数值代数第二次作业上机报告

陈奕行

October 7, 2019

1 程序使用说明

- poisson_choice.m 文件包含 poisson_choice 函数。输入 (f,n,choice) 中的 f 为满足 $-\Delta u = f$ 的二元函数；$\frac{1}{n}$ 为两个维度的网格步长；字符串 choice 表示求解离散方程的方法，若为'gauss' 则采取 Gauss 消去法，若为'gauss_band' 则采取带状高斯消去法，若为'cholesky' 则采取改进的平方根法。其中 'gauss_band' 函数及'cho' 均输入 m 作为半带宽，即带宽为 $2*m+1$。在 Poisson 方程中，带宽为 $2\times n+1$，于是我们令 $m=n$.

- 在构建迭代矩阵 A 时，对于带状 Gauss 以及改进的 Cholesky 算法。我们采取稠密矩阵初始化。对于 Guass 消去，我们则采取稀疏矩阵初始化。取 N=99 的实验证明，采用稀疏矩阵能大幅减少 Gauss 消去用时[1]，而采取稠密矩阵则能大幅减少带状 Gauss 消去及改进的 Cholesky 算法用时[2]。

- poisson_choice.m 中给出了可视化解的方案，但为了防止对问题的干扰，我们将其作为注释，仅仅在单独使用时可以将此代码块取消注释以展示离散解的形状。

- poisson_choice.m 的输出为 $[u,t]$，其中 $\vec{u}$ 为离散解 U 的向量化，储存在 $(n-1)^2$ 维向量之中，且 $\vec{u}((n-1)*i+j)=U_{ij}$；$t$ 为解方程 $Ax=b$ 一步所需的时间，而非程序运行总时间。

- plot_time.m 则用来给出输出。注意到尽管题目中未要求误差分析，但我们这里仍然使用 real_sol.m 给出真解，并计算其与数值解的 l_∞ 范数。[3]我们将给出等分数 N 与用时 t 的关系，及等分数 N 与误差 err 的关系，以说明此程序的效率及精度。plot_time

