西安交通大学实验报告

成绩	

课 程 ： _____ 实 验 日 期 年 月 日

专业班号 _____ 组 别 _____ 交 报 告 日 期 年 月 日

姓 名 _____ 学 号 _____ 报 告 退 发 （订正、重做）

同 组 者 _____ 教室审批签字

实验名称 二维导热物体温度场的计算机模拟实验

一、实验目的

(1) 学习电、热类比的原理及边界条件的处理；

(2) 通过计算机编程的方式求出墙角导热的离散温度场。

二、实验原理

二维稳态过程，导热方程为

$$\frac{\partial^2 t}{\partial x^2}+\frac{\partial^2 t}{\partial y^2}=0$$

二维稳态导热内部节点的差分方程为

$$t_{i+1,j}+t_{i-1,j}+t_{i,j+1}+t_{i,j-1}-4t_{i,j}=0$$

于是内部节点的迭代计算式为

$$t_{i,j}=\frac{t_{i+1,j}+t_{i-1,j}+t_{i,j+1}+t_{i,j-1}}{4}$$

对于恒温边界条件，除了绝热边界时使用对称性外，只使用上面一个迭代计算式即可。但是对于对流边界，边界上的点，按位置分为内角点、外角点和平直边界，按类型分为对流边界、绝热边界，计算步骤相比恒温边界下更为复杂。

按位置：

a) 内角点：4 个方向均有导热热流，有$\frac{\mathrm{dx}}{2}+\frac{dy}{2}$面积的对流换热

b) 外角点：2 个方向有导热，有$\frac{\mathrm{dx}}{2}+\frac{dy}{2}$面积的对流换热

c) 平直边界：3 个方向有导热，有dx或dy面积的对流换热

按类型：

a) 绝热边界：该点的绝热一侧没有热流量，基尔霍夫定律中，此方向的热流量代入 0 计算

b) 对流边界：该点该方向的对流换热量由牛顿冷却公式$\mathrm{q}=\mathrm{hA}(\mathrm{t}_{\infty}-\mathrm{t}_{\mathrm{i,j}})$计算得出

综上所述：

对流边界下的差分方程为：

$$\Phi_{i-1,j}+\Phi_{i+1,j}+\Phi_{i,j-1}+\Phi_{i,j+1}+\Phi_{\text{对流}}=0$$

其中，$\Phi_{i-1,j},\Phi_{i+1,j},\Phi_{i,j-1},\Phi_{i,j+1}$为导热量，$q_{\text{对流}}$为对流边界换热量。$\Phi_{i-1,j}=\frac{\lambda A\left(t_{i-1,j}-t_{i,j}\right)}{\mathrm{dx}}$，$\Phi_{\text{对流}}=hA(t_{\infty}-t_{i,j})$。

代入所有q的计算式，可解得

$$t_{i,j}=\frac{\sum_k \frac{\lambda A_k t_k}{dx}+h_{对流}A_{对流}t_\infty}{\sum_k \frac{\lambda A_k}{dx}+h_{对流}A_{对流}}$$

注意：

a) k为实际参与导热的几个方向，对于内角点有 4 项，外角点有 2 项，平直边界有 3 项，绝热边界还要去掉这一方向的那一项

b) A_k的值根据实际位置确定，内角点得两个方向为$0.5dx$两个方向为$1dx$,外角点的两个方向均为$0.5dx$，平直边界有两个$0.5dx$和一个$1dx$

