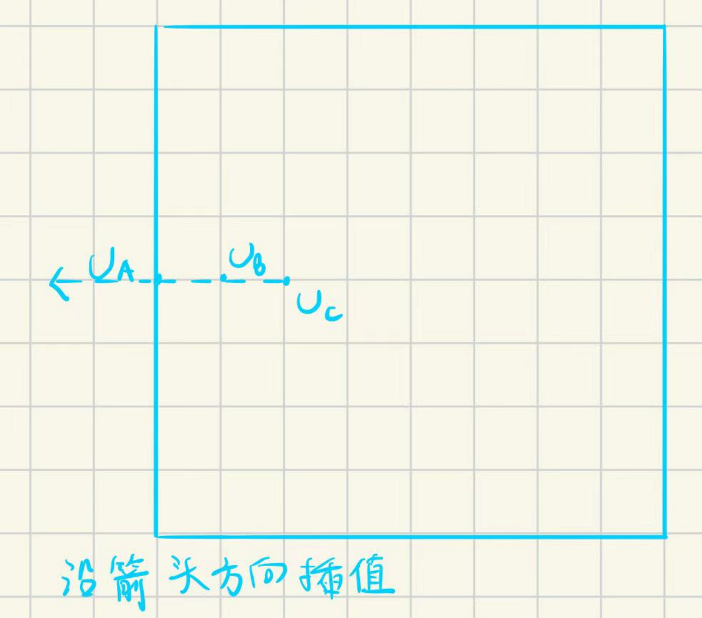

图 1: Neumann 边界条件

- **规则区域、混合边界条件:** 我设计的 Poission 方程求解器要求混合边界条件以正方形的边为单位给出，即：允许 $x=1$, $x=0$ 上提出 Neumann 边界条件，$y=1$, $y=1$ 上提出 Dirichlet 边界条件，但是不允许 $x=1$ 上部分点是 DIrichlet 边值条件，部分点是 Neumann 边值条件。对于混合边界条件，只要判断具体是上面的两种边界条件中哪一种，对应处理即可。
- **不规则区域、Dirichlet 边界:** 采用讲义上图片 2，公式（7.88）的离散方式。对于边界内的格点，直接设置值为 0。对于边界附近的点，采用公式：

$$L_hU_P := \frac{(1+\theta)U_P - U_A - \theta U_W}{\frac{1}{2}\theta(1+\theta)h^2} + \frac{(1+\alpha)U_P - U_B - \alpha U_S}{\frac{1}{2}\alpha(1+\alpha)h^2}.$$

近似拉普拉斯算子 Δ。

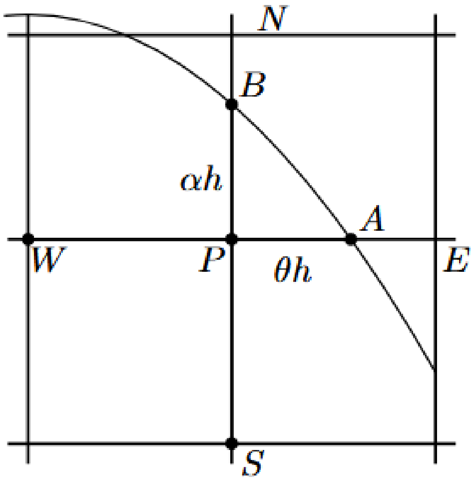

图 2: 讲义图片 7.4

- **不规则区域，Neumann 边界**：此时在规则边界上，采用之前的离散方式，在不规则边界附近的点，如下图所示：我们要近似 P 处的 $\frac{\partial^2}{\partial x^2}$，需要向圆内延申一个点 A，即此时