[1]约需 180 秒，若用稠密则运行 5min 后仍未得出结果，且距离得出结果较远。

[2]约为 1 秒，若用稀疏则需 1min 以上。

[3]由于对于一般的 f 真解是未知的，因而我们并不将其整合在 poisson_choice.m 中。

和 plot_time_2.m 的区别是前者处理 Gauss 消去 11 个数据点，后者处理另外两种方法 91 个数据点。

2 分析

根据下一节的结果可知，误差随着网格数增多减小，证明此数值方法的一致性。计算用时随 N 的增加趋势也与理论分析相一致。

3 结果展示

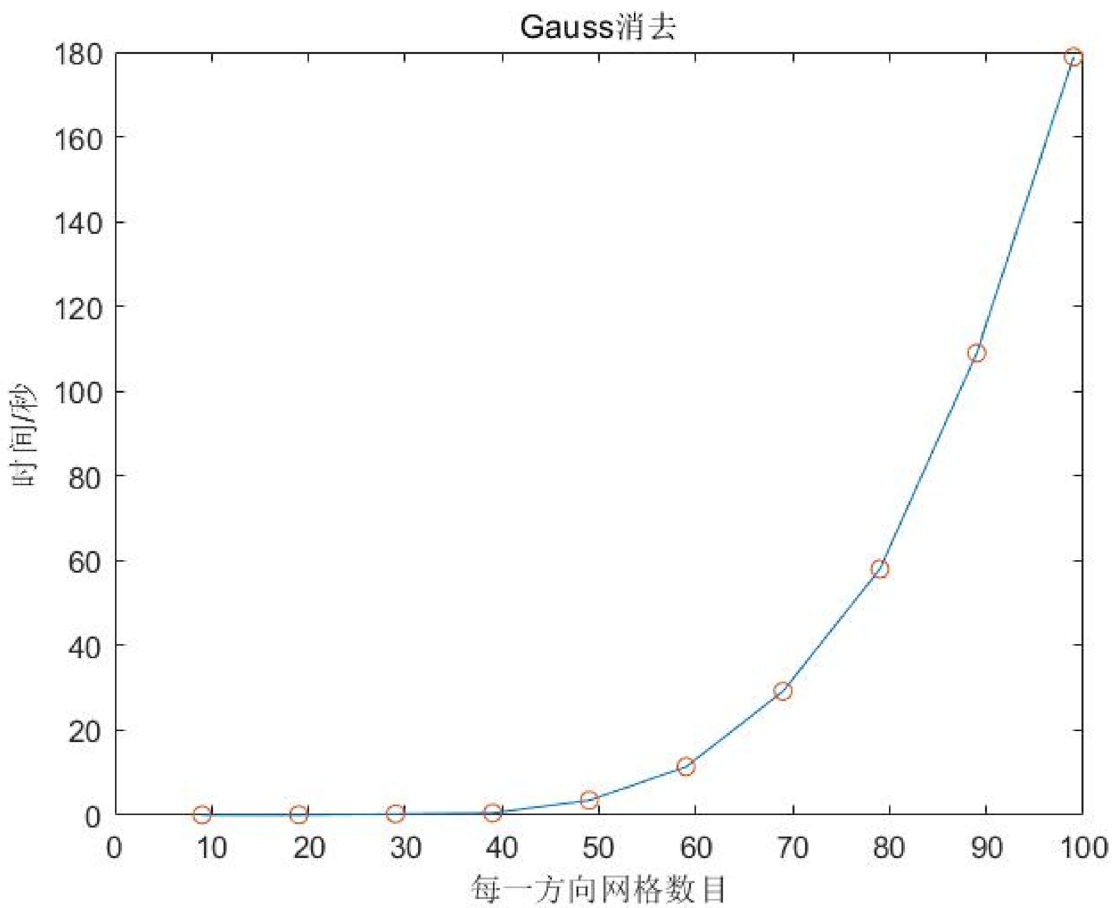

Figure 1: Gauss 消去用时

