并行与分布式计算基础 2021 秋季第二次作业
Gauss-Seidel 迭代法求解 Poisson 方程

叶子凌锋　罗昊

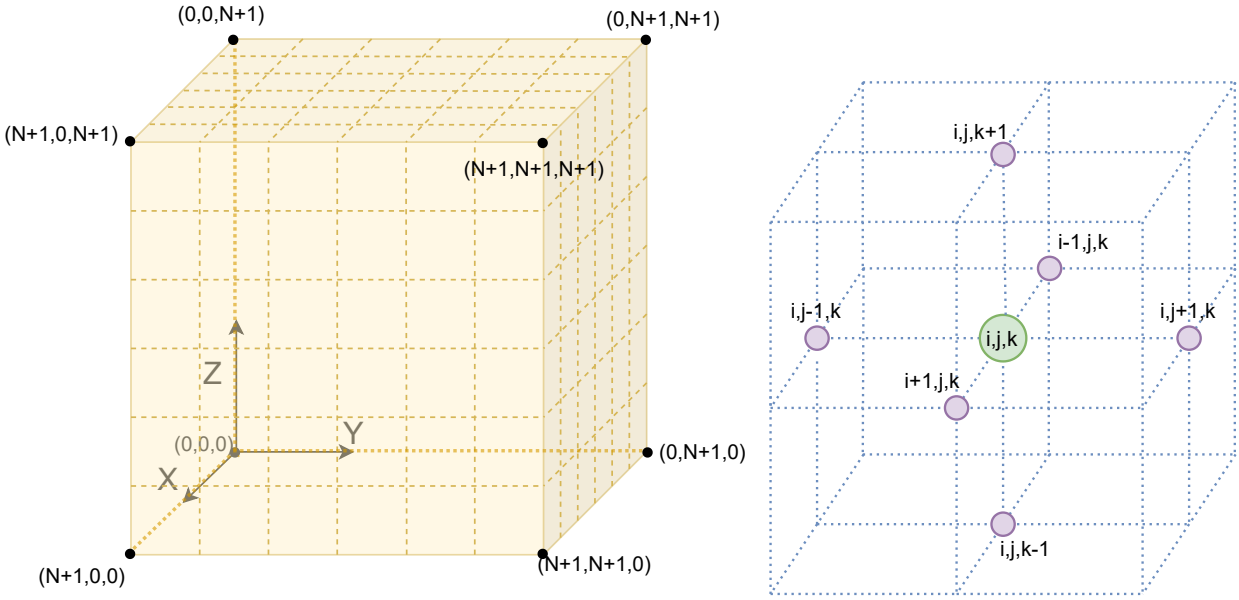
2021 年 11 月 30 日

1　Poisson 方程的离散格式

考虑在 $\Omega = [0,1] \times [0,1] \times [0,1]$ 上的 Poisson 方程:

$$\begin{cases} -\Delta u(x,y,z) = f(x,y,z), & (x,y,z) \in \Omega \\ u(x,y,z) = 0, & (x,y,z) \in \partial\Omega \end{cases} \tag{1}$$

采用均匀矩形网格离散. 用 7 点差分格式来求解该问题.



(a) 三维均匀矩形网格　　(b) 三维 7 点差分

图 1: 网格及差分格式

将三边分别等分为 $N+1$ 份, 网格步长 $h = 1/(N+1)$. 则 $x_i = ih, y_j = jh, z_k = kh$, 其中 $i,j,k = 0,\ldots,N+1$. 并记 $U_{i,j,k} = U(x_i,y_j,z_k)$. 采用二阶中心差分格式离散得到如下方程组:

$$\begin{cases} 6U_{i,j,k} - U_{i-1,j,k} - U_{i+1,j,k} - U_{i,j-1,k} - U_{i,j+1,k} - U_{i,j,k-1} - U_{i,j,k+1} = h^2 f_{i,j,k}, & 1 \le i,j,k \le N \\ U_{0,j,k} = U_{N+1,j,k} = U_{i,0,k} = U_{i,N+1,k} = U_{i,j,0} = U_{i,j,N+1} = 0, & 0 \le i,j,k \le N+1 \end{cases} \tag{2}$$

其中变量有 N^3 个, 为 $U_{i,j,k}$, $1 \le i,i,j \le N$.

