Pergamon

Computers Math. Applic. Vol. 29, No. 1, pp. 109–113, 1995
Copyright©1995 Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0898-1221/95 $7.00 + 0.00

0898-1221(94)00210-X

A Parallel Solver for Circulant Block-Tridiagonal Systems*

K.-L. CHUNG† AND Y.-H. TSAI
Department of Information Management
National Taiwan Institute of Technology
Taipei, Taiwan 10672, R.O.C.

W.-M. YAN
Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan 10764, R.O.C.

(Received May 1993; revised and accepted December 1993)

Abstract—Generalizing Müller and Scheerer's method which is used to parallelize the tridiagonal solvers, this paper presents a parallel method for solving the circulant block-tridiagonal systems. The applications of our result to solve the block-tridiagonal systems, the banded systems, and the circulant tridiagonal systems (for example, solving the closed B-spline curve fitting) are also addressed.

Keywords—Banded systems, Circulant block-tridiagonal systems, Closed B-spline curve fitting, Müller and Scheerer's method, Parallel algorithm.

1. INTRODUCTION

Solving the block-tridiagonal systems and the banded systems is very important in many applications [1]. Previously, extending the method of odd-even reduction [2], Rodrigue *et al.* [3] proposed a parallel banded solver; Kapur and Browne [4] proposed a parallel block tridiagonal solver on reconfigurable array computers; Utku, Salama, and Melosh [5] proposed a parallel solver for factorizing the Hermitian block tridiagonal systems. Based on the incomplete decompositions, van der Vorst [6] proposed parallel solvers for large tridiagonal and block-tridiagonal systems. In the Meier's paper [7], Wang's partition method [8] is generalized to the banded systems. Based on the iterative method and the preconditioner, Ruggiers and Galligani [9] proposed a parallel block-tridiagonal system solver.

Recently, Müller and Scheerer [10] proposed a partition method to parallelize the tridiagonal solvers. Linear systems having circulant coefficient matrices appear in many applications, for example, computer graphics [11,12]. How to generalize the method of Müller and Scheerer [10] to solve the circulant block-tridiagonal systems, the circulant tridiagonal systems, the block tridiagonal systems, and the banded systems is an interesting and important research problem.

Generalizing Müller and Scheerer's method, this paper presents a parallel method for solving the circulant block-tridiagonal systems. The applications of our result to solve the block-tridiagonal systems, the banded systems, and the circulant tridiagonal systems (for example, solving the closed B-spline curve fitting) are also addressed.

We wish to thank the referees for their valuable suggestions for improving the quality of this paper.
*This research was supported by the National Science Council of R.O.C. under Contract NSC82-0415-E011-180.
†Author to whom all correspondence should be addressed.

2. A PARALLEL SOLVER

This section presents a parallel solver for the circulant block-tridiagonal systems and the circulant tridiagonal systems. In addition, some applications of our parallel solver are also investigated.

Throughout this paper, matrices are represented by uppercase letters, vectors by bold lowercase letters, and scalars by plain lowercase letters. Following the same notations used in [9], given a circulant block-tridiagonal system $A\mathbf{x} = \mathbf{d}$, we let A be an $n \times n$ circulant block tridiagonal matrix

$$A = \begin{pmatrix} A_1 & B_1 & & & & C_1 \\ C_2 & A_2 & B_2 & & & \\ & C_3 & A_3 & B_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ B_n & & & & C_n & A_n \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}, \mathbf{d} = \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \vdots \\ \mathbf{d}_n \end{pmatrix},$$

where A_i, B_i, and C_i, $1 \le i \le n$, denote $q \times q$ submatrices; $\mathbf{x}_i$ and $\mathbf{d}_i$ denote the subvectors of $\mathbf{x}$ and $\mathbf{d}$, respectively.

