循环三对角 Toeplitz 线性方程组的分组降阶算法

李文强,刘 晓

河南师范大学数学与信息科学学院, 河南新乡 453007

摘要 运用并行算法中分而治之的思想,给出了一种求解循环三对角 Toeplitz 线性方程组的分组降阶串行算法。与求解同类问题的传统算法相比,分组降阶算法的优点在于它不仅大幅度减少了内存占用量,而且还大幅度减少了算术运算量。分组降阶算法可以通过 3 个步骤来实现。第一步是分组降阶,其基本思路是将一个 $n=\mu m$ 阶的方程组按行分成 μ 组,每组 m 个方程;n 维解向量也对应地分成 μ 组。第二步是构造参数方程组,也就是依据三对角系数矩阵的特点,给出各组解之间的关系式,把不属于该组的解分量看作参数。第三步是求解参数方程组和原方程组,在这一步中,首先求解参数方程组,然后再代入相应分组的关系式便可求出所有的解分量。对于三对角 Toeplitz 线性方程组,同样能减少内存占用量,从而在计算机性能不变的情况下,提高求解问题的规模,但与求解三对角 Toeplitz 线性方程组的传统算法相比运算量有所增加。数值实验结果表明,对于特定规模的方程组来说,总存在一个最佳的分组个数使得计算时间最少;随着方程组阶数的提高,最佳分组的个数也增大。

关键词 三对角 Toeplitz 线性方程组;循环三对角 Toeplitz 线性方程组;分组降阶算法

中图分类号 O242　　**文献标识码** A　　**doi** 10.3981/j.issn.1000-7857.2012.05.006

Grouping and Order Reducing Algorithm for Solving Toeplitz Type Circular Tri-diagonal Liner Algebraic Equation Systems

LI Wenqiang, LIU Xiao

College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, Henan Province, China

Abstract Based on the idea of the divide and conquer method, as is commonly adopted in parallel algorithms, a grouping and order reducing sequence algorithm is proposed for solving the Toeplitz type circular tri-diagonal linear algebraic equation systems in this paper. Compared with the traditional algorithm for the same problem, the advantage of this grouping and order reducing algorithm is that the computation cost and the computer memory requirement can be reduced. The whole algorithm includes three steps. The first step is grouping and order reducing of the original system. To put it better, the coefficient matrix and the right hand side of a Toeplitz type circular tri-diagonal system of order $n=\mu m$ is divided into μ subgroups. Consequently, the order of each subgroup is . The second step is the formation of the parameter equations. That is to say, according the characteristics of the tri-diagonal coefficient matrix, the relations between subgroups are obtained, and the solution components, which do not belong to the subgroups, are taken as parameters. Then, a parameter equation is formed based on the equation that includes the parameters. The third step is to solve the parameter equation. And then, the original system is solved by substituting the parameters into the corresponding subgroups. As for the tri-diagonal system, the grouping and order reducing algorithm can also reduce the requirement for the computer memory and increase the order of the system, but at the same time , increase the computation cost. Numerical experiments show that, on one hand, there is an optimal number for grouping in saving the computation cost if the order of the system is fixed. On the other hand, the optimal number for groupings increases with the increase of the order of the system.

Keywords Toeplitz type tridiagonal linear system; Toeplitz type circular tridiagonal linear system; grouping and reducing order algorithm

收稿日期:2011-09-19;修回日期:2012-01-09

基金项目:河南省教育厅自然科学研究计划项目(2010B110014);河南师范大学青年骨干教师支持计划

作者简介:李文强,副教授,研究方向为数值代数,电子信箱:liwenqiangxx@126.com

0　引言

在工程计算中常常会遇到循环三对角 Toeplitz 线性方程组

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

其中

$$\boldsymbol{A}=\begin{bmatrix} b & c & & & & a \\ a & b & c & & & \\ & \ddots & \ddots & \ddots & & \\ & & a & b & c & \\ c & & & a & b & \end{bmatrix},\ \boldsymbol{x}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix},\ \boldsymbol{f}=\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix}$$

如果

$$\boldsymbol{A}=\begin{bmatrix} b & c & & & & \\ a & b & c & & & \\ & \ddots & \ddots & \ddots & & \\ & & a & b & c & \\ & & & a & b & \end{bmatrix}$$

则方程组(1)就是三对角 Toeplitz 线性方程组。

对三对角 Toeplitz 线性方程组求解方法的研究一直是人们关注的问题[1-6]。对于 n 阶循环三对角 Toeplitz 线性方程组的求解,用传统的追赶法[1],内存占用量约为 $4n$ 个单元(1 单元=1 个浮点数(real 型数据)占用的字节单元个数),算术运算次数约为 $14n$;李青等[2]给出的算法,内存占用量约为 $5n$,算术运算量约为 $17n$;王兴波等[3]给出的算法,内存占用量约为 $5n$,算术运算量约为 $19n$;李文强等[4]给出的算法,内存占用量约为 $4n$,算术运算量为 $14n$。本文给出的分组降阶算法,内存占用量约为 $n+4m+6\mu$,算术运算量为 $9n+9m+10\mu$,其中 $n=\mu m$。

1　分组降阶算法分析

1.1　拟三对角 Toeplitz 线性方程组

假定 $n=\mu m(n,m,\mu\in\mathbf{Z}^+)$,把循环三对角 Toeplitz 线性方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 作如下分组

$$\boldsymbol{A}=\begin{bmatrix} \boldsymbol{T} & \boldsymbol{R} & & & & \boldsymbol{L} \\ \boldsymbol{L} & \boldsymbol{T} & \boldsymbol{R} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \boldsymbol{L} & \boldsymbol{T} & \boldsymbol{R} & \\ \boldsymbol{R} & & & \boldsymbol{L} & \boldsymbol{T} & \end{bmatrix},\ \boldsymbol{x}=\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{\mu-1} \\ \boldsymbol{x}_\mu \end{bmatrix},\ \boldsymbol{f}=\begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_{\mu-1} \\ \boldsymbol{f}_\mu \end{bmatrix}$$