2 求解线性方程组的古典迭代法

设我们要求解 n 维线性方程组 $Ax = b$. 设 $A = D - L - U$, 其中 D 为对角阵, L 为对角元为 0 的下三角阵, U 为对角元为 0 的上三角阵.

2.1 Jacobi 迭代法

Jacobi 迭代法的迭代格式为:

$$x^* = D^{-1}(L + U)x + D^{-1}b \tag{3}$$

具体而言, 相当于每步求解如下关于 x_i^*, $i = 1, \ldots, n$ 的线性方程组:

$$\begin{array}{llll} a_{11}x_1^* & +a_{12}x_2 & +\ldots & +a_{1n}x_n & = b_1 \\ a_{21}x_1 & +a_{22}x_2^* & +\ldots & +a_{2n}x_n & = b_2 \\ \ldots & \ldots & \ldots & \ldots & = \ldots \\ a_{n1}x_1 & +a_{n2}x_2 & +\ldots & +a_{nn}x_n^* & = b_n \end{array} \tag{4}$$

这是一个对角型方程组, x_i^* 之间互不依赖, 因此可以并行求出 x_i^*, $i = 1, \ldots, n$.

2.2 Gauss-Seidel 迭代法

Gauss-Seidel 迭代法的迭代格式为:

$$x^* = (D - L)^{-1}(U)x + (D - L)^{-1}b \tag{5}$$

具体而言, 相当于每步求解如下关于 x_i^*, $i = 1, \ldots, n$ 的线性方程组:

$$\begin{array}{llll} a_{11}x_1^* & +a_{12}x_2 & +\ldots & +a_{1n}x_n & = b_1 \\ a_{21}x_1^* & +a_{22}x_2^* & +\ldots & +a_{2n}x_n & = b_2 \\ \ldots & \ldots & \ldots & \ldots & = \ldots \\ a_{n1}x_1^* & +a_{n2}x_2^* & +\ldots & +a_{nn}x_n^* & = b_n \end{array} \tag{6}$$

这是一个下三角方程组, 需要按照 $i = 1, \ldots, n$ 顺序依次求出 x_i^*.

3 使用 Gauss-Seidel 迭代法求解 Poisson 方程离散问题

Poisson 方程离散得到的矩阵 A 是一个稀疏矩阵, 其 Gauss-Seidel 迭代相当于每步求解如下关于 $U_{i,j,k}^*$, 其中 $1 \le i, j, k \le N$ 的线性方程组:

$$\begin{cases} 6U_{i,j,k}^* - U_{i-1,j,k}^* - U_{i+1,j,k} - U_{i,j-1,k}^* - U_{i,j+1,k} - U_{i,j,k-1}^* - U_{i,j,k+1} = h^2 f_{i,j,k}, & 1 \le i, j, k \le N \\ U_{0,j,k}^* = U_{N+1,j,k}^* = U_{i,0,k}^* = U_{i,N+1,k}^* = U_{i,j,0}^* = U_{i,j,N+1}^* = 0, & 0 \le i, j, k \le N+1 \\ U_{0,j,k} = U_{N+1,j,k} = U_{i,0,k} = U_{i,N+1,k} = U_{i,j,0} = U_{i,j,N+1} = 0, & 0 \le i, j, k \le N+1 \end{cases} \tag{7}$$

实际操作过程中, 按照 $i = 1, \ldots, N$, $j = 1, \ldots, N$, $k = 1, \ldots, N$ 的顺序依次更新 $U_{i,j,k}$ 的值即可. 图 2 展示了三维网格中一部分点的更新顺序 (依赖关系).

