



南開大學
Nankai University

计算机学院
并行程序设计

Gröbner 基计算中的高斯消元并行化改进

小组成员姓名：丁屹、卢麒萱

小组成员学号：2013280、2010519

专业：计算机科学与技术

2022 年 3 月 30 日

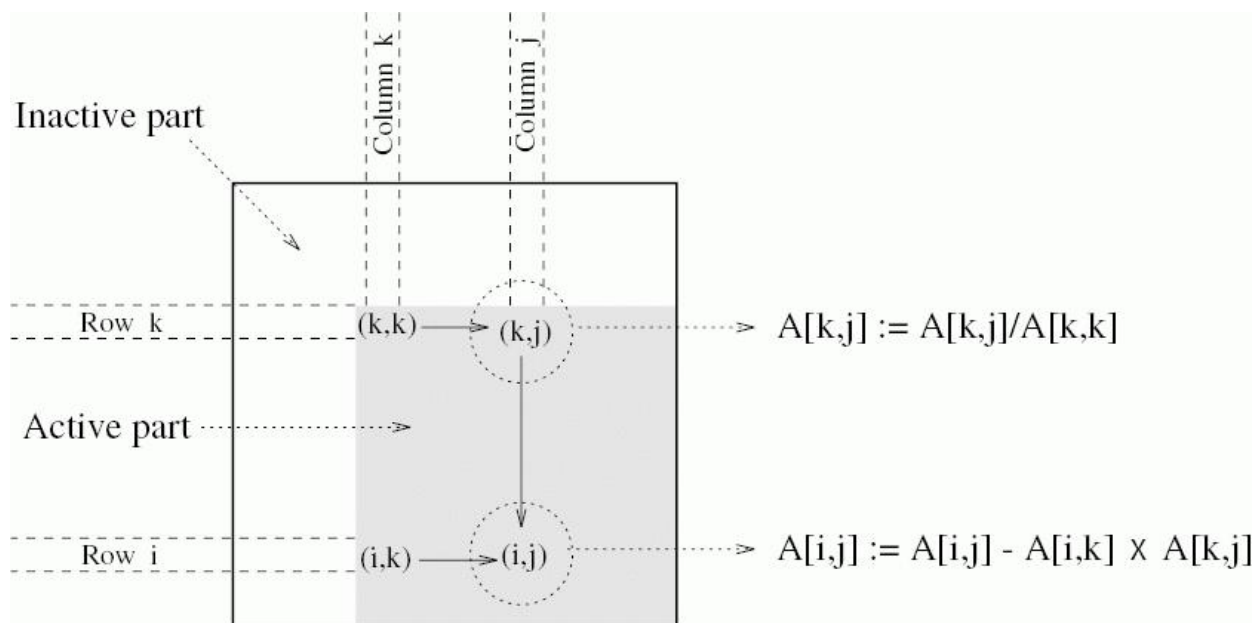
目录

1 研究问题	2
2 背景知识	2
2.1 普通高斯消去算法	2
2.2 数据结构	3
2.3 数据访问	3
2.4 并行化方法	3
3 研究计划	3
4 参考文献	4

1 研究问题

2 背景知识

2.1 普通高斯消去算法



普通高斯消去算法

```

1  procedure LU (A)
2  begin
3      for k := 1 to n do
4          for j := k + 1 to n do
5              A[k, j] := A[k, j] / A[k, k];
6          endfor;
7          A[k, k] := 1.0;
8          for i := k + 1 to n do
9              for j := k + 1 to n do
10                 A[i, j] := A[i, j] - A[i, k] * A[k, j];
11             endfor;
12             A[i, k] := 0;
13         endfor;
14     endfor;
15 end LU

```

观察高斯消去算法，注意到伪代码第 4, 5 行第一个内嵌循环中的 $A[k, j] := A[k, j] / A[k, k]$ 以及伪代码第 8, 9, 10 行双层 for 循环中的 $A[i, j] := A[i, j] - A[i, k] * A[k, j]$ 都是可以进行向量化的循环。可以通过 SIMD 扩展指令对这两步进行并行优化。

2.2 数据结构

可采用位向量方式存储每个消元子和被消元行，优点是消元操作变为位向量异或操作，算法实现简单，且适合并行化，易达到更高的并行效率，缺点是 Gröbner 基计算中产生的消元子和被消元行非常稀疏，非零元素（1 元素）在 5 % 以下，位向量存储和计算可能并非最优。也可采用类似倒排链表的存储及方式——可认为是稀疏 0 / 1 矩阵的紧凑存储方式，每个消元子和被消元行只保存 1 元素的位置，且按升序排列，从而类似倒排链表数据结构。优点是存储空间占用更少，缺点是算法设计更复杂，并行化难度高。

2.3 数据访问

矩阵规模可能非常庞大，达到数百万行 / 列，难以全部放入内存。此时，需要设计的是外存算法，考虑如何分批次将数据读入内存进行处理，同时又保证正确性。对外存算法，算法分析和时间测试除了考虑计算之外，还要考虑 I / O 时间。

2.4 并行化方法

1. 对位向量存储方式，两行间消元操作的并行化很直接，无论 SIMD、多线程还是 MPI、GPU，将向量拆分，子向量的异或即自然形成任务，可分配给不同的计算单元。当矩阵规模大到百万级别时，采取这种并行方式就够了。但当矩阵规模没有那么大时，一个消元操作计算量不足以支撑较大规模并行，就需要考虑消元操作间的并行，设计适合的并行任务划分，在提高并发度的同时保证正确性。
2. 对类倒排链表存储方式，可以考虑循环展开和多线程优化。
3. 当矩阵规模非常庞大，需使用外存算法时就要同时考虑计算和访存。多线程并行化时可考虑计算和访存异步模式，在前台线程进行消元计算时、后台线程读取下一步要处理的数据，这需要仔细设计计算和 I / O 步骤，降低依赖关系，以便实现异步模式。

3 研究计划

4 参考文献

[1][2]

参考文献

- [1] 牟晨琪. 计算机代数. <http://cmou.net/files/PolyAlg2-2020.pdf>, 2020.
- [2] 狄鹏. Grobner 基生成算法的并行. <https://www.academia.edu/18478407/Grobner%E5%9F%BA%E7%94%9F%E6%88%90%E7%AE%97%E6%B3%95%E7%9A%84%E5%B9%B6%E8%A1%8C>, 2008.