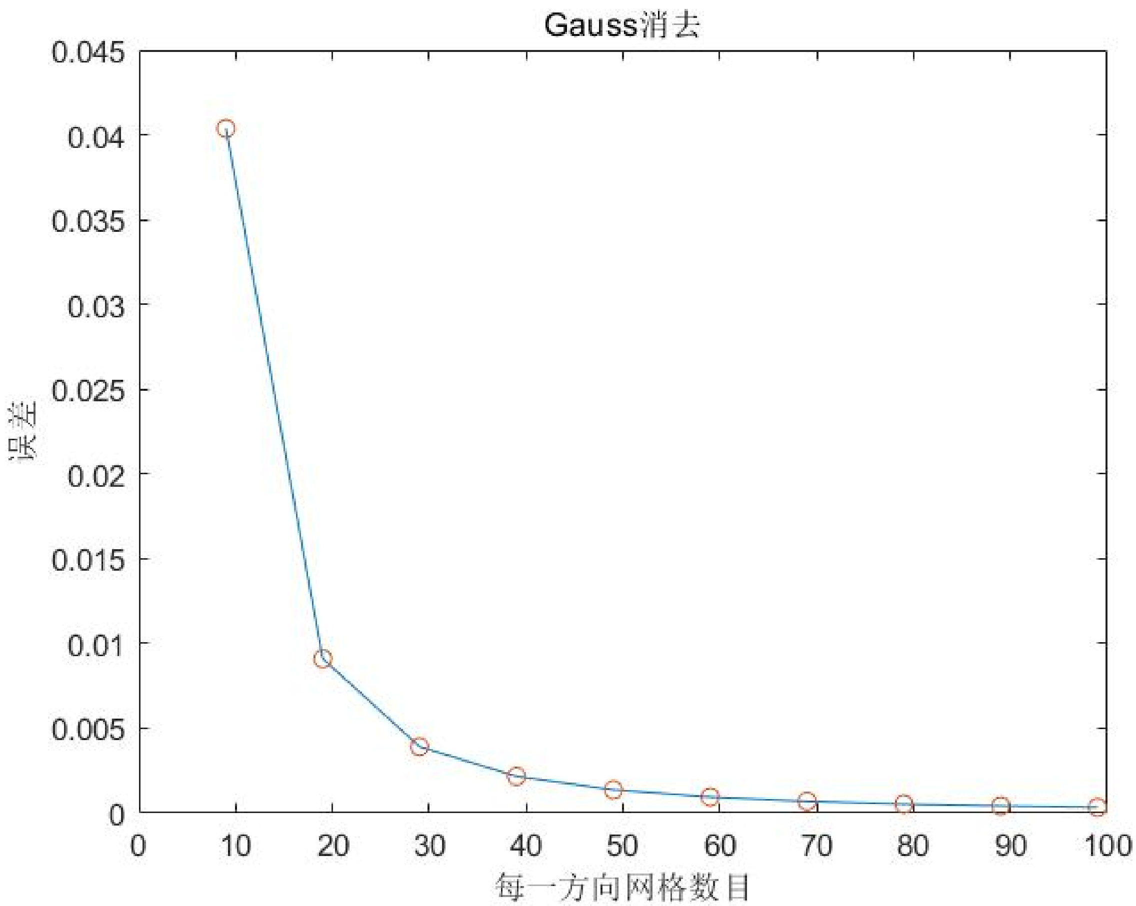

Figure 2: Gauss 消去误差

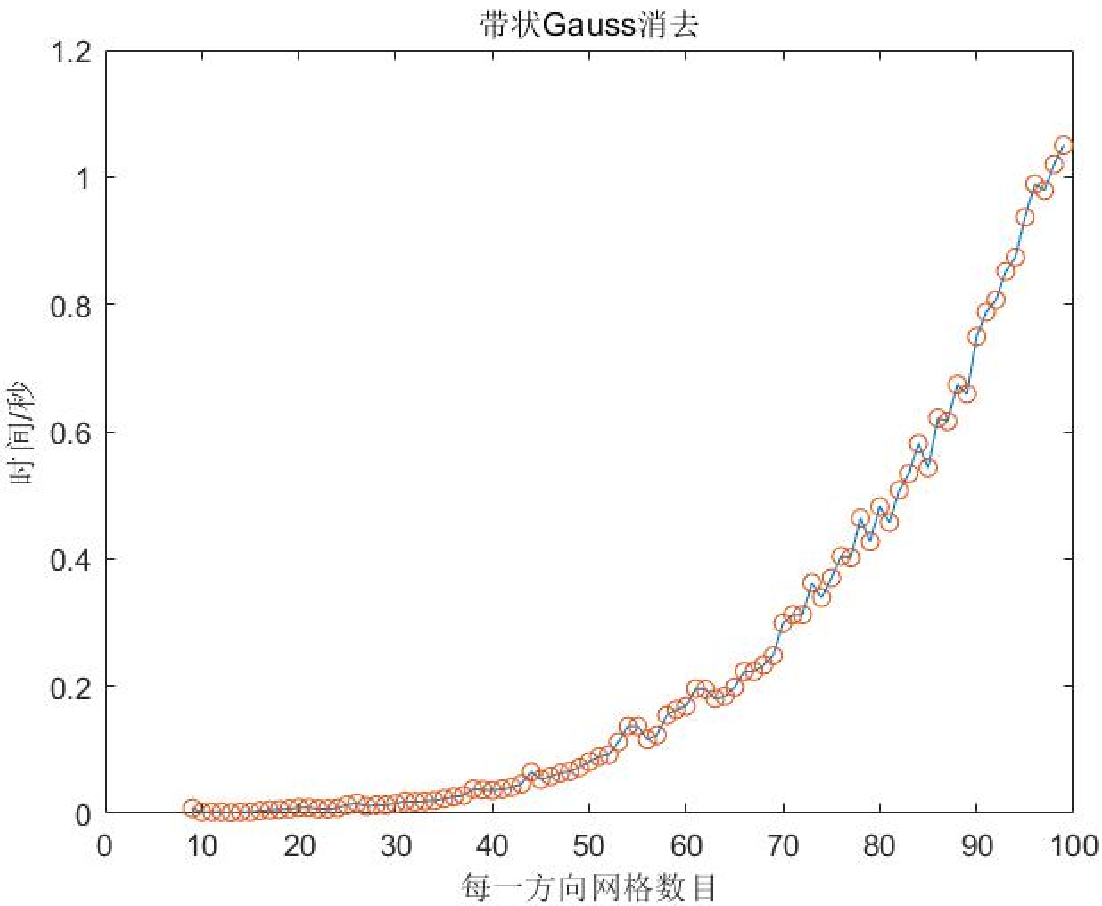

Figure 3: 带状 Gauss 消去用时

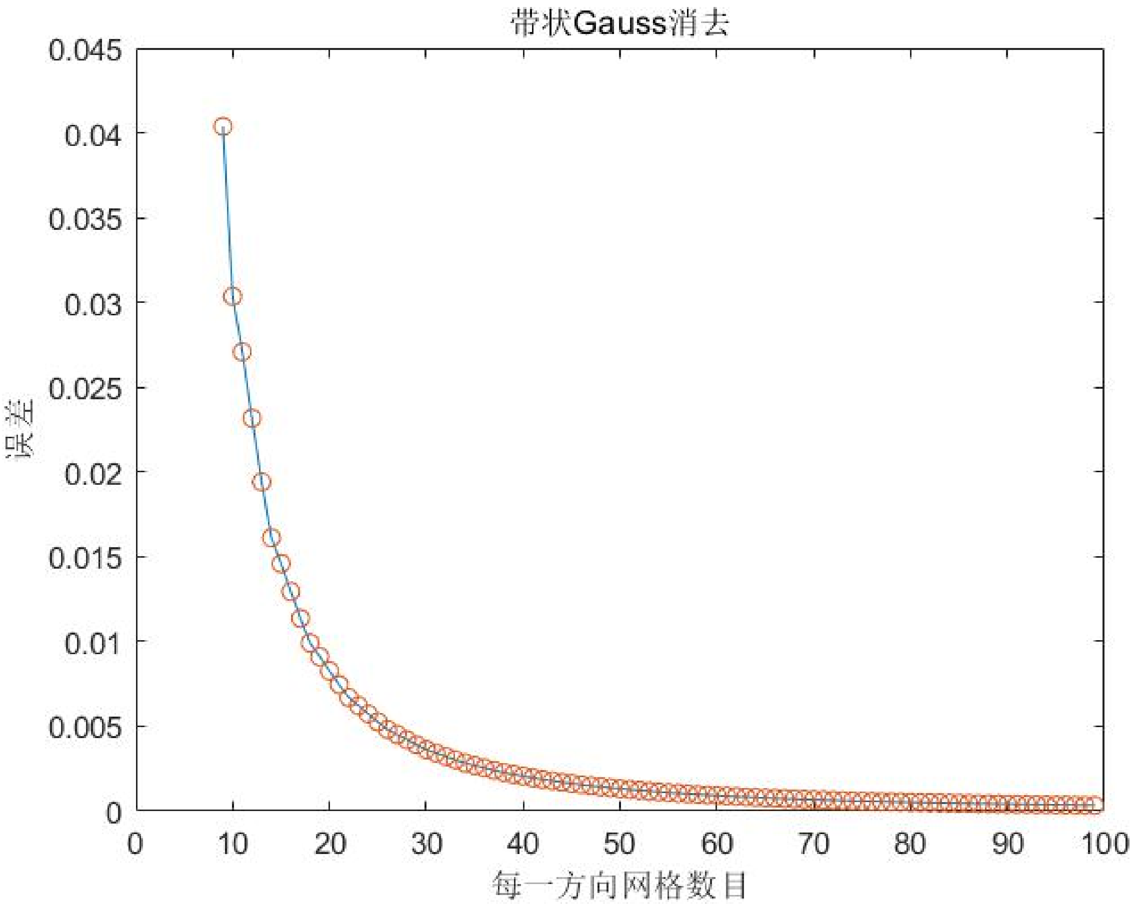