在完全保持 Gauss-Seidel 依赖顺序的情况下进行并行加速有一定难度, 而**通过去除或改变其中部分依赖关系, 可使得算法更易于并行, 但可能会降低收敛速度**. 如果去除所有依赖关系, 则该算法退化为

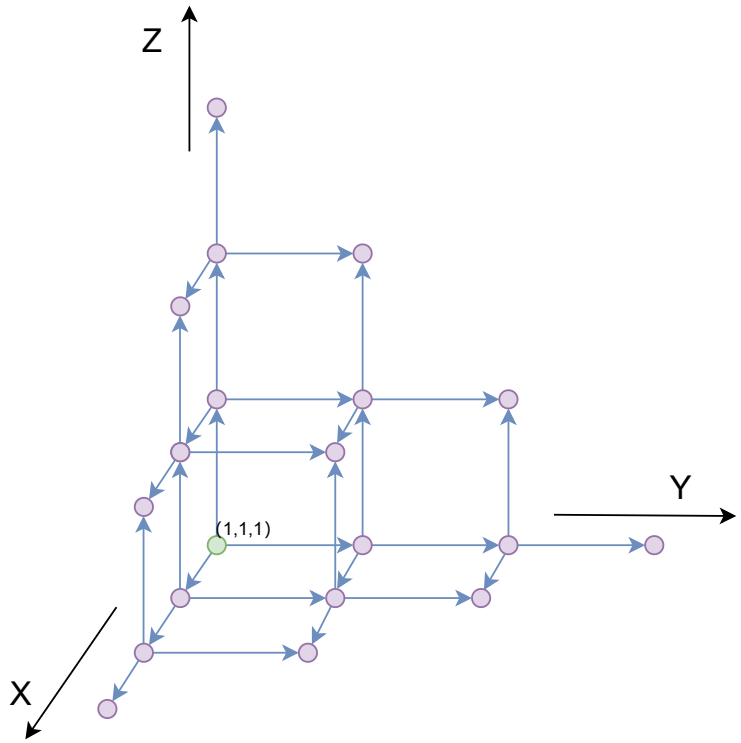


图 2: 使用 Gauss-Seidel 迭代法求解三维矩形网格上的 Poisson 方程组的依赖顺序

Jacobi 迭代法. 常见方法为直接分块、红黑着色、分块着色等等 (见图3), 参考文献中提到了其中部分方法, **仅供参考**, 建议大家自己设计并行方案.

此外, 每一步迭代后都需要计算残差范数, 以确定是否达到收敛准则. 记 $\|r^{(t)}\|$ 为第 t 次迭代后解的残差范数, 计算公式为:

$$\|r^{(t)}\| = \left(\sum_{1\le i,j,k\le N} \left(6U_{i,j,k}^{(t)} - U_{i-1,j,k}^{(t)} - U_{i+1,j,k}^{(t)} - U_{i,j-1,k}^{(t)} - U_{i,j+1,k}^{(t)} - U_{i,j,k-1}^{(t)} - U_{i,j,k+1}^{(t)} - h^2 f_{i,j,k} \right)^2 \right)^{\frac{1}{2}} \tag{8}$$

4 任务描述

本例要求同学使用 Gauss-Seidel 迭代求解离散的 Poisson 方程, 允许去除或改变 Gauss-Seidel 迭代的部分依赖以便于并行. 选取 $N = 512$, 右端项 $f(x, y, z) = 49152 \sin(128\pi x) \sin(128\pi y) \sin(128\pi z)$, 真解 $u(x, y, z) = \sin(128\pi x) \sin(128\pi y) \sin(128\pi z)$.

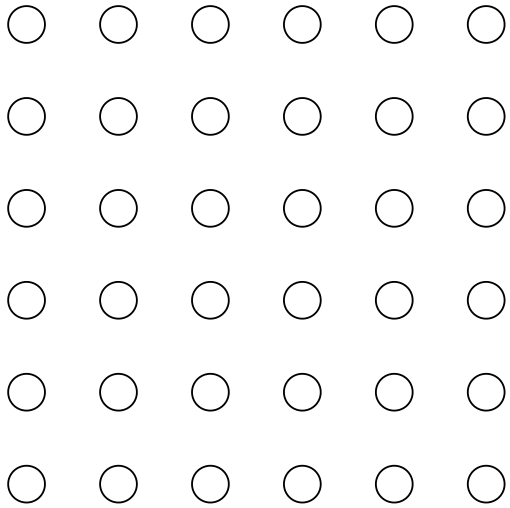
迭代初值为 $U_{i,j,k} = 0$, 收敛标准为 $\|r^{(t)}\| < 10^{-6}\|r^{(0)}\|$.

