改进的并行 ORTHOMIN(*m*)算法

赵利斌,田有先
ZHAO Li-bin,TIAN You-xian

重庆邮电大学 计算机科学与技术学院,重庆 400065
Department of Computer Science and Technology,Chongqing University of Posts and Telecommunications,Chongqing 400065,China
E-mail:zlb628@126.com

ZHAO Li-bin,TIAN You-xian.Improved parallel ORTHOMIN(*m*) algorithm.Computer Engineering and Applications,2009,45(6):52-54.

Abstract: Employing an intrinsic property of the ORTHOMIN(m) algorithm and eliminating data interdependence for inner product computation in the ORTHOMIN(m) algorithm,an improved parallel ORTHOMIN(m) algorithm called IORTHOMIN(m) algorithm is established in this paper.The convergence of IORTHOMIN(m) algorithm is as same as ORTHOMIN(m) algorithm,but the times of the synchronization overhead are reduced by a factor of two when computed using the IORTHOMIN(m) algorithm on distributed memory cluster systems based on MPI environment.The numerical result and theoretical analysis prove that the performance of the IORTHOMIN(m) algorithm is better than that of the ORTHOMIN(m) algorithm.
Key words: ORTHOMIN(m) algorithm;parallel computation;synchronization overhead;non-symmetric sparse linear systems

摘　要: 通过利用 ORTHOMIN(m)算法的固有性质,消除 ORTHOMIN(m)算法的内积计算数据相关性,给出了一种改进的 ORTHOMIN(m)(IORTHOMIN(m))算法。同 ORTHOMIN(m)算法对比,IORTHOMIN(m)算法与 ORTHOMIN(m)算法有相同的收敛性,在基于 MPI 的分布式存储并行机群上进行并行计算时,同步开销次数减少为 ORTHOMIN(m)算法的一半。数值计算结果与理论分析表明改进的 IORTHOMIN(m)算法的性能要优于 ORTHOMIN(m)算法。
关键词: ORTHOMIN(m)算法;并行计算;同步开销;非对称稀疏线性方程组
DOI: 10.3778/j.issn.1002-8331.2009.06.015　**文章编号:** 1002-8331(2009)06-0052-03　**文献标识码:** A　**中图分类号:** TP301.6

1　引言

天气预报、结构分析、热传导、电力网格分析、地球物理勘探等科学与工程计算中的许多问题都可以归结为求解线性方程组问题。Krylov 子空间方法常用来求解稀疏线性方程组,是许多应用软件的核心算法之一。前人对其并行实现已经作了很多工作,其中为了减少同步开销,D'Azevedo 等人[1]对 CG 算法进行了修改,得到了改进的 CG 算法,Merurant[2]和 Saad[3]也分别讨论了如何减少 CG 算法在并行机上的同步开销。刘等人[4]对 CR 算法进行了改进,给出了 CR 算法的改进形式 ICR 算法。这些算法常用来求解对称线性方程组。

广义共轭残差法(GCR)[5]是一种 Krylov 子空间方法,常用来求解新一代多尺度气象预报模式中三维赫姆霍兹方程[6-7]离散后得到的非对称稀疏线性方程组。但随着迭代步数的增加其计算量和存储量呈线性增长,因此在实际求解过程中,常采用截断技术或再启动技术。ORTHOMIN(m)[8]算法即是采用截断技术的 GCR 算法。ORTHOMIN(m)算法需要向量内积、矩阵向量乘积和向量校正的计算。在分布式存储的计算机上,矩阵和向量是分布在各台处理机上的,向量内积和矩阵向量乘积计算需要各处理机间的通信。必须进行全局通信的内积计算成为 ORTHOMIN(m)算法并行计算的瓶颈。针对非对称稀疏线性方程组的求解,利用 ORTHOMIN(m)算法的固有性质,通过改变算法的计算次序提出了改进的 ORTHOMIN(m)(IORTHOMIN(m))算法。该算法具有与 ORTHOMIN(m)算法相同的收敛速度,而同步开销次数减少为 ORTHOMIN(m)算法的一半。同时在基于 MPI 的分布式存储并行机群上给出数值示例。

