一种基于 Doolittle LU 分解的线性方程组并行求解方法

徐晓飞 曹祥玉 姚 旭 陈 盼
(空军工程大学电讯工程学院 西安 710077)

摘 要：矩阵方程的快速求解是矩量法计算电大问题的关键，LU 分解是求解线性方程组的有效方法。该文详细地分析了 Doolittle LU 分解过程，基于分解过程的特点，在 MPI(Message-Passing interface) 并行环境下，提出了按直角式循环对进程进行任务分配的并行求解方法。实验证明该方法可以有效地减少进程间数据通信量，从而加快计算速度。

关键词：Doolittle LU 分解；线性方程组；并行计算
中图分类号：TP311 **文献标识码**： A **文章编号**：1009-5896(2010)08-2019-04
DOI: 10.3724/SP.J.1146.2009.01401

Parallel Solving Method of Linear Equations Based on Doolittle LU Decomposition

Xu Xiao-fei Cao Xiang-yu Yao Xu Chen Pan
(*Telecommunication Engineering Institute*, *Air Force Engineering University*, *Xi'an* 710077, *China*)

Abstract: The fast matrix solving is the key of the moment method when computing the electrically large issues. LU decomposition is a efficient algorithm for solving linear equations. In this paper, Doolittle LU Decomposition is described detailedly. Based on the decomposition characteristics, a parallel solving method looping over squares is proposed in MPI (Message-Passing interface) parallel environment. The experiments indicate that the method can decrease communication quantity between processes and accelerate computing speed efficiently.

Key words: Doolittle LU decomposition; Linear equations; Parallel computation

1 引言

矩量法[1]作为求解电磁场数值问题的主要方法，被广泛应用于分析天线问题和各种复杂目标体的电磁散射问题。然而随着计算目标尺寸的增大，计算未知量随之增多，阻抗矩阵的填充与求解会消耗大量的计算资源，这成为矩量法求解的瓶颈。所以如何加快矩阵填充，尤其是矩阵方程的快速求解成为了研究的重点。在求解线性方程组的众多方法中，LU分解求解线性方程组算法是较为常用的一种求解方法。Golub等在1996年提出了分块LU方法[2]，Nico等人提出利用GPU加速计算分块LU的方法[3]，文献[4-9]介绍了线性方程组的几种并行求解方法。本文在MPI(Message-Passing Interface)[10]并行环境下，基于Doolittle LU分解方法[11]的原理，提出了一种线性方程组的并行求解方法。

2009-10-29 收到，2010-01-14 改回
国家自然科学基金(60671001)和空军工程大学电讯工程学院博士创新基金(200706)资助课题
通信作者：徐晓飞 x.f.xu@live.cn

2 Doolittle LU分解方法原理

对于线性方程组 $\boldsymbol{AX}=\boldsymbol{B}$，假设 $\boldsymbol{A}$ 的顺序主子式都不为零。则 $\boldsymbol{A}$ 可作 $\boldsymbol{LU}$ 分解，即 $\boldsymbol{A}=\boldsymbol{LU}$，其中 $\boldsymbol{L}$ 是单位下三角阵，$\boldsymbol{U}$ 是上三角阵

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix} \tag{1}$$

比较式(1)两侧，可以得到

$$a_{ij}=\sum_{k=1}^{\min(i,j)} l_{ik}u_{kj} \tag{2}$$

先算出 $\boldsymbol{U}$ 的第1行，在式(2)中令 $i=1$，得到

$$u_{1j}=a_{1j},\quad j=1,2,\cdots,n \tag{3}$$

接着在式(2)中令 $j=1$，从而算出的 $\boldsymbol{L}$ 第1列

$$l_{i1}=a_{i1}/a_{11},\quad i=2,\cdots,n \tag{4}$$

若已算出 $\boldsymbol{U}$ 的1至 $r-1$ 行元素，$\boldsymbol{L}$ 的1至 $r-1$ 列元素，可用式(5)先算出 $\boldsymbol{U}$ 的第 r 行元素

$$u_{rj} = a_{rj} - \sum_{k=1}^{r-1} l_{rk} u_{kj}, \quad j = r, r+1, \cdots, n \tag{5}$$

