消元子模式的高斯消元算法说明

杨浩然

2022/3/28

1 算法简介

本算法源自布尔 Gröbner 基计算。在 HFE80 的 Gröbner 基计算过程中，高斯消元时间占比可以达到 90% 以上。考虑到此算法中高斯消元的特殊性，为加快高斯消元速度，设计此算法。

本算法将矩阵各行分为两类：消元子、被消元行，后续伪代码将具体描述消元过程。

2 符号说明

R：所有消元子构成的集合

$R[i]$：首项为 i 的消元子

E：所有被消元行构成的数组

$E[i]$：第 i 个被消元行

$lp(E[i])$：被消元行第 i 行的首项

解释一下这里首项的含义：首项是指某行下标最大的非零项的下标，如某行为 011000，从左到右下标分别为 5,4,3,2,1,0，那么首项为 4，因为该行非零项下标为 3,4，其中最大值为 4。

3 算法伪代码

这里只给出串行算法伪代码，其中被消元行有 m 行，消元子最大首项为 t。

1. **for** $i := 0$ **to** $m - 1$ **do**
2. **while** $E[i]! = 0$ **do**
3. **if** $R[lp(E[i])]! = NULL$ **then**
4. $E[i] := E[i] - R[lp(E[i])]$
5. **else**
6. $R[lp(E[i])] := E[i]$
7. **break**
8. **end if**
9. **end while**
10. **end for**
11. **return** E

其中外层循环表示遍历每个被消元行。内层循环表示针对每个被消元行，如果该行未被消为 0，那么根据其首项选择消元子进行消元；当存在合适的消元子，则用该消元子进行消元；否则将该被消元行作为消元子，参与后续高斯消元过程。

4 提示

1. 为节约内存，可以用位图表示矩阵的一行，如果使用这种方法表示，那么算法第 4 行的减法实质为异或运算。

2. 之前给的示例为一种可行的具体实现方案，该方案优势在于消元子较多时可以通过控制消元子规模节约内存。就本实验而言，大家可以从本文算法出发，自行设计合适的并行算法。

3. 非零消元子和被消元行的数量之和可能大于矩阵列数，但这时必将有一部分被消元行被消为 0。

4. 测试样例中的“消元结果.txt”里面每一行表示一个被消元行，如果某行为空，那么表示对应被消元行会被消为全 0，读取文件时注意不要读串行。