2　算法表述

给定线性矩阵方程组:

$$\boldsymbol{Ax}=\boldsymbol{b} \tag{1}$$

其中矩阵 $\boldsymbol{A}\in R^{N\times N}$ 为非奇异非对称矩阵,$\boldsymbol{b}\in R^N$,$\boldsymbol{x}\in R^N$。

对于线性方程组(1)给出 ORTHOMIN(m)算法,如算法 1,初始值取 $x_0=0$。

算法 1　ORTHOMIN(m)算法

(1)计算 $r_0=\boldsymbol{b}-\boldsymbol{A}x_0$,取 $p_0=r_0$,$\boldsymbol{A}p_0=\boldsymbol{A}r_0$;

基金项目: 重庆市科委资助项目(the Foundation of Chongqing Science and Technology Commission under Grant No.CST2005BB0061);重庆市教委资助项目(the Foundation of Chongqing Municipal Education Commission under Grant No.KJ070514)。
作者简介: 赵利斌(1983-),男,硕士研究生,主要研究方向:并行数值计算理论与算法设计;田有先(1948-),男,教授,硕士生导师,主要研究方向:不动点理论、数值计算与高性能计算等。
收稿日期: 2008-01-17　**修回日期:** 2008-04-02

(2)对 $j=0,1,2\cdots$ 直到收敛计算：

(3)$\alpha_j=(r_j,\boldsymbol{A}p_j)/(\boldsymbol{A}p_j,\boldsymbol{A}p_j)$

(4)$x_{j+1}=x_j+\alpha_j p_j$

(5)$r_{j+1}=r_j-\alpha_j\boldsymbol{A}p_j$

(6)$\beta_{ij}=-(\boldsymbol{A}r_{j+1},\boldsymbol{A}p_i)/(\boldsymbol{A}p_i,\boldsymbol{A}p_i)$，$i=j-m+1,\cdots,j$

(7)$p_{j+1}=r_{j+1}+\sum\limits_{i=j-m+1}^{j}\beta_{ij}p_i$

(8)$\boldsymbol{A}p_{j+1}=\boldsymbol{A}r_{j+1}+\sum\limits_{i=j-m+1}^{j}\beta_{ij}\boldsymbol{A}p_i$

(9)结束计算

算法 1(ORTHOMIN(m)算法)的前 m 步迭代，以 GCR 算法的形式计算。从算法 1 中可以看出内积计算需要两次全局同步开销，即步(3)和步(6)。

为得到改进算法，利用 ORTHOMIN(m)算法中向量 $\boldsymbol{A}p_j$ 具有正交性[8]，即$(\boldsymbol{A}p_i,\boldsymbol{A}p_j)=0$，$i\neq j$，通过数学公式推导消除算法 1 中两次内积计算的数据相关性，用一次全局同步来完成所有内积计算。

由算法 1 中步(3)和步(5)及向量 $\boldsymbol{A}p_j$ 的正交性质，当 $i\leqslant j$ 时，有

$$(r_{j+1},\boldsymbol{A}p_i)=(r_j,\boldsymbol{A}p_i)-\alpha_j(\boldsymbol{A}p_j,\boldsymbol{A}p_i)=(r_{j-1},\boldsymbol{A}p_i)-\alpha_{j-1}(\boldsymbol{A}p_{j-1},\boldsymbol{A}p_i)=(r_i,\boldsymbol{A}p_i)-\alpha_i(\boldsymbol{A}p_i,\boldsymbol{A}p_i)=0 \tag{2}$$

令 $q_j=\boldsymbol{A}p_j$，$t_j=\boldsymbol{A}r_j$，则由算法 1 中步(6)和步(8)及式(2)得

$$(r_{j+1},q_{j+1})=(r_{j+1},\boldsymbol{A}p_{j+1})=(r_{j+1},\boldsymbol{A}r_{j+1})+\sum_{i=j-m+1}^{j}\beta_{ij}(r_{j+1},\boldsymbol{A}p_i)=(r_{j+1},t_{j+1}) \tag{3}$$

$$(q_{j+1},q_{j+1})=(t_{j+1},t_{j+1})+2\sum_{i=j-m+1}^{j}\beta_{ij}(t_{j+1},q_i)+\sum_{i=j-m+1}^{j}\beta_{ij}^2(q_i,q_i)=(t_{j+1},t_{j+1})-\sum_{i=j-m+1}^{j}\beta_{ij}^2(q_i,q_i) \tag{4}$$