环境要求: 基于数院集群, 最多使用 1 个节点, 8 个 CPU 核心或 1 块 GPU; 自由选择使用 MPI, OpenMP 和 CUDA. 必须使用集群上提供的编译器.

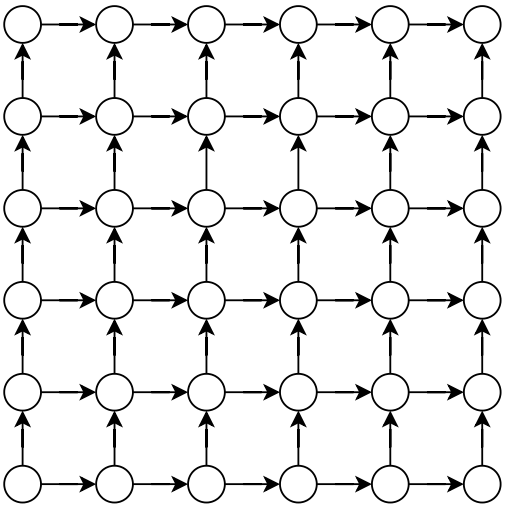
计时要求: **从迭代开始到残差达到收敛标准的总时间**, 且至少进行 34 次迭代. gauss_seidel 函数和 residual_norm 函数均需要纳入计时. 如使用 CUDA, 则在 CUDA 部分的代码里需要记录 kernel 时间, 不需记录数据在迭代开始和迭代结束时两次在 CPU 和 GPU 之间拷贝的时间.[1]

报告要求: 在报告中描述清楚你所使用的并行加速方案, 清晰展示程序优化后的计时结果.

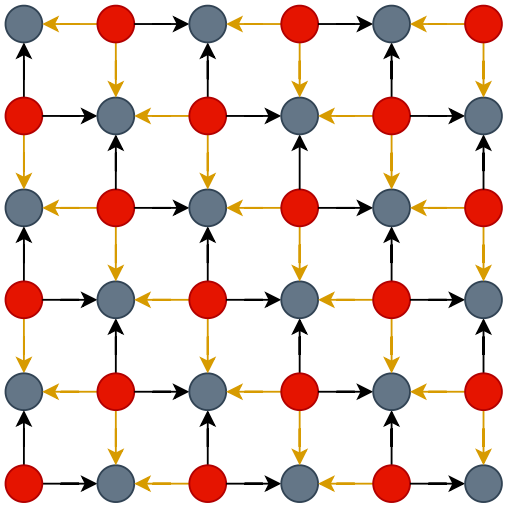
[1] 2021.12.4 修正. 即如果迭代过程中发生了数据在 CPU 和 GPU 之间的拷贝, 也是需要纳入计时的.



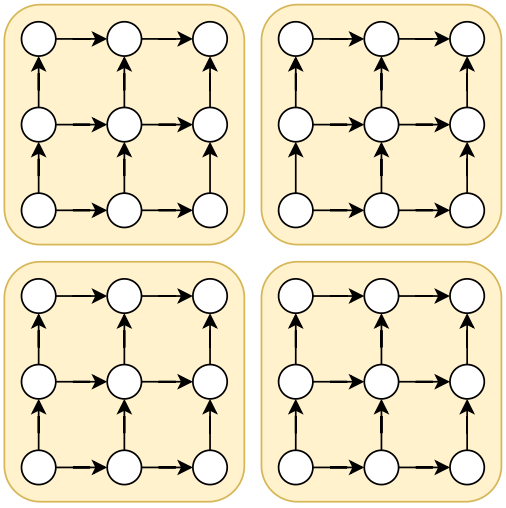
(a) Jacobi 迭代依赖顺序



(b) Gauss-Seidel 原始依赖顺序



(c) 红黑着色重排序的 Gauss-Seidel



(d) 分块 Jacobi(块内 Gauss-Seidel)

图 3: 二维矩形网格上的 5 点差分格式的 Jacobi 迭代的依赖顺序、Gauss-Seidel 迭代的原始依赖顺序、红黑着色重排序的 Gauss-Seidel 的依赖顺序、分块 Jacobi（块内 Gauss-Seidel）的依赖顺序. 展示此例仅用于帮助大家理解 **“去除或改变其中部分依赖关系”** 的可能形式.