这里 $\boldsymbol{L},\boldsymbol{R},\boldsymbol{T}\in\mathbf{R}^{m\times m}$,$\boldsymbol{x}_i,\boldsymbol{f}_i\in\mathbf{R}^m(i=1,2,\cdots,\mu)$,且

$$\boldsymbol{T}=\begin{bmatrix} b & c & & \\ a & \ddots & \ddots & \\ & \ddots & \ddots & c \\ & & a & b \end{bmatrix},\ \boldsymbol{x}_i=\begin{bmatrix} x_{1,i} \\ x_{2,i} \\ \vdots \\ x_{m,i} \end{bmatrix}=\begin{bmatrix} x_{(i-1)m+1} \\ x_{(i-1)m+2} \\ \vdots \\ x_{(i-1)m+m} \end{bmatrix},\ \boldsymbol{f}_i=\begin{bmatrix} f_{1,i} \\ f_{2,i} \\ \vdots \\ f_{m,i} \end{bmatrix}=\begin{bmatrix} f_{(i-1)m+1} \\ f_{(i-1)m+2} \\ \vdots \\ f_{(i-1)m+m} \end{bmatrix}$$

$$\boldsymbol{L}=a\boldsymbol{e}_1\boldsymbol{e}_m^{\mathrm{T}}=\begin{bmatrix} 0 & \cdots & 0 & a \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix},\ \boldsymbol{R}=c\boldsymbol{e}_m\boldsymbol{e}_1^{\mathrm{T}}=\begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \\ c & 0 & \cdots & 0 \end{bmatrix},$$

其中,$\boldsymbol{e}_1,\boldsymbol{e}_m$ 是 $\mathbf{R}^m$ 的单位向量。于是方程组(1)可以写成等价形式

$$\begin{cases} \boldsymbol{Lx}_\mu+\boldsymbol{Tx}_1+\boldsymbol{Rx}_2=\boldsymbol{f}_1 \\ \boldsymbol{Lx}_{i-1}+\boldsymbol{Tx}_i+\boldsymbol{Rx}_{i+1}=\boldsymbol{f}_i & i=2,3,\cdots,\mu-1 \\ \boldsymbol{Lx}_{\mu-1}+\boldsymbol{Tx}_\mu+\boldsymbol{Rx}_1=\boldsymbol{f}_\mu \end{cases} \tag{2}$$

用 $\boldsymbol{T}^{-1}$ 左乘式(2)两边,并注意到 $\boldsymbol{Lx}_i=x_{m,i}a\boldsymbol{e}_1$,$\boldsymbol{Rx}_i=x_{1,i}c\boldsymbol{e}_m$,则有

$$\begin{cases} x_{m,\mu}\boldsymbol{T}^{-1}(a\boldsymbol{e}_1)+\boldsymbol{x}_1+x_{1,2}\boldsymbol{T}^{-1}(c\boldsymbol{e}_m)=\boldsymbol{T}^{-1}\boldsymbol{f}_1 \\ x_{m,i-1}\boldsymbol{T}^{-1}(a\boldsymbol{e}_1)+\boldsymbol{x}_i+x_{1,i+1}\boldsymbol{T}^{-1}(c\boldsymbol{e}_m)=\boldsymbol{T}^{-1}\boldsymbol{f}_i,i=2,3,\cdots,\mu-1 \\ x_{m,\mu-1}\boldsymbol{T}^{-1}(a\boldsymbol{e}_1)+\boldsymbol{x}_\mu+x_{1,1}\boldsymbol{T}^{-1}(c\boldsymbol{e}_m)=\boldsymbol{T}^{-1}\boldsymbol{f}_\mu \end{cases}$$

令

$$\boldsymbol{s}=\boldsymbol{T}^{-1}(a\boldsymbol{e}_1)=(s_1,s_2,\cdots,s_m)^{\mathrm{T}},\boldsymbol{t}=\boldsymbol{T}^{-1}(c\boldsymbol{e}_m)=(t_1,t_2,\cdots,t_m)^{\mathrm{T}},$$
$$\boldsymbol{w}_i=\boldsymbol{T}^{-1}\boldsymbol{f}_i=(w_{1,i},w_{2,i},\cdots,w_{m,i})^{\mathrm{T}}\quad(i=1,2,\cdots,\mu) \tag{3}$$

便得到关系式

$$\begin{cases} \boldsymbol{x}_1=\boldsymbol{w}_1-x_{m,\mu}\boldsymbol{s}-x_{1,2}\boldsymbol{t} \\ \boldsymbol{x}_i=\boldsymbol{w}_i-x_{m,i-1}\boldsymbol{s}-x_{1,i+1}\boldsymbol{t} & i=2,3,\cdots,\mu-1 \\ \boldsymbol{x}_\mu=\boldsymbol{w}_\mu-x_{m,\mu-1}\boldsymbol{s}-x_{1,1}\boldsymbol{t} \end{cases} \tag{4}$$