c) 内外测流体的h不相等，对流面积为该网格实际与流体接触的面积角点为$0.5dx$，平直边界为$1dx$。

具体代码参见附录。

三、实验数据的整理

根据得到的温度场分布图，计算单位厚度墙体换热量Φ（W/m），如下表所示：

恒温边界	外表面	60.42871112
	内表面	60.42871112
对流边界	外表面	28.36094047
	内表面	28.36094047

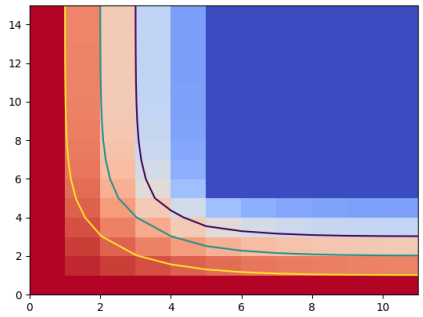
可以看出计算机模拟法的精确度是足够高的

四、附录

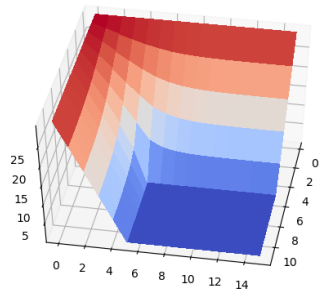
1. 实验数据

1.1 恒温边界：外温$t_1=30℃$，内温$t_2=0℃$

30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00	30.00
30.00	29.03	28.07	27.12	26.22	25.46	24.89	24.53	24.30	24.17	24.09	24.05	24.03	24.02	24.01	24.01
30.00	28.07	26.12	24.18	22.31	20.71	19.60	18.91	18.51	18.28	18.15	18.08	18.05	18.03	18.02	18.01
30.00	27.12	24.18	21.16	18.14	15.46	13.87	13.00	12.54	12.29	12.16	12.09	12.05	12.03	12.02	12.01
30.00	26.22	22.31	18.14	13.64	9.13	7.42	6.70	6.36	6.19	6.10	6.05	6.03	6.02	6.01	6.01
30.00	25.46	20.71	15.47	9.13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30.00	24.90	19.60	13.88	7.43	0.00										
30.00	24.53	18.92	13.01	6.70	0.00										
30.00	24.31	18.52	12.55	6.36	0.00										
30.00	24.18	18.30	12.32	6.20	0.00										
30.00	24.12	18.20	12.20	6.13	0.00										
30.00	24.10	18.17	12.17	6.11	0.00										



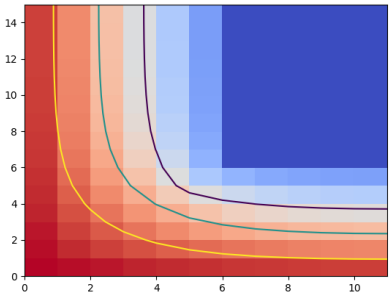
24℃, 18℃, 12℃等温曲线图



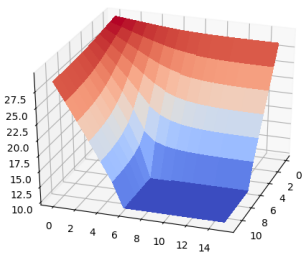
x − y − t 三维图像

1.2 对流边界：外温$t_1 = 30℃$，内温$t_2 = 10℃$

29.90	29.71	29.52	29.33	29.15	28.99	28.86	28.75	28.68	28.63	28.60	28.58	28.56	28.55	28.55	28.55
29.71	29.14	28.56	27.99	27.45	26.96	26.55	26.25	26.03	25.88	25.79	25.72	25.68	25.66	25.65	25.64
29.52	28.56	27.60	26.63	25.69	24.84	24.15	23.65	23.31	23.09	22.94	22.85	22.79	22.76	22.74	22.73
29.33	27.99	26.63	25.24	23.84	22.55	21.57	20.90	20.48	20.21	20.04	19.94	19.87	19.84	19.82	19.81
29.15	27.45	25.69	23.85	21.90	19.95	18.66	17.91	17.48	17.23	17.08	16.99	16.93	16.90	16.88	16.88
28.99	26.96	24.84	22.56	19.96	16.70	15.20	14.61	14.31	14.15	14.06	14.00	13.96	13.94	13.93	13.93
28.86	26.56	24.17	21.58	18.67	15.21										
28.76	26.26	23.68	20.93	17.93	14.62										
28.69	26.06	23.35	20.52	17.52	14.34										
28.65	25.93	23.15	20.28	17.29	14.19										
28.62	25.85	23.04	20.15	17.17	14.11										
28.61	25.83	23.00	20.11	17.14	14.09										