$$\frac{\partial^2 u}{\partial x^2}|_P = \frac{2U_P - U_A - U_C}{h^2}$$

为了求出 U_A，连接直线 AO 交格点于 T，交圆于 X，用 $U_B U_P$ 的值插值，得到 U_T，再利用

$$\frac{\partial u}{\partial \nu}|_X = \frac{U_A - U_T}{d(A,T)}$$

具体见图 3

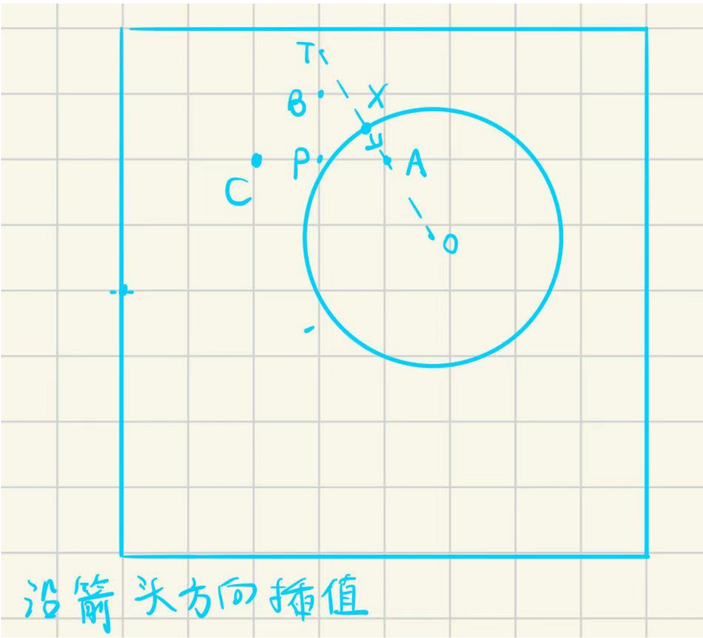

图 3: 不规则区域 Neumann 边界条件

- **不规则区域、混合边界条件**：同样地：只要判断在边界上究竟是 Neumann 边界条件还是 Dirichelt 边界条件，然后按照对应的方法离散即可。

2 程序运行结果

本次测试有三个函数，分别是 $u(x,y)=e^{y+sin(x)}$，$u(x,y)=log(x+2y+0.5)$，$u(x,y)=sin(e^x+y+1)$，由于测试用例很多，只展示对 $u(x,y)=e^{y+sin(x)}$ 的测试结果，所有测试结果的图片都保存在 `figure` 文件夹，数据保存在 `output` 文件夹。

- **规则边界，Dirichlet 边界条件：**


```
    {
      "boundary_condition": 0,
      "domain": 0,
      "l1_norm": 0.0018029833955581775,
      "l2_norm": 0.0008034791522725579,
      "linfinity_norm": 0.0005369806601569493
    },
    {
      "boundary_condition": 0,
      "domain": 0,
      "l1_norm": 0.0009259174102427131,
      "l2_norm": 0.000285369489145835,
      "linfinity_norm": 0.00013759499931875752
    },
    {
      "boundary_condition": 0,
      "domain": 0,
      "l1_norm": 0.0004660506595506736,
      "l2_norm": 0.00010100320375420975,
      "linfinity_norm": 3.445024838955035e-05
    },
    {
      "boundary_condition": 0,
      "domain": 0,
      "l1_norm": 0.0002334145927689324,
      "l2_norm": 3.571961958176341e-05,
      "linfinity_norm": 8.624139446133938e-06
    }

```


- **规则边界，Neumann 边界条件：**


```
      {
        "boundary_condition": 1,
        "domain": 0,
        "l1_norm": 0.1378620311788022,
        "l2_norm": 0.05911065964839879,
        "linfinity_norm": 0.03582693936896453
      },
      {
        "boundary_condition": 1,
        "domain": 0,
        "l1_norm": 0.12038180736898599,
        "l2_norm": 0.03311811432618216,
        "linfinity_norm": 0.012413965808491856
      },
      {
        "boundary_condition": 1,
        "domain": 0,
        "l1_norm": 0.08603561261640454,
        "l2_norm": 0.016000327648816745,
        "linfinity_norm": 0.003890650983532584
```

```
        },
        {
          "boundary_condition": 1,
          "domain": 0,
          "l1_norm": 0.05570034619663955,
          "l2_norm": 0.007162695526340437,
          "linfinity_norm": 0.0011621604459728374
        },

```

- 不规则边界，Dirichlet 边界条件：

```
        {
          "boundary_condition": 0,
          "domain": 1,
          "l1_norm": 0.0008467978248357655,
          "l2_norm": 0.0004510640402120567,
          "linfinity_norm": 0.0004572443924395486
        },
        {
          "boundary_condition": 0,
          "domain": 1,
          "l1_norm": 0.0002952036977586642,
          "l2_norm": 0.0001043678506377078,
          "linfinity_norm": 7.508401886680005e-05
        },
        {
          "boundary_condition": 0,
          "domain": 1,
          "l1_norm": 0.00011145442054616428,
          "l2_norm": 2.6232576960044132e-05,
          "linfinity_norm": 1.019374547972518e-05
        },
        {
          "boundary_condition": 0,
          "domain": 1,
          "l1_norm": 3.226979934467175e-05,
          "l2_norm": 5.513152287627239e-06,
          "linfinity_norm": 1.706298987880217e-06
        },