Using the concept of the partition method in [10], naturally, we partition the previous circulant block-tridiagonal system to p subsystems for p processors, then each processor works on n/p consecutive equations (without loss of generality, we assume that $n/p = m$ is an integer). For $k = 1, \dots, p$ and $i = 1, \dots, n/p$, let $A_i^{(k)} = A_{(k-1)m+i}$, $B_i^{(k)} = B_{(k-1)m+i}$, $C_i^{(k)} = C_{(k-1)m+i}$, $\mathbf{x}_i^{(k)} = \mathbf{x}_{(k-1)m+i}$, and $\mathbf{d}_i^{(k)} = \mathbf{d}_{(k-1)m+i}$, and let

$$C^{(k)} = \begin{pmatrix} C_1^{(k)} \\ \\ \\ \\ \end{pmatrix}, \quad A^{(k)} = \begin{pmatrix} A_1^{(k)} & B_1^{(k)} & & & & \\ C_2^{(k)} & A_2^{(k)} & B_2^{(k)} & & & \\ & C_3^{(k)} & A_3^{(k)} & B_3^{(k)} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & & & & C_m^{(k)} & A_m^{(k)} \end{pmatrix}, \quad B^{(k)} = \begin{pmatrix} \\ \\ \\ \\ B_m^{(k)} \end{pmatrix},$$

$$\mathbf{x}^{(k)} = \begin{pmatrix} \mathbf{x}_1^{(k)} \\ \mathbf{x}_2^{(k)} \\ \vdots \\ \mathbf{x}_m^{(k)} \end{pmatrix} \text{ and } \mathbf{d}^k = \begin{pmatrix} \mathbf{d}_1^{(k)} \\ \mathbf{d}_2^{(k)} \\ \vdots \\ \mathbf{d}_m^{(k)} \end{pmatrix},$$

therefore, the vectors $\mathbf{x}$ and $\mathbf{d}$ can be rewritten by

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \vdots \\ \mathbf{x}^{(p)} \end{pmatrix} \text{ and } \mathbf{d} = \begin{pmatrix} \mathbf{d}^{(1)} \\ \mathbf{d}^{(2)} \\ \vdots \\ \mathbf{d}^{(p)} \end{pmatrix}.$$

According to the above partition manner and the definition of symbolic notations, for the first subsystem, we have

$$A^{(1)}\mathbf{x}^{(1)} + B^{(1)}\mathbf{x}_1^{(2)} + C^{(1)}\mathbf{x}_m^{(p)} = \mathbf{d}^{(1)}. \tag{1}$$

Similarly, for the last (the p^{th}) subsystem, we have

$$B^{(p)}\mathbf{x}_1^{(1)} + C^{(p)}\mathbf{x}_m^{(p-1)} + A^{(p)}\mathbf{x}^{(p)} = \mathbf{d}^{(p)}. \tag{2}$$

For the remaining $(p-2)$ subsystems, the k^{th}, $2 \le k \le p-1$, subsystem is of the form:

$$C^{(k)}\,\mathbf{x}_m^{(k-1)} + A^{(k)}\,\mathbf{x}^{(k)} + B^{(k)}\,\mathbf{x}_1^{(k+1)} = \mathbf{d}^{(k)}. \tag{3}$$

In order to unify the equations of (1)–(3), we let

$$\mathbf{x}^{(0)} = \mathbf{x}^{(p)} \quad \text{and} \quad \mathbf{x}^{(p+1)} = \mathbf{x}^{(1)},$$

then each subsystem is given by

$$A^{(k)}\,\mathbf{x}^{(k)} = \mathbf{d}^{(k)} - B^{(k)}\,\mathbf{x}_1^{(k+1)} - C^{(k)}\,\mathbf{x}_m^{(k-1)}, \qquad \text{for } 1 \le k \le p.$$

Here, we assume that each $A^{(k)}$ is nonsingular. Let

$$A^{(k)}\,\mathbf{u}^{(k)} = \mathbf{d}^{(k)}, \qquad A^{(k)}\,Y^{(k)} = B^{(k)}, \quad \text{and} \quad A^{(k)}\,Z^{(k)} = C^{(k)}, \tag{4}$$

suppose one processor is assigned to solve (4) and each processor sequentially solves its own three subsystems of (4) independently and simultaneously by using the Gaussian elimination. We have

$$\mathbf{x}^{(k)} = \mathbf{u}^{(k)} - Y^{(k)}\,\mathbf{x}_1^{(k+1)} - Z^{(k)}\,\mathbf{x}_m^{(k-1)}, \tag{5}$$

since

$$\begin{aligned} A^{(k)}\left\{\mathbf{u}^{(k)} - Y^{(k)}\,\mathbf{x}_1^{(k+1)} - Z^{(k)}\,\mathbf{x}_m^{(k-1)}\right\} &= \mathbf{d}^{(k)} - B^{(k)}\,\mathbf{x}_1^{(k+1)} - C^{(k)}\,\mathbf{x}_m^{(k-1)} \\ &= A^{(k)}\,\mathbf{x}^{(k)}. \end{aligned}$$