式(4)就是该算法中分组解之间的关系式,称解分量 $x_{1,i},x_{m,i}$ $(i=1,2,\cdots,\mu)$为参数,共 2μ 个。

显然,式(3)含有 $\mu+2$ 个 m 阶的三对角 Toeplitz 线性方程组,可以全部采用追赶法求解。为了求出参数方程中的 2μ 个参数,从关系式(4)的每组方程中取出第 1 和第 m 两个方程,组成参数方程

$$\begin{cases} s_1x_{m,\mu}+x_{1,1}+x_{1,2}t_1=w_{1,1} \\ s_mx_{m,\mu}+x_{m,1}+x_{1,2}t_m=w_{m,1} \\ \begin{cases} s_1x_{m,i-1}+x_{1,i}+t_1x_{1,i+1}=w_{1,i} \\ s_mx_{m,i-1}+x_{m,i}+t_mx_{1,i+1}=w_{m,i} \end{cases} & i=2,3,\cdots,\mu-1 \\ s_1x_{m,\mu-1}+x_{1,\mu}+t_1x_{1,1}=w_{1,\mu} \\ s_mx_{\mu-1,m}+x_{\mu,m}+t_mx_{1,1}=w_{\mu,m} \end{cases} \tag{5}$$

其矩阵形式为

$$\begin{bmatrix} 1 & 0 & t_1 & & & & & & s_1 \\ 0 & 1 & t_m & 0 & & & & & s_m \\ 0 & s_1 & 1 & 0 & t_1 & & & & \\ & s_m & 0 & 1 & t_m & 0 & & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & & 0 & s_1 & 1 & 0 & t_1 & \\ & & & & s_m & 0 & 1 & t_m & 0 \\ t_1 & & & & & 0 & s_1 & 1 & 0 \\ t_m & & & & & & s_m & 0 & 1 \end{bmatrix}\begin{bmatrix} x_{1,1} \\ x_{m,1} \\ x_{1,2} \\ x_{m,2} \\ \vdots \\ x_{1,\mu-1} \\ x_{m,\mu-1} \\ x_{1,\mu} \\ x_{m,\mu} \end{bmatrix}=\begin{bmatrix} w_{1,1} \\ w_{m,1} \\ w_{1,2} \\ w_{m,2} \\ \vdots \\ w_{1,\mu-1} \\ w_{m,\mu-1} \\ w_{1,\mu} \\ w_{m,\mu} \end{bmatrix} \tag{6}$$

适当进行列交换,参数方程可化成为拟三对角方程,即

$$\begin{bmatrix} s_1 & t_1 & & & & & & & 1 \\ s_m & t_m & 1 & & & & & & \\ & 1 & s_1 & t_1 & & & & & \\ & & s_m & t_m & 1 & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & 1 & s_1 & t_1 & & \\ & & & & & s_m & t_m & 1 & \\ & & & & & & 1 & s_1 & t_1 \\ 1 & & & & & & & s_m & t_m \end{bmatrix}\begin{bmatrix} x_{m,\mu} \\ x_{1,2} \\ x_{m,1} \\ x_{1,3} \\ \vdots \\ x_{m,\mu-2} \\ x_{1,\mu} \\ x_{m,\mu-1} \\ x_{1,1} \end{bmatrix}=\begin{bmatrix} w_{1,1} \\ w_{m,1} \\ w_{1,2} \\ w_{m,2} \\ \vdots \\ w_{1,\mu-1} \\ w_{m,\mu-1} \\ w_{1,\mu} \\ w_{m,\mu} \end{bmatrix} \tag{7}$$

1.2　三对角 Toeplitz 线性方程组

仍然假定 $n=\mu m(n,m,\mu\in\mathbf{Z}^+)$,三对角 Toeplitz 线性方程组 $\boldsymbol{Ax}=\boldsymbol{f}$ 的分组形式为

$$\begin{cases}\boldsymbol{Tx}_1+\boldsymbol{Rx}_2=\boldsymbol{f}_1\\ \boldsymbol{Lx}_{i-1}+\boldsymbol{Tx}_i+\boldsymbol{Rx}_{i+1}=\boldsymbol{f}_i & i=2,3,\cdots,\mu-1\\ \boldsymbol{Lx}_{\mu-1}+\boldsymbol{Tx}_\mu=\boldsymbol{f}_\mu\end{cases}\tag{8}$$

令

$$\boldsymbol{s}=\boldsymbol{T}^{-1}(a\boldsymbol{e}_1)\quad \boldsymbol{t}=\boldsymbol{T}^{-1}(c\boldsymbol{e}_m)\quad \boldsymbol{w}_i=\boldsymbol{T}^{-1}\boldsymbol{f}_i,\quad i=1,2,\cdots,\mu\tag{9}$$

便得到关系式

$$\begin{cases}\boldsymbol{x}_1=\boldsymbol{w}_1-x_{1,2}\boldsymbol{t}\\ \boldsymbol{x}_i=\boldsymbol{w}_i-x_{m,i-1}\boldsymbol{s}-x_{1,i+1}\boldsymbol{t} & i=2,3,\cdots,\mu-1\\ \boldsymbol{x}_\mu=\boldsymbol{w}_\mu-x_{m,\mu-1}\boldsymbol{s}\end{cases}\tag{10}$$

