应用数学和力学，第17卷 第12期（1996年12月）
Applied Mathematics and Mechanics

应用数学和力学编委会编
重庆出版社出版

求解线性方程组的顺序算法和并行算法的理论时耗估价

萨尔曼·哈·阿巴什[1]

（钱伟长推荐，1995年5月24日收到）

摘　要

本文讨论了求解密集型线性方程组的两种并行算法。这两种算法都在下上单元(LU)分解法的基础上使用了前向和后向置换进行的。这些算法在数值上是稳定的，并在顺序平衡机上用各种处理程序进行试验，都得到良好效果。

关键词　下上单元(LU)分解法　前向和后向置换　多指令多数据程序（MIMD程序）　多任务　理论时耗估价

一、引　　论

线性代数的中心问题之一是线性联立方程组的求解。在未知数等于方程式数时的情况尤为重要。这里将研究线性方程组

$$A\boldsymbol{x}=\boldsymbol{b} \tag{1.1}$$

的求解问题，其中A为$n\times n$矩阵，而$\boldsymbol{x}$和$\boldsymbol{b}$都是$n\times 1$列矢量。

在一个顺序计算机上，高斯消去算法要求乘除$n^3/3+O(n^2)$次。Sameh 和 Kuck[1]曾证明，在多指令多数据并行机上，在$p=(n-1)^2$种处理程序中，最有效的程序只需乘除$3(n-1)$次。Lord及其同人[2]曾在HEP（高能物理）计算机($p=8$)[3]上对许多算法进行了测试，他们认为$p=(n-1)^2$是不适当的，并在$p=[n/2]$的假设下，对高斯消去法重新进行了考核。他们证明，算法需要乘除n^2-1次。它们的加速比是$(n^3/3)/(n^2-1)\approx n/3$，而他们的效率是$(n/3)/(n/2)=2/3$。

Davies[4]的早期工作采用了列 LU 因子分解法。人们对以列因子为基础的方案持反对态度，原因是在后向（反向）置换时，人们很难采用并行计算。它只在处理程序数量很大时，才是有意义的。

本文的目的是在求解线性方程组(1.1)时给出新的顺序和并行计算法，我们使用了：

(a) 顺序和并行LU-分解法；

(b) 顺序和并行分块LU-分解法；

接着还使用了前向和后向置换，并对这些算法进行了理论时耗估价。

---

1 巴林大学数学系，巴林邮政信箱 32038

本文投稿为英文，由韦凌德译　吴承平校

在理论上，可以证明并行LU-分解的时耗近似地为

$$(3n+1)t_r+n(3n^2+6n+7)t_s/2+(n^3+4n^2)t_f/2p$$

其中$p \ll n$，t_r为会合时间，t_s为一个浮点数的分布时间，t_f为一次运算所需时间，n为方程组的大小，p为处理程序数。当$p=n/2$时，算术运算的次数是$O(n^2)$，于是，加速比为$(n^3/3)/n^2=n/3$，而效率$=(n/3)/(n/2)=2/3$，它和 Lord 及其同人[4]找到的是相同的。在共享存储机中，$p \ll n$，而加速比$=(n^3/3)/(n^3/2p)=2p/3$。

在平行分块 LU-分解法中，所需算术运算次数，业已证明为$O(w^3q^3/p)$。我们可以证明，在顺序分块 LU-分解法中，所需算术运算次数应该是$O(w^5q^3)$。所以，加速比$=w^5q^3/(w^3q^3/p)=w^2p$，它和w的大小有关。

我们必须指出，本文提出的 LU-分解算法和分块 LU-分解算法也都适用于多道处理程序，而且，我们也证明了，对于求解线性方程组而言，分块 LU-分解算法，比 LU-分解算法还要快。

二、顺序和并行LU-分解算法

线性方程组$A\boldsymbol{x}=\boldsymbol{b}$可以分三步求解：

(a) 分解$A=LU$成下三角矩阵L和上三角矩阵U的乘积；

(b) 解$L\boldsymbol{y}=\boldsymbol{b}$（前向置换）；

(c) 解$U\boldsymbol{x}=\boldsymbol{y}$（后向置换）。

我们将在下文描述所推荐的顺序计算：

下述算法是把矩阵A的系数分解为L和U的系数，其中L的系数l_{ij}为输入项，当$i=2, 3, \cdots, n$；$j=1, 2, \cdots, i-1$时，$l_{ij}=a_{ij}$；当$i=1, 2, 3, \cdots, n$时，$l_{ii}=1$；设U的系数为u_{ij}，当$i=1, 2, \cdots, n$；$j=i, i+1, \cdots, n$时，$u_{ij}=a_{ij}$。

```
For i in 1, …, n-1 loop
  For j in i+1, …, n loop
  a_ji := a_ji/a_ii;
    For k in i+1, …, n loop
    a_jk := a_jk - a_ji × a_ik;
    End loop;
  End loop;
End loop;
```