从式(4)可以看出内积(q_j,q_j)可以通过迭代的方法求出，从而消除 ORTHOMIN(m)算法中两次内积计算中的数据相关性。

令 $a_j=(r_j,t_j)$，$b_j=(q_j,q_j)$，$c_j=(t_j,t_j)$，$d_{ij}=(t_j,q_i)$，$i=j-m,\cdots,j-1$ 则改进的 ORTHOMIN(m)算法如算法 2 所述。

算法 2 IORTHOMIN(m)算法

(1)计算 $r_0=\boldsymbol{b}-\boldsymbol{A}x_0$，$t_0=\boldsymbol{A}r_0$，取，$p_0=r_0$，$q_0=t_0$；

(2)对 $j=0,1,2\cdots$ 直到收敛计算：

(3)$\alpha_j=a_j/b_j$

(4)$x_{j+1}=x_j+\alpha_j p_j$

(5)$r_{j+1}=r_j-\alpha_j q_j$

(6)$t_{j+1}=\boldsymbol{A}r_{j+1}$

(7)$a_{j+1}=(r_{j+1},t_{j+1})$，$c_{j+1}=(t_{j+1},t_{j+1})$，$d_{i(j+1)}=(t_{j+1},q_i)$，$i=j-m+1,\cdots,j$

(8)$\beta_{ij}=-d_{i(j+1)}/b_i$，$i=j-m+1,\cdots,j$

(9)$b_{j+1}=c_{j+1}-\sum\limits_{i=j-m+1}^{j}\beta_{ij}^2 b_i$

(10)$p_{j+1}=r_{j+1}+\sum\limits_{i=j-m+1}^{j}\beta_{ij}p_i$

(11)$q_{j+1}=t_{j+1}+\sum\limits_{i=j-m+1}^{j}\beta_{ij}q_i$

(12)结束计算

算法 2(IORTHOMIN(m)算法)中 $i=\max\{0,j-m+1\}$，从算法 2 中可以看出内积计算只需要一次全局同步开销。

3 算法分析与比较

在分布式存储并行机群结构模型中，假设有 P 台处理机。在两个给定的处理器间完成单位字的一次通信时间为启动时间 t_s(包括打包、执行选路算法和建立通信界面的时间)和传输单位字的时间 t_w 之和。

在这里以按行划分为例讨论，设每个浮点运算时间均为 t_f，矩阵阶数 N 能被处理器台数 P 整除，假设不存在负载不平衡的问题。对一般的稀疏矩阵结构并行矩阵向量乘的纯计算时间为$(2q-1)(N/P)t_f$(每行进行了 $q-1$ 次加法，q 次乘法)，其中 q 为每行非零元素个数。而其通信时间与系数矩阵结构以及矩阵划分方式相关。在最坏情况下，按行划分需要的矩阵向量乘要用到全部向量，而向量分布存储在每个机器上，因此需要一次全收集(ALL_GATHER)的通信，通信时间为 $2(t_s+t_wN/P)\log P$。而对一般的稀疏矩阵结构，只需相邻处理机之间进行通信，其通信时间为 $2(t_s+qt_w)$，与处理器台数 P 和矩阵规模 N 无关。下面考虑这种情况的矩阵向量乘通信时间，此时矩阵向量乘的并行计算时间为

$$t_{mv}=(2q-1)(N/P)t_f+2(t_s+qt_w) \tag{5}$$

对于 k 个向量内积并行计算时间，每台处理机并行求解局部向量内积，计算时间为 $2(kN/P)t_f$。为计算 k 个整体内积需要对这 k 个局部内积均进行一个全归约(ALL_REDUCE)的通信。其通信时间为 $2(t_s+kt_w)\log P$。因此，k 个向量内积的并行计算时间为

$$t_{in}(k)=2(kN/P)t_f+2(t_s+kt_w)\log P \tag{6}$$

对于 k 个向量相加的向量校正，其不需要通信，故其并行计算时间为

$$t_{ve}(k)=2(k-1)(N/P)t_f \tag{7}$$

由于 ORTHOMIN(m)算法经过 m 次迭代后其每一个迭代步的计算和通信模式完全相同，且其为算法迭代的主要部分。为方便讨论，对算法的分析比较时只考虑 m 次迭代后的一个迭代步的并行计算和通信时间复杂性。对 IORTHOMIN(m)算法作类似的考虑。