称 $x_{1,i}(i=2,3,\cdots,\mu)$和 $x_{m,i}(i=1,2,\cdots,\mu-1)$共 $2(\mu-1)$个解分量为参数。

显然,式(9)含有 $\mu+2$ 个 m 阶的三对角 Toeplitz 线性方程组,可以全部采用追赶法求解。为了求出参数方程中的 $2(\mu-1)$个参数,从关系式(10)的第 1 组方程中取出第 m 个方程,第 $i(i=2,3,\cdots,\mu-1)$组方程中取出第 1,2 两个方程,最后 1 组方程中取出第 1 个方程,组成参数方程

$$\begin{cases}x_{m,1}+t_m x_{1,2}=w_{m,1}\\ \begin{cases}s_1x_{m,i-1}+x_{m,i}+t_1x_{1,i+1}=w_{1,i}\\ s_mx_{m,i-1}+x_{m,i}+t_mx_{1,i+1}=w_{m,i}\end{cases} & i=2,3,\cdots,\mu-1\\ s_1x_{m,\mu-1}+x_{1,\mu}=w_{1,\mu}\end{cases}\tag{11}$$

其矩阵形式为

$$\begin{bmatrix}1 & t_m & 0 & & & & & \\ s_1 & 1 & 0 & t_1 & & & & \\ s_m & 0 & 1 & t_m & 0 & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & 0 & s_1 & 1 & 0 & t_1 & \\ & & & s_m & 0 & 1 & t_m & \\ & & & & 0 & s_1 & 1 & \end{bmatrix}\begin{bmatrix}x_{m,1}\\ x_{1,2}\\ x_{m,2}\\ \vdots\\ x_{1,\mu-1}\\ x_{m,\mu-1}\\ x_{1,\mu}\end{bmatrix}=\begin{bmatrix}w_{m,1}\\ w_{1,2}\\ w_{m,2}\\ \vdots\\ w_{1,\mu-1}\\ w_{m,\mu-1}\\ w_{1,\mu}\end{bmatrix}\tag{12}$$

适当进行列变换,可化成三对角方程

$$\begin{bmatrix}t_m & 1 & & & & & \\ 1 & s_1 & t_1 & & & & \\ & s_m & t_m & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & s_1 & t_1 & \\ & & & & s_m & t_m & 1\\ & & & & & 1 & s_1\end{bmatrix}\begin{bmatrix}x_{1,2}\\ x_{m,1}\\ x_{1,3}\\ \vdots\\ x_{m,\mu-2}\\ x_{1,\mu}\\ x_{m,\mu-1}\end{bmatrix}=\begin{bmatrix}w_{m,1}\\ w_{1,2}\\ w_{m,2}\\ \vdots\\ w_{1,\mu-1}\\ w_{m,\mu-1}\\ w_{1,\mu}\end{bmatrix}\tag{13}$$

2　分组降阶算法过程

2.1　参数方程组的求解方法

文中对应给出的参数方程矩阵表示两种形式:一种是三对角形式,即式(7)和式(13);另一种是五对角形式,即式(6)和式(12)。事实上,如果原方程组(1)的系数矩阵 $\boldsymbol{A}$ 是严格对角占优的,则 $\boldsymbol{s}$ 和 $\boldsymbol{t}$ 的各个分量接近于 0,这就意味着三对角形式的参数方程(7)和(13)的系数矩阵不是对角占优的,而五对角形式的参数方程(6)和(12)的系数矩阵却是严格对角占优的。因此,当分组数较小时,可以用(拟)三对角线性方程组的追赶法求解参数方程组(7)或(13)得到参数。当分组数 μ 较大时,最好通过求解参数方程组(6)或(12)得到参数。

下面给出求解方程组(12)的具体算法过程。首先经行初等变换把方程组(12)的增广矩阵化为

$$\left[\begin{array}{cccccccc|c}1 & q_1 & 0 & & & & & & d_1\\ & 1 & 0 & q_2 & & & & & d_2\\ & & 1 & q_3 & 0 & & & & d_3\\ & & & \ddots & \ddots & \ddots & & & \vdots\\ & & & & 1 & 0 & q_{2\mu-4} & & d_{2\mu-4}\\ & & & & & 1 & q_{2\mu-3} & & d_{2\mu-3}\\ & & & & & & 1 & & d_{2\mu-2}\end{array}\right]\tag{14}$$

再逐一回代求出全部参数。

算法 1(三对角分组降阶法参数方程求解过程)

$q_1:=t_m, d_1:=w_{m,1}$

$$\begin{cases}g:=1-s_1q_{2i-1}, q_{2i}:=t_1/g, d_{2i}:=(w_{1,i+1}-s_1d_{2i-1})/g\\ g:=s_mq_{2i-1}, q_{2i+1}:=t_m+gq_{2i}, d_{2i+1}:=w_{m,i+1}-s_md_{2i-1}+gd_{2i}\end{cases}\quad i=1,2,\cdots,\mu-2$$

$g:=1-s_1r_{2\mu-3}, d_{2\mu-2}:=(w_{1,\mu}-s_1d_{2\mu-3})/g$

$x_{1,\mu}:=d_{2\mu-2}$

$$\begin{cases}x_{m,i}:=d_{2i-1}-q_{2i-1}x_{1,i+1}\\ x_{1,i}:=d_{2i-2}-q_{2i-2}x_{m,i}\end{cases}\quad i=\mu-1,\mu-2,\cdots,2$$

$x_{m,1}:=d_1-q_1x_{1,2}$

其中,":="表示赋值。

算法 1 需要 10μ 次乘除运算和 7μ 次加减运算,并需要新增 2μ 个内存单元用来存储向量 $\boldsymbol{d}$。若用传统的追赶法求解方程组(13)则需要 10μ 次乘除运算和 6μ 次加减运算。通过算法 1 求解方程组(12)得到参数,虽然比用追赶法求解方程组(13)得到参数多用了 μ 次加减运算,但方程组(12)确保了数值稳定性。