接着计算 $\boldsymbol{L}$ 的第 r 列

$$l_{ir} = \left(a_{ir} - \sum_{k=1}^{r-1} l_{ik} u_{kr} \right) \Big/ u_{rr}, \quad i = r+1, \cdots, n \tag{6}$$

上述矩阵 $\boldsymbol{A}$ 的 $\boldsymbol{LU}$ 分解计算步骤可按 $\boldsymbol{U}_1$ 行 $\boldsymbol{L}_1$ 列，$\boldsymbol{U}_2$ 行 $\boldsymbol{L}_2$ 列，…，依次进行，如图1所示

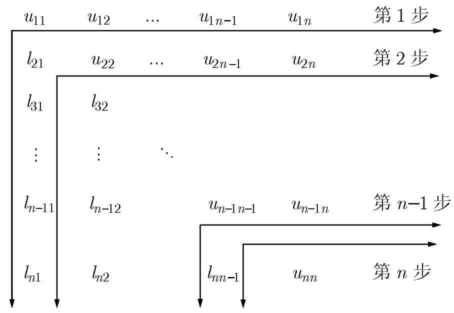


图1 Doolittle方法LU分解过程

对于线性方程组 $\boldsymbol{AX}=\boldsymbol{B}$，利用Doolittle方法求出 $\boldsymbol{L}$，$\boldsymbol{U}$ 后，令 $\boldsymbol{UX}=\boldsymbol{Y}$ 就得到一个与 $\boldsymbol{AX}=\boldsymbol{B}$ 等价的方程组

$$\left.\begin{aligned} \boldsymbol{LY} &= \boldsymbol{B} \\ \boldsymbol{UX} &= \boldsymbol{Y} \end{aligned}\right\} \tag{7}$$

先通过向前回代求解 $\boldsymbol{LY}=\boldsymbol{B}$，

$$\left.\begin{aligned} & y_1 = b_1 \\ & y_i = b_i - \sum_{j=1}^{i-1} l_{ij} y_j, \quad i = 2, \cdots, n \end{aligned}\right\} \tag{8}$$

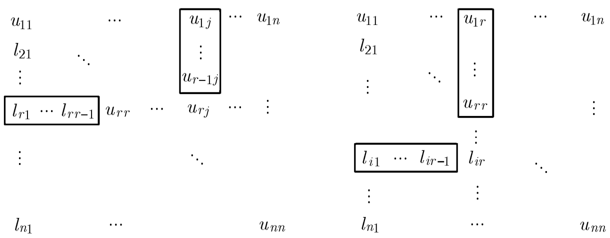
然后再通过向后回代求解 $\boldsymbol{UX}=\boldsymbol{Y}$，

$$\left.\begin{aligned} & x_n = y_n / u_{nn} \\ & x_i = \left(y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \quad i = n-1, \cdots, 2, 1 \end{aligned}\right\} \tag{9}$$

3 并行求解程序设计

首先观察式(5)，式(6)，计算 u_{rj} 所需要的元素为 a_{rj}，l_{rk}，u_{kj}，其中($k=1,2,\cdots,r-1$，$j=r,r+1,\cdots,n$)，计算 l_{ir} 所需要的元素为 a_{ir}，l_{ik}，u_{kr}，u_{rr}，其中($k=1,2,\cdots,r-1$，$i=r+1,\cdots,n$)，具体如图2所示。

通过观察，计算 u_{rj} 需要第 j 列上方的所有元素和第 r 行前 $r-1$ 个元素，计算 l_{ir} 需要第 i 行左侧所有元素和第 r 列前 r 个元素。所以，如果按照通常行循环[12]对进程进行任务分配，那么在计算 u_{rj} 时，就需要通过进程间通信获得第 j 列 u_{rj} 上方的所有元素。如果按照列循环[12]方式对进程进行任务分配，