由以上讨论知，ORTHOMIN(m)算法中的每步迭代中需要 1 次矩阵向量乘、4 次向量校正和 $m+2$ 次内积计算，m 个参数 β_{ij} 和 1 个参数 α_j 校正的浮点计算，该算法每步迭代并行计算时间为

$$t^j=t_{mv}+t_{in}(m)+t_{in}(2)+2t_{ve}(2)+2t_{ve}(m+1)+(m+1)t_f=(2q+6m+7)(N/P)t_f+(m+1)t_f+2(2t_s+(m+2)t_w)\log P+2(t_s+qt_w) \tag{8}$$

对于 IORTHOMIN(m)算法中的每步迭代中则需要 1 次矩阵向量乘、4 次向量校正和 $m+2$ 次向量内积计算，m 个参数 β_{ij} 和 1 个参数 α_j 校正的浮点计算，b_{j+1} 校正的浮点计算，该算法一步迭代并行计算时间为

$$\tilde{t}^j=t_{mv}+t_{in}(m+2)+2t_{ve}(2)+2t_{ve}(m+1)+(4m+1)t_f+(2q+6m+7)(N/P)t_f+(4m+1)t_f+2(t_s+(m+2)t_w)\log P+2(t_s+qt_w) \tag{9}$$

由于 m 的选取一般不大即有 $m\cdot N$，比较式(8)和式(9)知，随着处理机台数增加，相对于全局通信而言，IORTHOMIN(m)算法相对 ORTHOMIN(m)算法几乎不增加计算量。

由式(8)和式(9)得，当处理机台数满足下式时

$$\log P>\frac{3mt_f}{2t_s} \tag{10}$$

IORTHOMIN(m)算法比 ORTHOMIN(m)算法的并行计算时间要好。

由式(8)求解$\frac{\partial t^j}{\partial P}=0$可以得到 ORTHOMIN($m$)算法并行计算时间最小时的处理机台数

$$P=\frac{(2q+6m+7)Nt_f\ln 2}{4t_s+(2m+4)t_w} \tag{11}$$

由式(9)求解$\frac{\partial \tilde{t}^j}{\partial P}=0$可以得到 IORTHOMIN($m$)算法并行计算时间最小时的处理机台数

$$\tilde{P}=\frac{(2q+6m+7)Nt_f\ln 2}{2t_s+(2m+4)t_w} \tag{12}$$

在分布式存储并行机中 $t_s>>t_w$,且两算法中的选取一般不大,故有$\tilde{P}\approx 2P$,从该式可以看出 IORTHOMIN(m)算法具有更好的扩展性。

当 $t_s>>t_w>t_f$ 时,与 ORTHOMIN(m)算法相比,IORTHOMIN(m)算法的性能提高比率为

$$\eta=\frac{t^j-\tilde{t}^j}{t^j}\approx\frac{2t_sP\log P}{(2q+6m+7)Nt_f+4t_sP\log P+2t_sP}\to 50\%,P\to\infty \tag{13}$$

式(13)说明 IORTHOMIN(m)算法与 ORTHOMIN(m)算法相比性能提高比可达到 50%。

4 数值示例

本文数值计算的测试环境为包括 8 个节点的工作站机群,主节点的基本配置为 Intel Pentium 4 2.0 G,512 M DDR,其余各节点的基本配置为 Intel Pentium 4 2.0 G,512 M DDR。各节点通过 1 000 Mb/s 的以太网连接在一起。并行计算在基于消息传递的分布式并行编程环境 MPI 上实现。

本章将通过对阶数分别为 2 400、4 000 的两个非对称稀疏线性方程组的求解,来比较 ORTHOMIN(m)算法和 IORTHOMIN(m)算法,求解中通过从外存读取矩阵系数来计算矩阵向量乘,表中 η 为式(13),取可接受的残差值 R=0.000 01。