再来讨论拟三对角分组降阶法对应的参数方程的求解过程。先把矩阵(6)的首末两列交换,即调换首末两个解分量,得到增广矩阵(以 $\mu=4$ 为例):

$$\left[\begin{array}{cccccccc|c}s_1 & 0 & t_1 & 0 & 0 & 0 & 0 & 1 & w_{1,1}\\ s_m & 1 & t_m & 0 & 0 & 0 & 0 & 0 & w_{m,1}\\ 0 & s_1 & 1 & 0 & t_1 & 0 & 0 & 0 & w_{1,2}\\ 0 & s_m & 0 & 1 & t_m & 0 & 0 & 0 & w_{m,2}\\ 0 & 0 & 0 & s_1 & 1 & 0 & t_1 & 0 & w_{1,3}\\ 0 & 0 & 0 & s_m & 0 & 1 & t_m & 0 & w_{m,3}\\ 0 & 0 & 0 & 0 & 0 & s_1 & 1 & t_1 & w_{1,4}\\ 1 & 0 & 0 & 0 & 0 & s_m & 0 & t_m & w_{m,4}\end{array}\right]$$

先从第三行起直到最后一行作消元,再从倒数第二行起直到第一行作消元,将上述增广矩阵化为下列同解形式:

$$\left[\begin{array}{cccccccc|c} p_1 & 0 & q_1 & 0 & 0 & 0 & 0 & 1 & d_1 \\ p_2 & 1 & q_2 & 0 & 0 & 0 & 0 & 0 & d_2 \\ p_3 & 0 & 1 & 0 & q_3 & 0 & 0 & 0 & d_3 \\ p_4 & 0 & 0 & 1 & q_4 & 0 & 0 & 0 & d_4 \\ p_5 & 0 & 0 & 0 & 1 & 0 & q_5 & 0 & d_5 \\ p_6 & 0 & 0 & 0 & 0 & 1 & q_6 & 0 & d_6 \\ p_7 & 0 & 0 & 0 & 0 & 0 & 1 & q_7 & d_7 \\ p_8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & d_8 \end{array}\right] \rightarrow$$

$$\left[\begin{array}{cccccccc|c} p_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_1 \\ p_2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & d_2 \\ p_3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & d_3 \\ p_4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & d_4 \\ p_5 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & d_5 \\ p_6 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & d_6 \\ p_7 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & d_7 \\ p_8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & d_8 \end{array}\right]$$

然后利用首末两个方程计算出 $x_{m,\mu}$，再逐个求出剩余的全部参数。

算法 2(拟三对角分组降阶法参数方程求解过程)

$p_1:=s_1, p_2:=s_m, q_1:=t_1, q_2:=t_m, d_1:=w_{1,1}, d_2:=w_{m,1}$

$$\begin{cases} g:=1-s_1t_m, p_{2i-1}:=-s_1p_{2i-2}/g, q_{2i-1}:=t_1/g, d_{2i-1}:=(w_{1,i}-s_1d_{2i-2})/g \\ g:=s_mt_m, p_{2i}:=gp_{2i-1}-s_mp_{2i-2}, q_{2i}:=t_m+gq_{2i-1}, d_{2i}:=w_{m,i}-s_md_{2i-2}+gd_{2i-1} \end{cases} \quad i=2,3,\cdots,\mu$$

$p_{2\mu}:=p_{2\mu}+1, p_{2\mu}:=p_{2\mu}/q_{2\mu}, d_{2\mu}:=d_{2\mu}/q_{2\mu}$

$p_{2\mu-1}:=p_{2\mu-1}-p_{2\mu}q_{2\mu-1}, d_{2\mu-1}:=d_{2\mu-1}-d_{2\mu}q_{2\mu-1}$

$p_1:=p_1-p_{2\mu}, d_1:=d_1-d_{2\mu}$

$$\begin{cases} p_{2i}:=p_{2i}-q_{2i}p_{2i+1}, d_{2i}:=d_{2i}-q_{2i}d_{2i+1} \\ p_{2i-1}:=p_{2i-1}-q_{2i-1}p_{2i+1}, d_{2i-1}:=d_{2i-1}-q_{2i-1}d_{2i+1} \end{cases} \quad i=\mu-1,\mu-2,\cdots,1$$

$x_{m,\mu}:=d_1/p_1, x_{m,1}:=d_2-p_{2i}x_{m,\mu}$

$x_{1,i}:=d_{2i-1}-p_{2i-1}x_{m,\mu}, x_{m,i}:=d_{2i}-p_{2i}x_{m,\mu} \quad i=2,3,\cdots,\mu-1$

$x_{1,\mu}:=d_{2\mu-1}-p_{2\mu-1}x_{m,\mu}, x_{1,1}:=d_{2\mu}-p_{2\mu}x_{m,\mu}$

算法 2 需要 18μ 次乘除运算和 12μ 次加减运算，新增 2μ 个内存单元用于存储向量 $\boldsymbol{d}$。若用追赶法直接求解拟三对角方程(7)则需要 18μ 次乘除运算和 10μ 次加减运算。通过算法 2 求解矩阵(6)得到参数，虽然增加了 2μ 次加减运算，但能确保数值稳定性。

2.2　循环三对角 Toeplitz 线性方程组的分组降阶算法

算法 3(求解循环三对角 Toeplitz 线性方程组的分组降阶算法)

第 1 步　数据初始化：设 $n=\mu m$，向量 $\boldsymbol{f}_i\in\mathbf{R}^m(i=1,2,\cdots,\mu)$，且

$$f_{j,i}:=f_{im+j}, i=1,2,\cdots,\mu; j=1,2,\cdots,m$$

第 2 步　分解 $\boldsymbol{T}=\mathrm{diag}(a,b,c)=\mathrm{diag}(a,p_i,0)\cdot\mathrm{diag}(0,1,q_i)\in\mathbf{R}^{m\times m}, \boldsymbol{p},\boldsymbol{q}\in\mathbf{R}^m$：