Figure 4: 带状 Gauss 消去误差

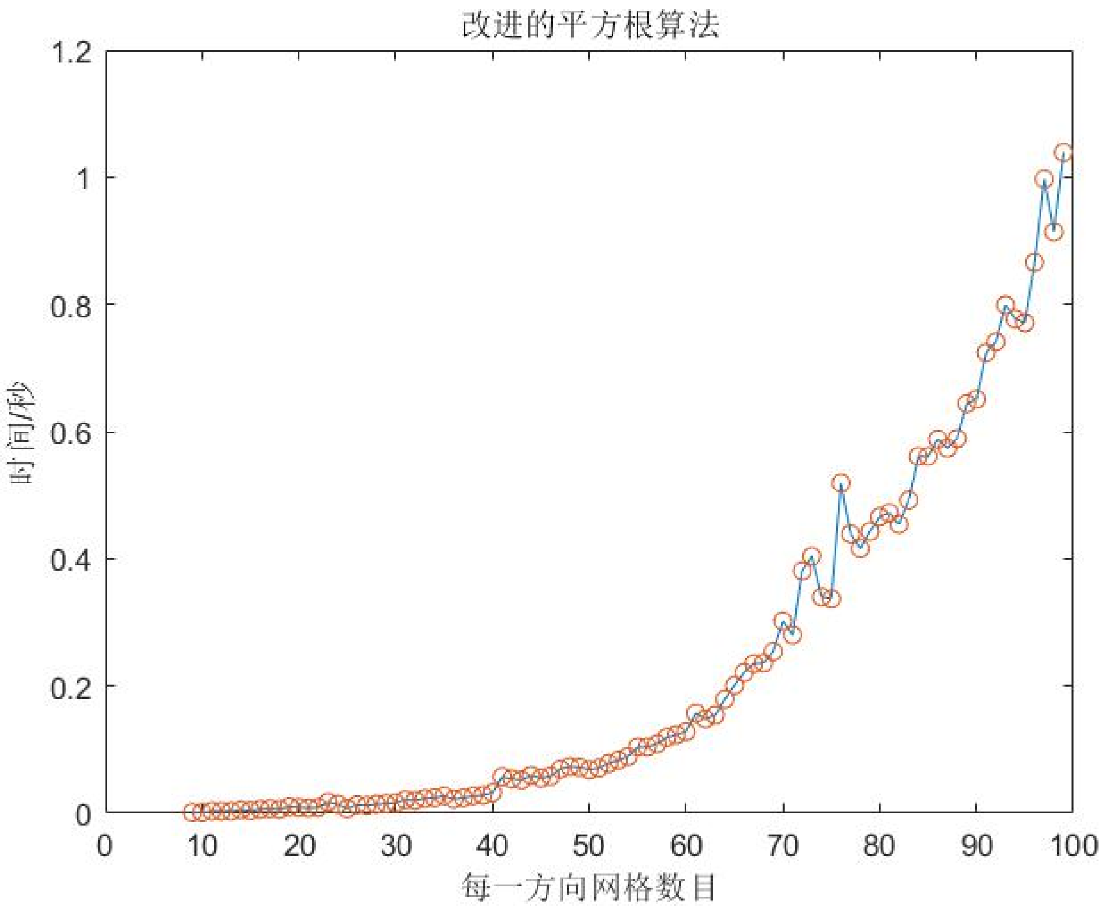

Figure 5: 改进的平方根算法用时

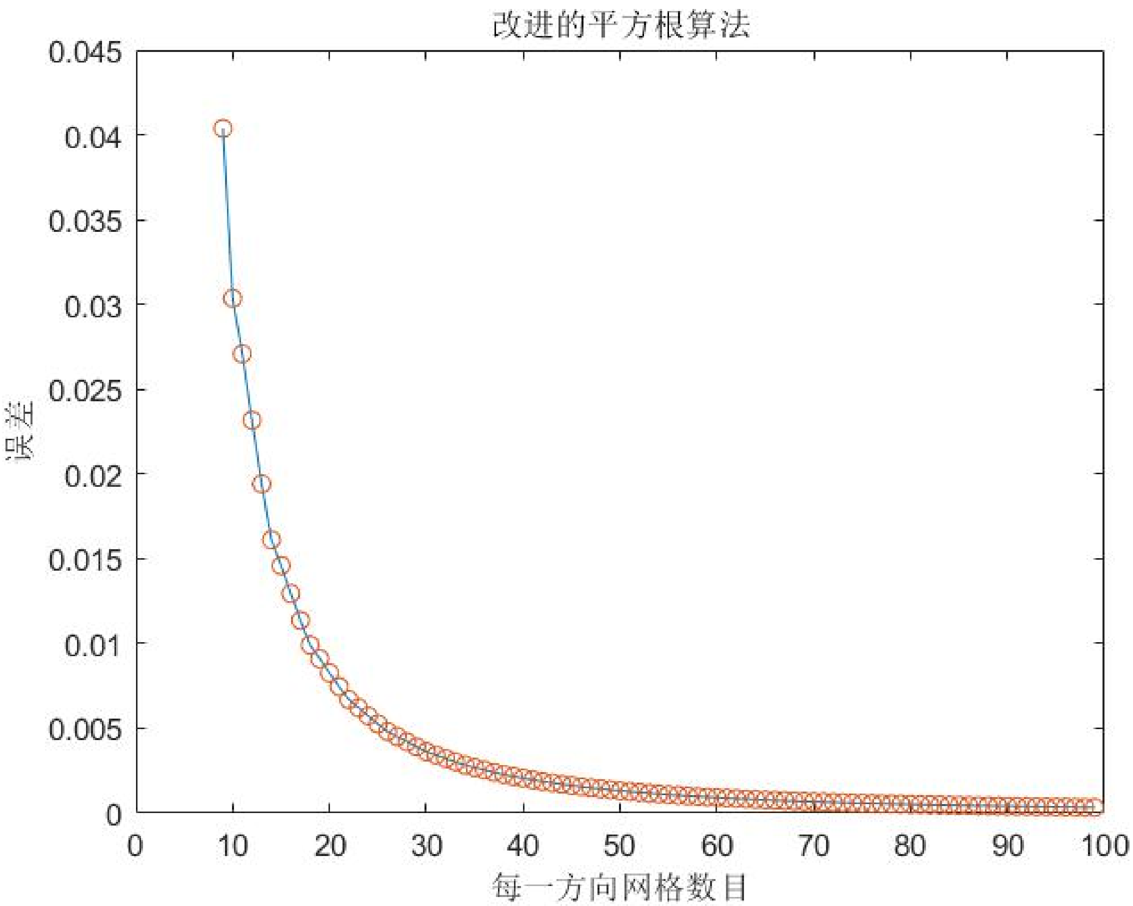

Figure 6: 改进的平方根算法误差