表 1 2 400 阶矩阵 ORTHOMIN(m)算法和 IORTHOMIN(m)算法的计算性能对比

处理机数	ORTHOMIN(m)/s	IORTHOMIN(m)/s	η/(%)
1	114.6	115.8	-1.05
2	56.7	54.8	3.35
4	35.6	33.5	5.90
8	23.2	21.1	9.05

表 2 4 000 阶矩阵 ORTHOMIN(m)算法和 IORTHOMIN(m)算法的计算性能对比

处理机数	ORTHOMIN(m)/s	IORTHOMIN(m)/s	η/(%)
1	317.5	319.8	-0.72
2	154.8	148.3	4.20
4	85.4	76.5	10.42
8	55.7	48.2	13.46

从表 1 及表 2 中的数据的比较中可以看出,随着处理机台数的增加,IORTHOMIN(m)算法比 ORTHOMIN(m)算法的性能提高比率逐渐增加,这说明 IORTHOMIN(m)算法比 ORTHOMIN(m)算法有更好的扩展性,其更适合并行计算。IORTHOMIN(m)算法比 ORTHOMIN(m)算法性能提高的多少依赖于并行机速度的快慢、机器互连网络的结构和速度以及求解问题的性质与规模等多种因素。

5 结论

给出了对于求解大型非对称稀疏线性方程组的一种改进的 ORTHOMIN(m)算法。其与 ORTHOMIN(m)算法有相同的收敛速度,几乎不增加计算量。在基于 MPI 的分布式存储并行机群结构上,通过消除内积计算数据相关性使其同步开销次数减少为 ORTHOMIN(m)算法的一半。从理论和数值分析两方面说明了 IORTHOMIN(m)算法对于求解大型非对称稀疏线性方程组是一种具有较好扩展性的算法,与 ORTHOMIN(m)算法相比更利于并行实现,更适合作为应用软件的核心算法。在求解新一代多尺度气象预报模式中三维赫姆霍兹方程等工程计算问题时应能进一步提高并行执行效率。

参考文献:

[1] D'Azevedo E,Romaine C.Reducing communication costs in the conjugate gradient algorithm on distributed memory multiprocessors[R].University of Tennessee,Knoxville,1992.

[2] Merurant G.Multitasking the conjugate gradient on the Cray X-MP/48[J].Parallel Computing,1987,5(2):267-280.

[3] Saad Y.Krylov subspace methods on supercomputers[J].SIAM Journal on Scientific and Statistical Computing,1989,10(8):1200-1232.

[4] 刘杰,刘兴平,迟利华,等.一种改进的适合并行计算的共轭剩余算法[J].计算机学报,2006,29(3):495-499.

[5] Saad Y.Iterative methods for sparse linear systems[M].2nd ed.Boston:PWS Publishing Company,2003:143-200.

[6] 张理论.面向气象预报数值模拟的高效并行计算研究[D].长沙:国防科学技术大学研究生院,2002.

[7] 沈桐立,田永祥,葛孝贞,等.数值天气预报[M].北京:气象出版社,2003:421-449.

[8] Eisenstat S C,Elman H C,Schultz M H.Variational iterative methods for non-symmetric systems of linear equations[J].SIAM Journal on Numerical and Analysis,1983,20(2):345-357.

[9] 刘杰,刘兴平,迟利华,等.一种改进的适合并行计算的 TFQMR 算法[J].计算机研究与发展,2005,42(7):1235-1240.

[10] Meza J C,Symes W W.Conjugate residual methods for almost symmetric linear systems [J].Journal of Optimization Theory and Applications,1992,72(3):415-440.

[11] Yang L T,Brent R P.The improved BiCGSTAB method for large and sparse un-symmetric linear systems on parallel distributed memory architectures[C]//Proceeding of the 5th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP-02),Beijing,2002:324-328.

[12] D'Azevedo E,Eijkhout V,Romaine C.A matrix framework for conjugate gradient methods and some variants of CG with less synchronization overhead[C]//Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing,1993:644-646.

[13] Matheswaran M,Webb K J,Siegel H J.MCGS:A modified conjugate gradient squared algorithm for non-symmetric linear systems[J].The Journal of Supercomputing,2001,9(1):1-25.