Before obtaining $\mathbf{x}^{(k)}$ for $1 \le k \le p$, the two parametric vectors, $\mathbf{x}_1^{(k+1)}$ and $\mathbf{x}_m^{(k-1)}$, on the right hand side of (5) shared by adjacent subsystems must be determined first. We define r_k to be the number of entries of vector $\mathbf{x}_1^{(k)}$; s_k to be the number of entries of vector $\mathbf{x}_m^{(k)}$. Let $\overline{Y}^{(k)}$ be the first r_k row of $Y^{(k)}$, and $\overline{Z}^{(k)}$ be the first r_k row of $Z^{(k)}$, in addition, $\underline{Y}^{(k)}$ and $\underline{Z}^{(k)}$ be the last s_k row of $Y^{(k)}$ and $Z^{(k)}$, respectively. By (5), we have the following parametric system:

$$\begin{aligned} \mathbf{x}_1^{(k)} &= \mathbf{u}_1^{(k)} - \overline{Y}^{(k)}\,\mathbf{x}_1^{(k+1)} - \overline{Z}^{(k)}\,\mathbf{x}_m^{(k-1)}; \\ \mathbf{x}_m^{(k)} &= \mathbf{u}_m^{(k)} - \underline{Y}^{(k)}\,\mathbf{x}_1^{k+1} - \underline{Z}^{(k)}\,\mathbf{x}_m^{(k-1)}. \end{aligned}$$

It is clear that the above parametric system has two shared parametric vectors, namely $\mathbf{x}_1^{(k+1)}$ and $\mathbf{x}_m^{(k-1)}$. After reordering each parametric system in order to align these two parametric vectors, respectively, we have

$$\begin{aligned} \mathbf{x}_1^{(k)} + \overline{Z}^{(k)}\,\mathbf{x}_m^{(k-1)} + \overline{Y}^{(k)}\,\mathbf{x}_1^{(k+1)} &= \mathbf{u}_1^{(k)}; \\ \underline{Z}^{(k)}\,\mathbf{x}_m^{(k-1)} + \underline{Y}^{(k)}\,\mathbf{x}_1^{(k+1)} + \mathbf{x}_m^{(k)} &= \mathbf{u}_m^{(k)}. \end{aligned}$$

In general, for $1 \le k \le p$, we have the remaining smaller parametric system, called the reduced parametric system, to be solved.

$$\begin{aligned} \mathbf{x}_1^{(1)} + \overline{Z}^{(1)}\,\mathbf{x}_m^{(0)} + \overline{Y}^{(1)}\,\mathbf{x}_1^{(2)} &= \mathbf{u}_1^{(1)}; \\ \underline{Z}^{(1)}\,\mathbf{x}_m^{(0)} + \underline{Y}^{(1)}\,\mathbf{x}_1^{(2)} + \mathbf{x}_m^{(1)} &= \mathbf{u}_m^{(1)}; \\ \mathbf{x}_1^{(2)} + \overline{Z}^{2}\,\mathbf{x}_m^{(1)} + \overline{Y}^{(2)}\,\mathbf{x}_1^{(3)} &= \mathbf{u}_1^{(2)}; \\ \underline{Z}^{(2)}\,\mathbf{x}_m^{(1)} + \underline{Y}^{(2)}\,\mathbf{x}_1^{(3)} + \mathbf{x}_m^{(2)} &= \mathbf{u}_m^{(2)}; \\ &\vdots \\ \mathbf{x}_1^{(p)} + \overline{Z}^{(p)}\,\mathbf{x}_m^{(p-1)} + \overline{Y}^{(p)}\,\mathbf{x}_1^{(p+1)} &= \mathbf{u}_1^{(p)}; \\ \underline{Z}^{(p)}\,\mathbf{x}_m^{(p-1)} + \underline{Y}^{(p)}\,\mathbf{x}_1^{(p+1)} + \mathbf{x}_m^{(p)} &= \mathbf{u}_m^{(p)}. \end{aligned}$$