(a) 计算 u_{rj} 所需要的元素　(b) 计算 l_{ir} 所需要的元素

图2 Doolittle LU 分解

那么在计算 l_{ir} 时就需要通过进程间通信获得第 i 行 l_{ir} 左侧所有元素。所以以上两种方式都不适合。本文基于Doolittle分解过程的特点提出了如图3所示的直角式循环方式对进程进行任务分配。

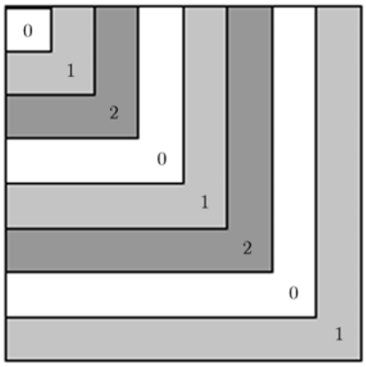


图3 按直角式循环分配

图3所示为3个进程按直角式循环分配，这种分配方式是以对角元素为角点，对角元素左侧和上方的元素分入一个进程，这样循环下去，直到将整个矩阵分配完毕。这样分配是根据上述分解方法的特点提出的，好处是在计算到第 r 步时，既计算 $u_{rr},\cdots,u_{rn}$，$l_{r+1,r},\cdots,l_{nr}$ 时，只需计算 u_{rr} 完毕后，将 u_{rr} 左侧与上方元素广播给各个进程，其它元素即可进行计算，而不再需要进程间的通信。这样，在计算完所有的元素后，总共需要广播 n 次，从而大大减少了数据通信量，从而加快了计算速度。

采用这种分配方式，关键是对矩阵的某个元素在进程内的局部编号，下面以一个 8×8 的矩阵为例(图4)，分析矩阵元素的编号方法。在每个进程内矩阵元素存储在一维数组 Zmn 中，在图4中，对于某个元素 a_{ij}^{k}，i,j 为元素的行号和列号，k 为该元素在所属进程内的编号，其计算公式为

$$k = \mathrm{I\_NUM}(i) + i - j + \max(i,j) \tag{10}$$

例如元素 a_{45} ,其编号为 $k=3+4-5+5=7$，再例如元素 a_{74}，其编号为 $k=8+7-4+7=18$，

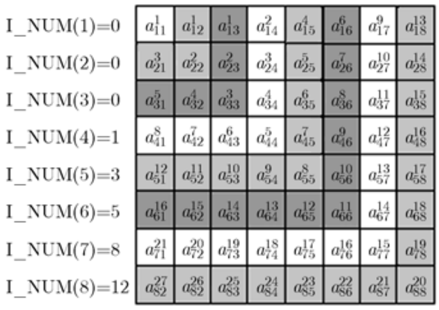


图4　进程内元素编号

这种编号首先需要生成数组 I_NUM 来记录进程内已经存储的元素个数，其生成方法为

```
DO I=1, PROCESS_NUM ！PROCESS_NUM为进程个数
  I_NUM(I)=0
ENDDO
DO I=PROCESS_NUM+1,N !N为矩阵维数
  I_NUM(I)=I_NUM(I-PROCESS_NUM)+2*(I-PROCESS_NUM)-1
ENDDO
```

通过上述过程，就完成了对矩阵元素在进程内的编号。下面就可以对方程组进行并行求解了，程序具体实现过程如表1。

表1　程序实现过程

```
DO I=1, N
  DO J=I,N！按列循环计算 u_ii,…,u_in
    IF ON THIS PROCESS！判断是否为当前进程
      CALCULATE U(I,J)
      IF (I==J) THEN
      CALL MPI_BCAST(U(1 … I,I) AND L(I,I … I-1))！广播 u_ii 左侧与上方元素
      ENDIF
    ENDIF
  ENDDO
  DO J=I+1,N！按行循环计算 l_{i+1i},…,l_ni
    IF ON THIS PROCESS！判断是否为当前进程
    CALCULATE L(J,I)
    ENDIF
  ENDDO
ENDDO
```

求解出 $\boldsymbol{L}$， $\boldsymbol{U}$ 后，就可以利用式(8)，式(9)回代即可解出 $\boldsymbol{X}$ 。

4　实验结果与分析

为了验证本文提出方法的有效性，笔者将并行处理前后的实验数据进行了对比。实验环境为Intel Core2 6600 双核(单核2.40 GHz), 2 G DDR Ⅱ 667内存。实验数据对比如表2所示。