```

- **不规则边界，Neumann 边值条件：**

```
        {
          "boundary_condition": 1,
          "domain": 1,
          "l1_norm": 0.13788848778244225,
          "l2_norm": 0.06283645850410362,
          "linfinity_norm": 0.04618947221941205
        },
        {
          "boundary_condition": 1,
          "domain": 1,
          "l1_norm": 1.1408564488934814,
          "l2_norm": 0.34398640322369506,
          "linfinity_norm": 0.14857164575712067
        },
        {
          "boundary_condition": 1,
```

```
      "domain": 1,
      "l1_norm": 0.9267684236403655,
      "l2_norm": 0.1850332405786746,
      "linfinity_norm": 0.04965998049302822
    },
    {
      "boundary_condition": 1,
      "domain": 1,
      "l1_norm": 0.867053614219916,
      "l2_norm": 0.11938541717425366,
      "linfinity_norm": 0.021128588510432422
    },

```

3 收敛性和误差分析

程序运行结果中的 `error.json` 文件中的误差范数按照讲义中定义 7.13 的方式计算，即：误差在格点 $X := \{x_1, x_2, \ldots, x_N\}$ 的 L_q-范数 g 定义为：

$$\|g\|_{L_q} = \left(h \sum_{i=1}^{N} |g_i|^q \right)^{\frac{1}{q}}. \tag{7.15}$$

。

- **规则边界，Dirichlet 边界条件**：truncation error 来自离散拉普拉斯算子的误差，理论和实验都证明了 l_∞ 的误差是 $l_\infty = O(h^2)$。

$$\begin{aligned} R_{ij}(u) = &-\frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \\ &- \frac{u(x_i, y_{j-1}) - 2u(x_i, y_j) + u(x_i, y_{j+1})}{h^2} - f_{ij}, \end{aligned}$$

- **规则边界，Neumann 边界条件**：截断误差来自离散 Δ 算子和对边界上函数值的估。理论上二阶插值得到的误差应该是 $O(h^2)$，实际实验结果不尽人意，无法达到理论上的精度，只有 $l_\infty = O(h^{2/3})$ 的收敛精度。

- 不规则边界，Dirichlet 边界条件: 截断误差来自离散拉普拉斯算子，同样可以证明截断误差为 $O(h^2)$,$l_\infty = O(h^2)$

$$\tau_p = \frac{(1+\theta)U_p - U_A - \theta U_w}{\theta(1+\theta)h^2} + \frac{(1+\alpha)U_p - U_B - \alpha U_S}{2(1+\alpha)h^2} + \Delta u\big|_p$$

$$U_A = U_p + \theta h \frac{\partial u}{\partial x}\Big|_p + \frac{1}{2}\theta^2 h^2 \frac{\partial^2 u}{\partial x^2}\Big|_p + \frac{1}{6}\theta^3 h^3 \frac{\partial^3 u}{\partial x^3}\Big|_p + O(h^4)$$

$$U_W = U_p - h \frac{\partial u}{\partial x}\Big|_p + \frac{1}{2} h^2 \frac{\partial^2 u}{\partial x^2}\Big|_p - \frac{1}{6} h^3 \frac{\partial^3 u}{\partial x^3}\Big|_p + O(h^4)$$

$$U_B = U_p + \alpha h \frac{\partial u}{\partial y}\Big|_p + \frac{1}{2}\alpha^2 h^2 \frac{\partial^2 u}{\partial y^2}\Big|_p + \frac{1}{6}\alpha^3 h^3 \frac{\partial^3 u}{\partial y^3}\Big|_p + O(h^4)$$

$$U_S = U_p - h \frac{\partial u}{\partial y}\Big|_p + \frac{1}{2} h^2 \frac{\partial^2 u}{\partial y^2}\Big|_p - \frac{1}{6} h^3 \frac{\partial^3 u}{\partial y^3}\Big|_p + O(h^4)$$

$$\begin{aligned}\Rightarrow \tau_p &= \frac{h}{3}\frac{\theta(1-\theta^3)}{\theta(1+\theta)}\frac{\partial^3 u}{\partial x^3}\Big|_p + \frac{h}{3}\frac{2(1-\alpha)}{(1+\alpha)}\frac{\partial^3 u}{\partial y^3}\Big|_p + O(h^2) \\ &= \frac{1-\theta}{3} h \frac{\partial^3 u}{\partial x^3}\Big|_p + \left(\frac{1-\alpha}{3}\right) h \frac{\partial^3 u}{\partial y^3}\Big|_p + O(h^2)\end{aligned}$$

- **不规则边界，Neumann 边值条件**：对于 $\partial\mathbb{D}$ 附件的点，我们用了一阶插值来近似 ghost cell 的值，理论上精度是一阶的，实验数据也确实证明了 $l_\infty = O(h)$