Equivalently, we have

$$\begin{pmatrix} \overline{Z}^{(1)} & \overline{Y}^{(1)} & & & & & & I \\ \underline{Z}^{(1)} & \underline{Y}^{(1)} & I & & & & & \\ & I & \overline{Z}^{2} & \overline{Y}^{(2)} & & & & \\ & & \underline{Z}^{(2)} & \underline{Y}^{(2)} & I & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & & I & \overline{Z}^{(p)} & \overline{Y}^{(p)} \\ I & & & & & & \underline{Z}^{(p)} & \underline{Y}^{(p)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_m^{(p)} \\ \mathbf{x}_1^{(2)} \\ \mathbf{x}_m^{(1)} \\ \mathbf{x}_1^{(3)} \\ \vdots \\ \mathbf{x}_1^{(p)} \\ \mathbf{x}_m^{(p-1)} \\ \mathbf{x}_1^{(1)} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_1^{(1)} \\ \mathbf{u}_m^{(1)} \\ \mathbf{u}_1^{(2)} \\ \mathbf{u}_m^{(2)} \\ \vdots \\ \mathbf{u}_m^{(p-1)} \\ \mathbf{u}_1^{(p)} \\ \mathbf{u}_m^{(p)} \end{pmatrix}. \tag{6}$$

It is observed that the structure of the above reduced parametric system preserves the same structure of the original system, a circulant block-tridiagonal structure.

Previously, some authors [13] also examined the reduced system in parallel, based on the assumption: $p \ll$ the order of A. Following the same assumption and approach, we now collect the coefficients of all parametric vectors into one specific processor. Once the specific processor solves the reduced system, the solutions of $\mathbf{x}_1^{(k)}$ and $\mathbf{x}_m^{(k)}$ are sent to the k^{th} processor for $1 \le k \le p$. Then each processor performs the substitutions to obtain the final results.

We next consider the case of the block-tridiagonal system, that is, $C_1 = O$ and $B_n = O$ in (1), by (5), derives that $Z^{(1)} = O$ and $Y^{(p)} = O$. Finally, the reduced system becomes a block-tridiagonal system.

$$\begin{pmatrix} \underline{Y}^{(1)} & I & & & & \\ I & \overline{Z}^{2} & \overline{Y}^{(2)} & & & \\ & \underline{Z}^{(2)} & \underline{Y}^{(2)} & I & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & & & & I & \overline{Z}^{(p)} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1^{(2)} \\ \mathbf{x}_m^{(1)} \\ \mathbf{x}_1^{(3)} \\ \vdots \\ \mathbf{x}_1^{(p)} \\ \mathbf{x}_m^{(p-1)} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_m^{(1)} \\ \mathbf{u}_1^{(2)} \\ \mathbf{u}_m^{(2)} \\ \vdots \\ \mathbf{u}_m^{(p-1)} \\ \mathbf{u}_1^{(p)} \end{pmatrix}. \tag{7}$$

Since a banded system is a special case of the circulant block-tridiagonal systems, it can be solved in parallel by using our parallel scheme for solving the block tridiagonal system.

In what follows, we concentrate on solving the closed curve fitting problem as the representation of the circulant tridiagonal class [11]. Closed curve fitting is important in computer-aided design, graphics, pattern recognition, and picture processing [14]. The task of closed curve fitting is to construct a smooth closed curve that fits a given set of n points in the space. B-spline curve interpolation is a good fitting tool because a local shape change of the curve affects only its vicinity. Some sequential methods for efficient curve interpolation in $O(n)$ time using B-splines have been proposed in [14].