这一算法需算$(n-1)$步，约要付出$n^3/3$乘除次数的时耗代价。前向和后向的置换算法应该如下：

```
For i in 1, …, n loop
  z_i := y_i;
End loop;
  For j in 1, …, n-1 loop
    For i in j+1, …, n loop
    z_i := z_i - a_ij × z_j;
    End loop;
```

```
End loop;
For i in reverse 1, …, n loop
x_i := z_i/a_ii;
End loop;
For j in reverse 2, …, n loop
  For i in reverse 1, …, j-1 loop
  x_i := x_i - (a_ii/a_ij) × x_j;
  End loop;
End loop;
```

上述程序在一开始就把y写成z，这样就求得z_1.在计算z_i，$i=2,\cdots,n$时，z_1和$y_i(=z_i)$，$i=2$，…，n是必需取用的．最后，用$x_n=z_n/a_{nn}$计算求得x_n．前向和后向置换所付出的时耗代价是$O(n^2)$．

在实际操作中，我们需有固定数量的处理程序p，它们和矩阵大小n是无关的．而且，并行性的专用编组是和方程组有关的．我们可以把矩阵A分成若干行，每行有不同的任务，可以将所需计算并行进行．我们曾用 Ada 语言（美国国防部的标准高级语言）实施这种算法．Ada语言提供了描述 MIMD 并行算法的任务条件，而且并不要求同时满足任务数和程序数完全相等这一限制．

并行算法如下：

用多任务并行地处理单行的办法重写上述算法如下：

```
The task required to triangulate the martix A is
Accept pivot row(i), elimination row(j), i;
compute the multipliers
For j>i, k>i do
a_jk := a_jk - a_ji × a_ik;
return row j;
```

主程序为

```
  For i in 1, …, n-1 loop
    for j in i+1, …, n loop
    send row(i), row(j), i to forward elimination task;
    End loop;
    collect modified row(j);
    End loop;
  End loop;
```

这里涉及的并行性是很直接前进的，但我们必须指出，任务数完全由n决定，任务数并不等于程序数，而且也无法控制编组工作．

前向置换的任务，给出如下：

```
Accept the vector y;
Set z_i=y_i, i=1, 2, …, n;
Return z;
Accept a_ij, z_j
```

```
  z_i:=z_i-a_ij×z_j;
  Return z_i;
```

后向置换的任务，给出如下：

```
  Accept a_ii,  z_i;
  Compute x_i=z_i/a_ii,      i=1, 2, …, n
  Return x_i;
  Accept a_ij,  x_j;
  Compute x_i:=x_i-(a_ij/a_ii)×x_j;
  Return x_i;
```

上述任务显示了矢量z的并行计算．初始，有下述各步进行对$i=1,\ \cdots,\ n$的并行计算；先是$z_i=y_i$，然后是$z_i=z_i-a_{ij}\times z_j$对$j=i+1,\ \cdots,\ n$进行并行计算．用后向置换求得x_n值，再把x_n置换入其它任务中，从而修正了右侧的数值．

并行LU-分解法的理论时耗估价是由算术运算时耗估价t_f，会合时耗估价t_r，数据分布时耗估价组合而成．业已在顺序平衡计算机上数度[2]测量了这些时耗估价值．任务数在1到8之间，有关程序数从2到9之间；其结果为$t_f=0.25$毫秒，$t_r=2$毫秒，$t_s=0.02$毫秒．LU-分解法的并行算法总时耗为

$$n(3n+1)\times t_r+n(3n^2+6n+7)\times t_s/2+(n^3+4n^2)\times t_f/2p \tag{2.1}$$

这里的程序数$p\ll n$．

对一台顺序计算机（并设$p=5$）而言，并行算法的理论时耗估价，当$n=100$时，约为116.8秒，而对相当于并行算法的最好串行运算而言，估价约为220.8秒．于是，加速比约为1.89，这和实际加速比相当接近．

三、顺序和并行分块LU-分解算法

让我们考虑一种把矩阵分块进行的算法．考虑方程组(1.1)，把A分成q^2块，每块的阶为$w\times w$，也可以把矢量$\boldsymbol{x}$及$\boldsymbol{b}$分为q块，每块的阶为$w\times 1$，其中q是块行的总数．由于矩阵A是由块行$(1,\ \cdots,\ q)$组成的诸多块组成的，而$\boldsymbol{x}$和$\boldsymbol{b}$是由块矢$(1,\ \cdots,\ q)$所组成的诸多块矢组成的．

对于没有主项的分块LU-分解法（用三角形法的分块矩阵）的程序算法为

```
 For i in 1, …, q-1 loop
     transpose (A_ii,  pivot);
     triangulate (pivot);
   For j in i+1, …, n loop
     transpose (A_ji,  elim);
     solve (pivot,  elim,  mult);
     A_ji=mult;
       For k in k=i+1, …, q loop
         A_jk:=A_jk-mult×A_ik;
       End loop;
   End loop;
```

End loop;

上述顺序算法的理论时耗估价约为

$$\sum_{i=1}^{q-1} w^3\left[\sum_{j=i+1}^{q}\left(w^2+\sum_{j=i+1}^{q} w^2\right)\right]\Big/3$$