$p_1:=b, q_1:=c/p_1; p_i:=b-aq_{i-1}, q_i:=c/p_i \quad i=2,3,\cdots,m-1;$

$p_m:=b-aq_{m-1}$

第 3 步　追赶法求解 $\boldsymbol{Ts}=a\boldsymbol{e}_1, \boldsymbol{Tt}=c\boldsymbol{e}_m, \boldsymbol{Tw}_i=\boldsymbol{f}_i$ $(i=1,2,\cdots,\mu)$：

$s_1:=a/p_1, w_{1,i}:=w_{1,i}/p_1 \quad i=1,2,\cdots,\mu$

$$\begin{cases} s_j:=-as_{j-1}/p_j \\ w_{j,i}:=(f_{j,i}-aw_{j-1,i})/p_j \end{cases} \quad i=1,2,\cdots,\mu \qquad j=2,3,\cdots,m$$

$t_m:=c/p_m$

$$\begin{cases} s_j:=s_j-q_js_{j+1}, t_j:=t_j-q_jt_{j+1} \\ w_{j,i}:=w_{j,i}-q_jw_{j+1,i} \end{cases} \quad i=1,2,\cdots,\mu \qquad j=2,3,\cdots,m$$

第 4 步　用算法 2 求解参数 $x_{1,1},x_{m,1},x_{1,2},x_{m,2},\cdots,x_{1,\mu},x_{m,\mu}$。

第 5 步　求出剩余解：

$x_{j,1}:=w_{j,1}-x_{m,\mu}s_j-x_{1,2}t_j \quad j=2,3,\cdots,m-1$

$x_{j,i}:=w_{j,i}-x_{m,i-1}s_j-x_{1,i+1}t_j \quad i=2,3,\cdots,\mu-1; j=2,3,\cdots,m-1$

$x_{j,\mu}:=w_{j,\mu}-x_{m,\mu-1}s_j-x_{1,1}t_j \quad j=2,3,\cdots,m-1$

算法的内存开销源于向量 $\boldsymbol{p},\boldsymbol{q},\boldsymbol{s},\boldsymbol{t}\in\mathbf{R}^m, \boldsymbol{f}_i\in\boldsymbol{R}^m$ $(i=1,2,\cdots,\mu)$，以及参数方程中的 $\boldsymbol{d}\in\mathbf{R}^{2\mu}$，内存占用量约等于 $n+4m+2\mu$ 个实数单元。用文献[1]和[4]的算法求解拟三对角 Toeplitz 线性方程组时，需要的内存占用量约为 $4n=4\mu m$ 个实数单元。当 μ 和 n 足够大时，两种算法的内存占用量之比为

$$\frac{\mu m+4m+2\mu}{4\mu m}=\frac{1}{4}+\frac{1}{\mu}+\frac{1}{2m}\approx\frac{1}{4} \qquad (15)$$

也就是说，传统追赶算法的内存占用量几乎是新算法的 4 倍。

计算量统计：第 2 步，乘除 $2m$ 次，加减 m 次；第 3 步，乘除 $3n+4m$ 次，加减 $2n+2m$ 次；第 4 步，乘除 18μ 次，加减 12μ 次；第 6 步，乘除 $2n-4\mu$ 次，加减 $2n-4\mu$ 次，总共 $5n+6m+14\mu$ 次乘除法和 $4n+3m+8\mu$ 次加减法。用文献[1]和[4]的算法求解拟三对角 Toeplitz 线性方程组，运算量为 $9n=9\mu m$ 次乘除法和 $5n=5\mu m$ 次加减法。降阶法比追赶法减少的乘除法次数约为 $4n-6m-14\mu$，减少的加减法次数约为 $n-3m-8\mu$。

2.3　三对角 Toeplitz 线性方程组的分组降阶算法

算法 4(求解三对角 Toeplitz 线性方程组的分组降阶算法)

第 1 步　数据初始化：设 $n=\mu m$，向量 $\boldsymbol{f}_i\in\mathbf{R}^m(i=1,2,\cdots,\mu)$，且

$$f_{j,i}:=f_{im+j} \quad i=1,2,\cdots,\mu; j=1,2,\cdots,m$$

第 2 步　分解 $\boldsymbol{T}=\mathrm{diag}(a,b,c)=\mathrm{diag}(a,p_i,0)\cdot\mathrm{diag}(0,1,q_i)\in\mathbf{R}^{m\times m}, \boldsymbol{p},\boldsymbol{q}\in\mathbf{R}^m$：

$p_1:=b, q_1:=c/p_1; p_i:=b-aq_{i-1}, q_i:=c/p_i \quad i=2,3,\cdots,m-1$

$p_m:=b-aq_{m-1}$

第 3 步　追赶法求解 $\boldsymbol{Ts}=a\boldsymbol{e}_1, \boldsymbol{Tt}=c\boldsymbol{e}_m, \boldsymbol{Tw}_i=\boldsymbol{f}_i$ $(i=1,2,\cdots,\mu)$：

$s_1:=a/p_1, w_{1,i}:=w_{1,i}/p_1 \quad i=1,2,\cdots,\mu$

$$\begin{cases} s_j:=-as_{j-1}/p_j \\ w_{j,i}:=(f_{j,i}-aw_{j-1,i})/p_j \end{cases} \quad i=1,2,\cdots,\mu \qquad j=2,3,\cdots,m$$