In the following, we describe how to transform the closed B-spline curve fitting problem into the solution of a circulant linear system. Suppose we are given a set of points, $\mathbf{b}_i = (x_i, y_i, z_i)$ for $1 \le i \le n$. According to [14], the corner points of a fourth-order closed B-spline curve, which is a piecewise cubic curve, can be expressed by a weighted average of three control points:

$$\begin{aligned} \mathbf{b}_1 &= \frac{1}{6}(\mathbf{c}_n + 4\mathbf{c}_1 + \mathbf{c}_2), \\ \mathbf{b}_n &= \frac{1}{6}(\mathbf{c}_{n-1} + 4\mathbf{c}_n + \mathbf{c}_1), \quad \text{and} \\ \mathbf{b}_i &= \frac{1}{6}(\mathbf{c}_{i-1} + 4\mathbf{c}_i + \mathbf{c}_{i+1}), \qquad \text{for } 2 \le i \le n-1. \end{aligned}$$

They form a system of n equations in n unknowns for each coordinate. The above system of equations can be equivalently transformed into

$$\begin{pmatrix} 4 & 1 & & & 1 \\ 1 & 4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 4 & 1 \\ 1 & & & 1 & 4 \end{pmatrix} \begin{pmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_{n-1} \\ \mathbf{c}_n \end{pmatrix} = 6 \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_{n-1} \\ \mathbf{b}_n \end{pmatrix}. \tag{8}$$

After solving the circulant linear system of (8), we can obtain the control points, $\mathbf{c}_i$'s, of the n defining polygon vertices.

By the previous arguments in solving the circulant block-tridiagonal systems, (8) can be solved in parallel by using our parallel scheme since (8) is also a special case of the circulant block-tridiagonal systems.

3. CONCLUSIONS

We have presented a parallel method to solve the circulant block-tridiagonal systems in a machine-independent manner. In addition, our parallel scheme can be applied to solve the other three special classes of linear systems, namely the block-tridiagonal systems, the banded systems, and the circulant tridiagonal systems. On a real parallel organization such as hypercube networks, how to balance each work load in each processor to achieve the best performance in our method is the future research topic.

REFERENCES

1. G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins Press, London, (1989).
2. D. Heller, Some aspects of the cyclic reduction algorithm for block tridiagonal linear systems, *SIAM J. Numer. Anal.* **13**, 484–496 (1976).
3. G.H. Rodrigue, N.K. Madsen, and J.I. Karush, Odd-even reduction for banded linear equations, *J. ACM* **26**, 72–81 (1979).
4. R.N. Kapur and J.C. Browne, Techniques for solving block tridiagonal systems on reconfigurable array computers, *SIAM J. Sci. Stat. Comput.* **5**, 701–719 (1984).
5. S. Utku, M. Salama, and R.J. Melosh, A family of permutations for concurrent factorization of block tridiagonal matrices, *IEEE Trans. on Computers* **C–38**, 812–824 (1989).
6. H.A. van der Vorst, Large tridiagonal and block tridiagonal linear systems on vector and parallel computers, *Parallel Computing* **5**, 45–54 (1987).
7. U. Meier, A parallel partition method for solving banded systems of linear equations, *Parallel Computing* **2**, 33–43 (1985).
8. H.H. Wang, A parallel method for tridiagonal equations, *ACM Trans. on Math. Software* **7**, 170–183 (1981).
9. V. Ruggiero and E. Galligani, A parallel algorithm for solving block tridiagonal linear systems, *Computers Math. Applic.* **24** (4), 15–21 (1992).
10. S.M. Müller and D. Scheerer, A method to parallelize tridiagonal solvers, *Parallel Computing* **17**, 181–188 (1991).
11. R.F. Boisvert, Algorithms for special tridiagonal systems, *SIAM J. Sci. Stat. Comput.* **12**, 423–442 (1991).
12. O. Rojo, A new method for solving symmetric circulant tridiagonal systems of linear equations, *Computers Math. Applic.* **20** (12), 61–67 (1990).
13. P. Amodio, L. Brugnano, and T. Politi, Parallel factorizations for tridiagonal matrices, *SIAM J. Numer. Anal.* **30**, 813–823 (1993).
14. R.H. Bartels, J.C. Beatty, and B.A. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Pub., San Mateo, CA, (1987).