26℃, 22℃, 18℃等温曲线图



x − y − t 三维图像

2. 计算程序源代码

2.1 Python 源代码

```
# python main.py           # 恒温边界
# python main.py convection # 对流边界

import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
import sys

X = np.arange(0, 16)
Y = np.arange(0, 12)
X, Y = np.meshgrid(Y, X)
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
# ax = fig.gca(projection='3d')
t = np.zeros((16, 12))


conduction = True
for i in sys.argv:
    if i == 'convection':
        conduction = False
        break

h1 = 10.6
lam = 0.53
dx = 0.1
lambda_over_dx = lam / dx
h2 = 3.975

if conduction:
    t1 = 30
    t2 = 0
    t[0, :] = t1
    t[:, 0] = t1
    t[5:16, 5:12] = t2
else:
    t1 = 30
    t2 = 10
    t[0, :] = t1
    t[:, 0] = t1
    t[5:16, 5:12] = t2

if conduction:
    for k in range(0, 1000):

        for m in range(1, 5):
            for n in range(1, 11):
                t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] + t[m, n + 1]) / 4

            n = 11
            t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] * 2) / 4

        for m in range(5, 15):
            for n in range(1, 5):
                t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] + t[m, n + 1]) / 4

        m = 15
        for n in range(1, 5):
            t[m, n] = (t[m - 1, n] * 2 + t[m, n - 1] + t[m, n + 1]) / 4

else:
    for k in range(0, 1000):

        # m = 0
        m = 0
        n = 0
        t[m, n] = (h1 * t1 + lambda_over_dx * (t[m + 1, n] / 2 + t[m, n + 1] / 2)) / (h1 + lambda_over_dx)

        for n in range(1, 11):
            t[m, n] = (h1 * t1 + lambda_over_dx * (t[m + 1, n] + t[m, n - 1] / 2 + t[m, n + 1] / 2)) / (h1 + lambda_over_dx * 2)

        n = 11
        t[m, n] = (h1 * t1 / 2 + lambda_over_dx * (t[m + 1, n] / 2 + t[m, n - 1] / 2)) / (h1 / 2 + lambda_over_dx)

        # m = 1 .. 4
        for m in range(1, 5):
            n = 0
            t[m, n] = (h1 * t1 + lambda_over_dx * (t[m - 1, n] / 2 + t[m + 1, n] / 2 + t[m, n + 1])) / (h1 + lambda_over_dx * 2)

            for n in range(1, 11):
                t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] + t[m, n + 1]) / 4

            n = 11
            t[m, n] = (t[m - 1, n] / 2 + t[m + 1, n] / 2 + t[m, n - 1]) / 2

        # m = 5
        m = 5
        n = 0
        t[m, n] = (h1 * t1 + lambda_over_dx * (t[m - 1, n] / 2 + t[m + 1, n] / 2 + t[m, n + 1])) / (h1 + lambda_over_dx * 2)

        for n in range(1, 5):
            t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] + t[m, n + 1]) / 4

        n = 5
        t[m, n] = (h2 * t2 + lambda_over_dx * (t[m - 1, n] + t[m + 1, n] / 2 + t[m, n - 1] + t[m, n + 1] / 2)) / (h2 + lambda_over_dx
* 3)

        for n in range(6, 11):
```

```
            t[m, n] = (h2 * t2 + lambda_over_dx * (t[m - 1, n] + t[m, n - 1] / 2 + t[m, n + 1] / 2)) / (h2 + lambda_over_dx * 2)

        n = 11
        t[m, n] = (h2 * t2 / 2 + lambda_over_dx * (t[m - 1, n] / 2 + t[m, n - 1] / 2)) / (h2 / 2 + lambda_over_dx)

        # m = 6 .. 14
        for m in range(6, 15):
            n = 0
            t[m, n] = (h1 * t1 + lambda_over_dx * (t[m - 1, n] / 2 + t[m + 1, n] / 2 + t[m, n + 1])) / (h1 + lambda_over_dx * 2)

            for n in range(1, 5):
                t[m, n] = (t[m - 1, n] + t[m + 1, n] + t[m, n - 1] + t[m, n + 1]) / 4

            n = 5
            t[m, n] = (h2 * t2 + lambda_over_dx * (t[m - 1, n] / 2 + t[m + 1, n] / 2 + t[m, n - 1])) / (h2 + lambda_over_dx * 2)

        # m = 15
        m = 15
        n = 0
        t[m, n] = (h1 * t1 / 2 + lambda_over_dx * (t[m - 1, n] / 2 + t[m, n + 1] / 2)) / (h1 / 2 + lambda_over_dx)

        for n in range(1, 5):
            t[m, n] = (t[m - 1, n] + t[m, n - 1] / 2 + t[m, n + 1] / 2) / 2

        n = 5
        t[m, n] = (h2 * t2 / 2 + lambda_over_dx * (t[m - 1, n] / 2 + t[m, n - 1] / 2)) / (h2 / 2 + lambda_over_dx)

# surf = ax.plot_surface(X, Y, t, cmap=cm.coolwarm, linewidth=0, antialiased=False)
ax.pcolormesh(X, Y, t, cmap=cm.coolwarm)
if conduction:
    ax.contour(X, Y, t, [12, 18, 24])
else:
    ax.contour(X, Y, t, [18, 22, 26])
plt.show()

t = np.transpose(t)
if conduction:
    np.savetxt("conduction.csv", t, delimiter=",")
else:
    np.savetxt("convection.csv", t, delimiter=",")
```

2.2 运行方法：

计算恒温边界：执行 python main.py

计算对流边界：执行 python main.py convection