$t_m:=c/p_m$

$$\begin{cases} s_j:=s_j-q_js_{j+1}, t_j:=t_j-q_jt_{j+1}, \\ w_{j,i}:=w_{j,i}-q_jw_{j+1,i} \end{cases} \quad i=1,2,\cdots,\mu \qquad j=2,3,\cdots,m$$

第 4 步　用算法 1 解出参数 $x_{m,1},x_{1,2},x_{m,2},x_{1,3},x_{m,3},\cdots x_{1,\mu-1},x_{m,\mu-1},x_{1,\mu}$。

第 5 步　给出剩余解:

$$x_{j,1}:=w_{j,1}-x_{1,2}t_j \qquad j=1,2,\cdots,m-1$$

$$x_{j,i}:=w_{j,i}-x_{m,i-1}s_j-x_{1,i+1}t_j \qquad i=2,3,\cdots,\mu-1;\ j=2,3,\cdots,m-1$$

$$x_{j,\mu}:=w_{j,\mu}-x_{m,\mu-1}s_j \qquad j=2,3,\cdots,m$$

算法的内存开销源于向量 $\boldsymbol{p},\boldsymbol{q},\boldsymbol{s},\boldsymbol{t}\in\mathbf{R}^m,\boldsymbol{f}_i\in\mathbf{R}^m$ $(i=1,2,\cdots,\mu)$,以及参数方程中的 $\boldsymbol{d}\in\mathbf{R}^{2(\mu-1)}$,内存占用量约等于 $(4+\mu)\cdot m+2\mu$ 个实数单元。传统追赶法需要内存空间的占用量约等于 $3n=3\mu m$ 个实数单元。当 μ 和 n 足够大时,两种算法的内存占用量之比为

$$\frac{1}{3}\left[1+\frac{4}{\mu}+\frac{2}{m}\right]\approx\frac{1}{3},\mu\in\mathbf{Z}^{+}\text{且}\,\mu\geqslant 2 \qquad (16)$$

也就是说,传统追赶法内存占用量几乎是降阶算法的 3 倍。

运算量统计:第 2 步需 $2m$ 次乘除法和 m 次加减法,第 3 步需 $(3n+4m-2\mu)$ 次乘除法和 $(2n+2m-2\mu)$ 次加减法,第 4 步需 10μ 次乘除法和 7μ 次加减法,第 6 步需 $(2n-2m-4\mu)$ 次乘除法和 $(2n-2m-4\mu)$ 次加减法,共 $(5n+4m+4\mu)$ 次乘除法和 $(4n+m+\mu)$ 次加减法。追赶法的运算量为 $5n=5\mu m$ 次乘除法和 $3n=3\mu m$ 次加减法。显然,对于标准三对角 Toeplitz 线性方程组的求解,降阶算法比追赶法多用 $(4m+4\mu)$ 次乘除运算和 $(n+m+\mu)$ 次加减运算。

3　数值实验

数值试验的平台是曙光天阔 W560I 工作站(CPU 是 Intel Xeron 2.33GHz,内存 4GB),操作系统是 Windows Server 2003,使用 Fortran 语言编程,数据类型为双精度浮点型。

例 1　在周期性边界条件下,用四阶紧致跳点差分格式[2](图 1)计算导函数 $f'(x)$,计算公式为

$$f'(x_{i-1/2})+22f'(x_{i+1/2})+f'(x_{i+3/2})=24\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} \qquad (17)$$

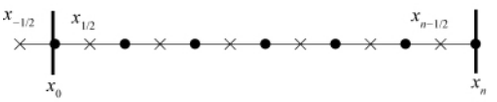

图 1　跳点网格示意图

Fig. 1　Schematic digram of the staggered grid

在计算中,已经给出了 $f(x)$ 在 n 个整数点(●)上的函数值,需要计算 $f'(x)$ 在 $n-1$ 个半点(×)上的函数值。在形成上式的系数矩阵时,需要边界外 $f'(x_{-1/2})$ 和 $f'(x_{n+1/2})$ 的值,而在周期性边界条件下有 $f'(x_{-1/2})=f'(x_{n-1/2})$ 和 $f'(x_{n+1/2})=f'(x_{1/2})$ 成立,从而在边界点 x_0 上的差分格式可以写成

$$f'(x_{n-1/2})+22f'(x_{1/2})+f'(x_{i/2})=24\frac{f(x_1)-f(x_0)}{x_1-x_0} \qquad (18)$$

在边界点 $x_{n-1/2}$ 上的差分格式可以写成

$$f'(x_{n-3/2})+22f'(x_{n-1/2})+f'(x_{1/2})=24\frac{f(x_n)-f(x_{n-1})}{x_n-x_{n-1}} \qquad (19)$$

结合式(17)—式(19),便可形成循环三对角方程组

$$\begin{bmatrix}22 & 1 & & & 1\\ 1 & 22 & 1 & & \\ \ddots & \ddots & \ddots & \ddots & \ddots\\ & & 1 & 22 & 1\\ 1 & & & 1 & 22\end{bmatrix}\begin{bmatrix}f'(x_{1/2})\\ f'(x_{3/2})\\ \vdots\\ f'(x_{n-3/2})\\ f'(x_{n-1/2})\end{bmatrix}=\begin{bmatrix}d_1\\ d_2\\ \vdots\\ d_{n-1}\\ d_n\end{bmatrix}$$

其中,$d_i=24\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}$,$i=0,1,\cdots,n-1$。

为了验证本文算法的计算效率,用本文算法和文献[1]的追赶法分别对算例 1 进行计算,结果如表 1 所示。

表 1　循环三对角方程组的求解时间

Table 1　Cost for solving the cyclic tri-diagonal system

方程的阶/10^6		5	10	15	20	50	65	100	200	260
降阶法	分组数	5	5	5	5	5	5	5	500	1000
	时间/s	0.15625	0.328125	0.46875	0.703125	1.6875	2.046875	3.437500	8.953125	11.09375*
追赶法[1]/s		0.328125	0.65625	1.00000	1.328125	3.343750	4.359375*	—	—	—

注:* 表示实验可达到的最大阶数,—表示内存溢出无法计算。

Notes: * means the maximum order in experiments; —means failure due to overflow.