代码要求: 提交源代码, 编译脚本和运行脚本. 可以根据需要修改本次作业提供的参考程序, 主要保证计时部分代码的符合上述计时要求.

如有其他疑问请咨询课程助教.

5 提交要求与评分标准

将代码和文档打包后上传至 http://mu2.davidandjack.cn:8888/ , 压缩包命名为 “学号-hw2” , 在 2022 **年** 1 **月** 14 **日** 23 **点** 59 **分**前提交. 每位同学提交次数限制为 3 次.

评分标准:(1) 性能是主要评价因素, 从迭代开始到残差达到收敛标准的总时间越短越好;(2) 根据文档内容丰富程度、代码质量酌情加分.

6 其他注意事项

1. 提供的参考程序的输出包含以下内容: 迭代次数及残差、迭代计时、有效带宽估算、解的误差、以及 OpenMP 的最大线程数, 这些信息供同学们参考.
2. 提供的参考串行版本程序运行大约需要 30s. 同学们在刚开始优化时可以将 MAXITER 调小, 例如 10 , 以减少程序运行时间, 方便调试; 但是最终版需要保证残差能够达到收敛标准.
3. 提供的参考程序的所有部分均可根据需要修改, 但需保证最终解的残差达到收敛标准、与真解的误差符合要求. 正确程序的相对误差 (relative error) 应为 0.0528 左右.
4. 串行版本程序需要 34 步迭代收敛. 即使修改后的算法使用更少的迭代步数就达到收敛标准, 也必须记录进行 34 步迭代的时间作为计时标准.
5. 参考程序提供的编译选项中: -Ofast 提示编译器进行激进优化, 可能会改变浮点数求和顺序; -march=native 是提示编译器开启适合处理器的指令集 (例如 AVX2 向量指令集) 和一些其他配置. 同学们可以调整或尝试其他编译选项. 建议采用较新版本的 GCC 编译器, 也可以选择 intel 编译器或集群上提供的其他编译器.
6. 如果提交任务的节点上同时有其他程序正在运行, 可能会对性能产生影响.

参考文献

[1] Takeshi Iwashita, Hiroshi Nakashima, and Yasuhito Takahashi. Algebraic block multi-color ordering method for parallel multi-threaded sparse triangular solver in iccg method. In 2012 IEEE 26th International Parallel and Distributed Processing Symposium, pages 474–483. IEEE, 2012.

[2] Yiqun Liu, Chao Yang, Fangfang Liu, Xianyi Zhang, Yutong Lu, Yunfei Du, Canqun Yang, Min Xie, and Xiangke Liao. 623 tflop/s hpcg run on tianhe-2: Leveraging millions of hybrid cores. The International Journal of High Performance Computing Applications, 30(1):39–54, 2016.

[3] Jongsoo Park, Mikhail Smelyanskiy, Karthikeyan Vaidyanathan, Alexander Heinecke, Dhiraj D Kalamkar, Md Mosotofa Ali Patwary, Vadim Pirogov, Pradeep Dubey, Xing Liu, Carlos Rosales, et al. Optimizations in a high-performance conjugate gradient benchmark for ia-based multi-and many-core processors. The International Journal of High Performance Computing Applications, 30(1):11–27, 2016.

[4] Qianchao Zhu, Hao Luo, Chao Yang, Mingshuo Ding, Wanwang Yin, and Xinhui Yuan. Enabling and scaling the hpcg benchmark on the newest generation sunway supercomputer with 42 million heterogeneous cores. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pages 1–13, 2021.