

# 并行数值方法大作业中期报告

陳日康 信息与计算科学 22336049

## 一、问题理解

本次作业的目标是并行求解一个由热传导问题离散化得到的线性方程组  $Ax=b$ 。其中稀疏矩阵  $A$  以 CSR 格式存储在 `Matrix.dat` 文件中，右端向量  $b$  为零向量。不需要自行构建方程组，而是重点考察如何高效读取数据、并行构建求解器并验证结果的数值精度，最终输出结果至 `Result.dat` 并验证残差范数是否满足  $\leq 10^{-6}$ 。

## 二、当前进展

目前已完成的主要内容包括矩阵读取模块的实现与调试。通过编写 `readMatrix()` 函数，成功从文件中提取出矩阵的非零元数组 `nA`、列指标数组 `JA` 与行指针数组 `IA`，并将右端项向量 `b`。

在求解方法方面，我实现了 `Jacobi` 方法和共轭梯度法两种迭代求解器，并使用 `OpenMP` 进行了并行加速，现在可以通过命令行参数传入线程数量，让程序以指定的并行度运行，还加上了残差范数  $\|Ax-b\|$  以检验解的精度计算，并能将解向量写入文件进行保存。通过初步实验，`Jacobi` 方法在收敛速度上较慢，而共轭梯度法在迭代次数控制在合理范围内时，能够达到较高的求解精度和较短的运行时间，收敛效果较好，能满足残差要求。因此，我会选择使用共轭梯度法进行实验。此外，程序还实现了解向量的输出功能，自动将结果保存至 `Result.dat` 文件。

```
chenih@Chenih5: ~/DOPNM
chenih@Chenih5:~/DOPNM$ g++ -std=c++11 -O2 -fopenmp main.cpp -o solve
chenih@Chenih5:~/DOPNM$ ./solve 4
读取完成, 矩阵维度: 628813, 非零元个数: 3045143
前5个非零元:
nA[0] = 0.0154441, JA = 0
nA[1] = -0.0036142, JA = 410872
nA[2] = -0.00436601, JA = 161935
nA[3] = -0.00335591, JA = 411920
nA[4] = -0.00410802, JA = 577567

IA 指针前5项:
IA[0] = 0
IA[1] = 5
IA[2] = 9
IA[3] = 14
IA[4] = 19
IA[5] = 24

b 向量前5项:
b[0] = 0
b[1] = 0
b[2] = 0
b[3] = 0
b[4] = 0

运行耗时: 11.8105 秒
最终残差范数 ||Ax - b|| = 0.659356
chenih@Chenih5:~/DOPNM$
```

Jacobi 方法下残差较大, 不满足精度要求。

```
chenih@Chenih5: ~/DOPNM
chenih@Chenih5:~/DOPNM$ g++ -std=c++11 -O2 -fopenmp main.cpp -o solve
chenih@Chenih5:~/DOPNM$ ./solve 4
读取完成, 矩阵维度: 628813, 非零元个数: 3045143
前5个非零元:
nA[0] = 0.0154441, JA = 0
nA[1] = -0.0036142, JA = 410872
nA[2] = -0.00436601, JA = 161935
nA[3] = -0.00335591, JA = 411920
nA[4] = -0.00410802, JA = 577567

IA 指针前5项:
IA[0] = 0
IA[1] = 5
IA[2] = 9
IA[3] = 14
IA[4] = 19
IA[5] = 24

b 向量前5项:
b[0] = 0
b[1] = 0
b[2] = 0
b[3] = 0
b[4] = 0

CG 收敛于第 1585 次迭代, 残差 = 9.71459e-07
运行耗时: 18.0083 秒
最终残差范数 ||Ax - b|| = 9.7146e-07
chenih@Chenih5:~/DOPNM$
```

共轭梯度法收敛良好, 残差符合要求。

### 三、后续计划

接下来我打算继续完善共轭梯度法的数值稳定性，测试更多不同线程数下的表现，并记录运行时间和收敛情况，为后续实验报告提供更多数据。最后还可能考虑将结果做一些可视化展示，比如还原为二维网格的形式，看一下解的空间分布情况。