考虑分组数对降阶法的影响,对同一方程选取不同的分组数进行计算,结果如表 2 所示。

实验结果表明:对于拟三对角 Toeplitz 线性方程组来说,无论是内存占用量还是运算速度,分组降阶算法的优势是非常明显的。从表 1 可以看出,两种算法能够求解的方程组的最大阶数几乎是 4 倍关系,这和算法的内存占用量的分析是完全相符合的。表 2 反映了分组降阶算法的另一个需要注意的问题,对于不同规模的方程组来说,分组个数对算法的效率影响是明显的,主要表现在如下两个方面。第一,对于特定规模的方程组来说,总存在一个最佳的分组个数。当分组个数较小或者较大时,都会影响算法的计算效率。但总的来看,分组个数较多时,本文的算法还是能够在一定程度上节约计算时间的。第二,随着方程组阶数的提高,最佳分组的个数也增大。计算机存取速度的分级设计和参数方程组的规模是影响最佳分组个数的主要原因。这是因为,计算机的存储器被分成若干等级,其中直接被 CPU 使用的一部分具有最小的容量和最快的速度,而普通的内存具有比它们大几个量级的容量和慢几个量级的速度。因此,随着分组个数的增加,一方

表 2　降阶法分组数 μ 对算法性能的影响

Table 2　Effects of the group numbers μ on the efficiency of the algorithm

n	μ	计算时间/s	n	μ	计算时间/s
$2^{10}\times10^4$	8	0.4375000	$2^{10}\times10^5$	8	4.515625
	16	0.4531250		16	4.484375
	32	0.4375000		32	4.390625
	64*	0.3125000		64	4.500000
	128	0.3281250		128*	3.312500
	256	0.3437500		256	3.343750
	512	0.3437500		512	4.281250
	1024	0.3906250		1024	4.031250

注:* 表示实验得到 μ 的最佳值。

Note: * represents the optimal value of μ obtained by experiments.

面,内存占有量减小,计算速度增加;另一方面,参数方程组的阶数增加导致计算速度减小。

4　结论

循环三对角 Toeplitz 线性方程组是工程计算中的一种常见问题。本文根据 Toeplitz 线性方程组的特点,运用并行算法中分而治之的思想,给出了一种求解循环三对角 Toeplitz 线性方程组的分组降阶串行算法。与求解同类问题的传统算法相比,分组降阶算法的优点在于它不仅大幅度减少了内存占用量,而且还大幅度减少了算术运算量。分组降阶算法分为 3 个步骤:第 1 步是分组降阶,即将一个 $n=\mu m$ 阶的方程组按行分成 μ 组,每组 m 个方程;第 2 步是构造参数方程组,即依据三对角系数矩阵的特点,给出各组解之间的关系式,把不属于该组的解分量看作参数。第 3 步是求解参数方程组和原方程组,依次先求解参数方程组,然后再代入相应分组的关系式便可求出所有的解分量。对于三对角 Toeplitz 线性方程组,同样能减少内存占用量,从而在计算机性能不变的情况下,提高求解问题的规模,但与求解三对角 Toeplitz 线性方程组的传统算法相比运算量有所增加。数值实验结果表明,对于特定规模的方程组来说,总存在一个最佳的分组个数使得计算时间最少;随着方程组阶数的提高,最佳分组的个数也增大。

参考文献(References)

[1] 王省富. 样条函数及其应用[M]. 西安: 西安交通大学出版社, 1989.
Wang Xingfu. Spline function and its application [M]. Xi'an: Xi'an Jiaotong University Press, 1989.

[2] 李青, 王能超. 解循环三对角线性方程组的追赶法 [J]. 小型微型计算机系统, 2002, 23(11): 1393-1395.
Li Qing, Wang Nengchao. *Journal of Chinese Computer Systems*, 2002, 23(11): 1393-1395.

[3] 王兴波, 钟志华. 求解周期性三对角方程组的广义 Thomas 算法[J]. 计算力学学报, 2004, 21(1): 73-76.
Wang Xingbo, Zhong Zhihua. *Chinese Journal of Computational Mechanics*, 2004, 21(1): 73-76.

[4] 李文强, 刘晓. 追赶法并行求解循环三对角方程组[J]. 科技导报, 2009, 27(18): 90-93.
Li Wenqiang, Liu Xiao. *Science and Technology Review*, 2009, 27(18): 90-93.

[5] 李文强, 马民. 求解循环三对角方程组的追赶法 [J]. 科技导报, 2009, 27(14): 69-72.
Li Wenqiang, Ma Min. *Science and Technology Review*, 2009, 27(14): 69-72.

[6] 崔喜宁, 吕全义. 周期三对角 Toeplitz 线性方程组的并行算法[J]. 昆明理工大学学报: 理工版, 2005, 30(5): 114-119.
Cui Xining, Lu Quanyi. *Journal of Kunming University of Science and Technology: Science and Technology Edition*, 2005, 30(5): 114-119.

(责任编辑　马宇红,代丽)