表2　并行处理前后实验数据比较

系数矩阵大小	传统LU分解耗时(s)	本文并行方法耗时(s)
1000×1000	6	4
5000×5000	35	19

从表2可以看出，采用本文方法求解时间约为传统LU分解的1/2，说明方法的有效性。

5　结束语

快速求解线性方程组在众多科学工程问题中都极具意义，而并行计算是快速求解大规模问题的有效方法。本文设计的线性方程组并行求解方法，是基于Doolittle LU分解方法的特点而设计的，按直角式循环对进程进行任务分配，经过实验验证，该方法可以有效地减少数据通信量、加快计算速度，具有良好的并行性。

参 考 文 献

[1]　Harrington R F. Field Computation by Moment Methods[M]. Piscataway: IEEE Press, 1993: 62–79.

[2]　Gene H G and Charles F V L. Matrix Computations[M]. 3 rd ed. Baltimore: Johns Hopkins University Press, 1996: 81–109.

[3]　Nico G, Naga K G, Michael H, *et al*.. LU-GPU: Efficient algorithms for solving dense linear systems on graphics hardware[C]. Proceedings of the ACM/IEEE SC|05Conference, Washington, 2005: 12–18.

[4]　张维儒, 潘无名. 基于MPI的并行计算实现Jacobi迭代[J]. 软件导刊, 2008, 7(9): 16–17.
Zhang Wei-ru and Pan Wu-ming. Parallel jacobi iterative algorithm based on MPI. *Software Guide*, 2008, 7(9): 16–17.

[5]　韩明华, 彭宇行, 李思昆等. 基于Linux集群电磁散射并行计算实现[J]. 计算机研究与发展. 2005, 42(6): 1085–1088.
Han Ming-hua, Peng Yu-hang, and Li Si-kun, *et al*.. Implementation of parallel computation for electromagnetic scattering on linux cluster. *Journal of Computer Research and Development*, 2005, 42(6): 1085–1088.

[6]　付朝江. 基于工作站机群并行求解有限元线性方程组[J]. 计算机工程与设计, 2008, 29(24): 6441–6443.
Fu Chao-jiang. Parallel computing for solving finite element linear systems of equations on workstation cluster. *Computer Engineering and Design*, 2008, 29(24): 6441–6443.

[7]　胡晓力, 田有先. 解大规模线性方程组的Mann迭代并行算法[J]. 计算机应用与软件, 2008, 25(8): 62–64.

Hu Xiao-li and Tian You-xian. Mann iteration's parallel algorithm for solving large-scale linear systems of equations. *Computer Applications and Software*, 2008, 25(8): 62–64.

[8] 吴丹红. 一种加速大规模线性方程组求解的并行方法[J]. 机电工程, 2008, 25(4): 58–59.

Wu Dan-hong. Parallel accelerating method of solving large-scale linear equations[J]. *Mechanical & Electrical Engineering Magazine*, 2008, 25(4): 58–59.

[9] Wang X F and Ziavras S G. Parallel direct solution of linear equations on FPGA-based machines[C]. International Parallel and Distributed Processing Symposium, Nice, 2003: 1–8.

[10] 都志辉. 高性能计算并行编程技术—MPI并行程序设计[M]. 北京: 清华大学出版社, 2001: 12–36.

[11] 张池平, 施云惠. 计算方法[M]. 北京: 科学出版社, 2002: 36–38.

[12] 张玉. 电磁场并行计算[M]. 西安: 西安电子科技大学出版社, 2006: 47–52.

徐晓飞： 男，1982年生，博士生，研究方向为电磁场高效数值计算.

曹祥玉： 女，1964年生，教授，博士生导师，研究方向为电磁场数值计算、天线、电磁兼容等.

姚 旭： 男，1983年生，博士生，研究方向为波束赋形.

陈 盼： 男，1984年生，硕士生，研究方向为宽频带微带天线及阵列研究.