它等于

$$\begin{aligned}&\sum_{i=1}^{q} w^3(w^2q+(wq)^2-2w^2qi-iw^2+w^2i^2)/3\\&=w^5[q(q-1)+q^2(q-1)-2q^2(q-1)/2-q(q-1)/2\\&\quad+q(q-1)(2q-1)/6]/3\\&=w^5(q^3-q)=O(w^5q^3)\end{aligned} \tag{3.1}$$

求解$Ly=\boldsymbol{b}$的前向置换算法如下文：

为了求解$Ly=\boldsymbol{b}$，我们可以写成

```
y=b1
For i in 2, …, q loop
sum:=bi
For j in 1, …, i-1 loop
sum:=sum-mij×yj
End loop;
yj:=sum;
End loop;
```

前向置换法的理论时耗估价为

$$\begin{aligned}\sum_{i=2}^{q}\sum_{i=1}^{i-1} w^2&=w^2[(q^2+q-2)/2-(q-1)]\\&=w^2(q^2/2-q/2-2)=O(w^2q^2)\end{aligned} \tag{3.2}$$

下面是求解$Ux=y$的后向置换算法：

```
Triangulate Aqq;
Solve for Lqq and Uqq;
For i in reverse 1, …, q-1 loop
sum:=0.0;
For j in reverse i+1, …, q loop
sum:=sum+Aij×xj;
End loop;
vi:=yi-sum;
Solve Lii×zi:=vi;
      Uii×xi:=zi;
```

后向置换法的时耗估价是

$$\sum_{i=2}^{q-1}\left\{\left[\sum_{j=i+1}^{q} w^2\right]+\frac{w^3}{3}\right\}=\frac{w^3(q-1)(3q+2)}{6}=O(w^2q^2) \tag{3.3}$$

串行算法的总时耗估价为

$$[w^5(q^3-q)+w^2(q^2/2-q/2-2)+w^3(q-1)(3q+2)/6] \tag{3.4}$$

并行分块LU-分解法是在[5]用多任务法给出的，其理论时耗估价为：

(a) 用来分解分块矩阵所需时耗为

$$2q(q-1)(q+1)\times t_r/3+q(q-1)(17q-4)w^2t_s/6+q(q-1)(3q+7)w^3t_f/3p$$

(b) 前向置换法所需时耗为

$$q(2q-1)\times t_r+q[(3q-1)+(q-1)w]w\times t_s+q(q-1)w^2\times t_f/p$$

(c) 后向置换法所需时耗为

$$[q(q-1)+2]\times t_r+q[q(q+3)w/2+3q(q-1)w^2/2]\times t_s$$
$$+q[qw^3/3+w^2(3q-1)]t_f/p$$

于是，我们所关心的算术运算时耗约为

$$(w^3q^3+w^2q^2+w^2q)t_f/p \tag{3.5}$$

从(3.1)式，近似时耗为

$$[w^5q^3+w^2q^2/2+w^2q^2/2]t_f \tag{3.6}$$

所以，当$n=100$，$q=20$，和$w=5$时，用 5 份程序的并行算法所需理论时耗（估价）(3.5)约为50.5秒，而串行算法所需时耗(3.6)是788.8秒。

四、结　论

本文所述的 LU-分解算法和分块 LU-分解算法同样适用于多重处理机系统。它们由于算术运算量很大，所以，也能达到很高效率。分块 LU-分解法一般比 LU-分解法快，所以在求解线性方程组时，应选用这一算法。

参 考 文 献

[1] A. Smeh and D. J. Kuck, On stable linear system solver, *J. Assoc. Comput. Mach.*, (25) (1978), 81—89.

[2] R. E. Lord, J. S. Kowalik and S. P. Kumar, Solving linear algebric equations on MIMD computer, *J. Assoc. Comput. Mach.*, 30(1) (1983), 103—117.

[3] M.J. Quinn, *Designing Efficient Algorithm for Parallel Computers*, McGraw-Hill International Editions, Computer Science Series (1988).

[4] G. J. Davies, Column LU Factorization with Partial Pivoting on a Hypercube Multiprocessors, Technical Report ORNL-6219, Mathematical Science, Oak Ridge, TN 37831 (1985).

[5] S. H. Abbas, Parallel algorithms of linear systems and initial value problems, Ph. D. Thesis, University of Liverpool (1990).

The Theoretical Cost of Sequential and Parallel Algorithms for Solving Linear System of Equations

Salman H. Abbas

(Department of Mathematics, University of Bahrain, P. O. Box 32038, Bahrain)

Abstract

In this paper two parallel algorithms for solving dense linear equations are discussed. The algorithms are based on LU-decomposition followed by for-ward and back-ward substitutions. The algorithms are numerically stable and have been tested on the Sequent Balance Machine with efficient utilization of all processors.

Key words LU decomposition, for-ward and back-ward substitutions, MIMD machine, multi-tasking, theoretical cost