南　　开　　大　　学

计　算　机　学　院

并行程序设计实验报告

期中开题报告

2010234 徐文斌

年级：2020 级

专业：计算机科学与技术

指导教师：王刚

2022 年 3 月 30 日

摘要

线性方程组的求解问题是线性代数研究的一个重点方向，高斯消去算法正是一个有效地求解该问题的算法。因此，本学期的并行课程实验，我选择对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法的并行化进行研究。本报告介绍了实验的研究内容，分析了前人所做出的相关工作。并对本学习的实验研究做了阶段性的学习计划。

关键字：并行、普通高斯消去、Gröbner 基计算的高斯消去、研究计划

目录

一、 问题描述

(一) 普通高斯消去

高斯消元法（Gaussian Elimination）是数学上线性代数中的一个算法。高斯消元法的原理是：若用初等行变换将增广矩阵 [A,B] 化为 [C,D]，则 AX=B 与 CX=D 是同解方程组。所以我们可以用初等行变换把增广矩阵转换为行阶梯阵，然后回代求出方程的解。如图1所示，将线性方程组表示为矩阵形式，通过把矩阵转化为行阶梯形矩阵，并将结果回代即可得到方程组的解。高斯消元法可用来为线性方程组求解，求出矩阵的秩，以及求出可逆方阵的逆矩阵。

$$\begin{cases} a_{11}x_1+a_{12}x_2+\cdots+a_{1n}x_n=b_1 \\ a_{21}x_1+a_{22}x_2+\cdots+a_{2n}x_n=b_2 \\ \cdots\cdots \\ a_{n1}x_1+a_{n2}x_2+\cdots+a_{nn}x_n=b_n \end{cases}$$

$$\Big\Downarrow \text{矩阵表示}$$

$$AX=B \quad (|A|\neq 0)$$

$$A=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, X=\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, B=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

图 1

高斯消元法的算法复杂度是 O(n^3)，如果系数矩阵的规模是 n × n，那么高斯消元法所需要的计算量大约与 n^3 成比例。高斯消元法可以用在电脑中来解决数千条等式及未知数。不过，如果有过百万条等式时，该算法会十分费时。随着多核处理器的日益普及，程序员们可以利用并行高斯消元算法来提高计算的速度。

(二) 特殊高斯消去——Gröbner 基的计算

特殊高斯消去计算来自一个实际的密码学问题——Gröbner 基的计算，与普通高斯消去计算的区别为：

- 1 运算均为有限域 GF(2) 上的运算，即矩阵元素的值只可能是 0 或 1。其加法运算实际上为异或运算：0+0=0、0+1=1、1+0=1、1+1=0。由于异或运算的逆运算为自身，因此减法也是异或运算。乘法运算 0*0=0、0*1=0、1*0=0、1*1=0。因此，高斯消去过程中实际上只有异或运算——从一行中消去另一行的运算退化为减法。

- 2 矩阵行分为两类，“消元子”和“被消元行”，在输入时即给定。消元子是在消去过程中充当“减数”的行，不会充当“被减数”。所有消元子的首个非零元素（即首个 1，称为首项）的位置（可通过将消元子放置在特定行来令该元素位于矩阵对角线上）都不同，但不会涵盖所有对角线元素。被消元行在消去过程中充当“被减数”，但有可能恰好包含消元子中缺失的对角线 1 元素，此时它“升格”为消元子，补上此缺失的对角线 1 元素。

二、 相关研究工作

（一） 高斯–若尔当消元法

高斯–若尔当消元法（Gauss-Jordan Elimination），简称 G-J 消元法，是数学中的一个算法，是高斯消元法的另一个版本，其方法与高斯消去法相同。唯一相异之处就是该算法产生出来的矩阵是一个简化行梯阵式，而不是高斯消元法中的行梯阵式。相比起高斯消元法，该算法的效率较低，却可把方程组的解用矩阵一次过表示出来。相对于高斯消元法，Gauss-Jordan 消元法最后的得到线性方程组更容易求解，它得到的是简化行列式。其转化后的增广矩阵形式如图2所示，因此它可以直接求出方程的解，而无需使用替换算法。

$$\left(\begin{array}{cccc|c} 1 & 0 & \cdots & 0 & s_1 \\ 0 & 1 & \cdots & 0 & s_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & s_n \end{array}\right)$$

图 2

（二） 部分选主元高斯消去算法

普通高斯消去算法中，未知量是按其出现于方程组中的自然顺序消去的，所以又叫顺序消去法。实际上已经发现顺序消去法有很大的缺点。设用作除数的 $a_{kk}^{(k-1)}$ 为主元素，首先，消元过程中可能出现 $a_{kk}^{(k-1)}$ 为零的情况，此时消元过程无法进行下去；其次如果主元素 $a_{kk}^{(k-1)}$ 很小，由于舍入误差和有效位数消失等因素，其本身常常有较大的相对误差，用其作除数，会导致其它元素数量级的严重增长和舍入误差的扩散，使得所求的解误差过大，以致失真。

误差分析的理论和计算实践均表明：顺序消元法在系数矩阵 A 为对称正定时，可以保证此过程对舍入误差的数值稳定性，对一般的矩阵则必须引入选取主元素的技巧，方能得到满意的结果。高斯消去法的一个简单变形——部分选主元高斯消去法，可以产生可靠的结果。在部分选主元的高斯消去法的第 i 步，我们在第 i 行到第 n-1 行中寻找第 i 列元素的绝对值最大的行并将这一行与第 i 行交换（变成主元）。在部分主元消去法中，未知数仍然是顺序地消去的，只是选各方程中要消去的那个未知数的系数的绝对值最大的作为主元素，然后用顺序消去法的公式求解。[1]

三、 研究计划

（一） 阶段划分

实验将以 arm 平台为主，根据不同的优化方案对程序进行编写和性能测试，并结合相应的性能评测工具如 perf 等对程序性能差异进行分析，找出较好的优化方案。

实验总体将分为五个子任务，分别是 SIMD 向量优化、Pthread 多线程实验、OpenMP 多线程实验、MPI 分布式内存实验及 CUDA GPU 实验进行。

（二） 具体安排

- 阶段 1

  利用 SIMD 超标量优化方法对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法进行优化并编写程序进行性能测试和分析。

- 阶段 2

  利用 Pthread 框架对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法进行多线程并行优化并编写程序进行性能测试和分析。

- 阶段 3

  利用 OpenMp 框架对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法进行多线程并行优化并编写程序进行性能测试和分析。

- 阶段 4

  利用 MPI 框架对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法进行分布式内存架构并行优化并编写程序进行性能测试和分析。

- 阶段 5

  利用 CUDA 平台对普通高斯消去算法及 Gröbner 基计算的特殊高斯消去算法进行 GPU 优化并编写程序进行性能测试和分析。

完成各次子实验后，对普通高斯消去及 Gröbner 基计算的特殊高斯消去算法的各个优化方案进行总结和比较，从而得到适合该算法的并行优化方案。

参考文献

[1] 田希山. 用部分选主元的高斯消去法并行求解线性方程组. [j]. 电脑知识与技术,2011,7(16):3960-3963+3966.