International Journal of Computer Research
Volume 10, Number 1, pp. 89-97

A Scalable Parallel Algorithm for Periodic Symmetric Toeplitz Tridiagonal Systems

XIAN-HE SUN*
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
sun@cs.iit.edu
Department of Computer Science
Louisiana State University
Baton Rouge, LA 70803-4020
sun@csc.lsu.edu

Abstract: Symmetric Toeplitz tridiagonal systems arise in many scientific applications. An efficient algorithm, the Simple Parallel Prefix (SPP) algorithm, was previously proposed for solving symmetric Toeplitz tridiagonal systems on SIMD and vector computers. Based on the SPP algorithm, a scalable parallel algorithm is proposed for solving periodic symmetric Toeplitz tridiagonal systems in this study. The newly proposed algorithm has the same parallel computation count as that of the SPP algorithm for non-periodic systems, and it requires only shift communication.

1 INTRODUCTION

A matrix is Toeplitz if its entries along each diagonal are the same. Symmetric Toeplitz tridiagonal systems arise in many scientific applications. They appear in multigrid methods, Alternating Direction Implicit (ADI) method, wavelet collocation method, and in line-SOR preconditioners for conjugate gradient methods [3]. In addition to solving PDE's, Toeplitz tridiagonal systems also arise in digital signal processing, image processing, stationary time series analysis, as well as in spline curve fitting [7].

Because of its importance, intensive research has been done on the development of efficient parallel tridiagonal solvers. Many algorithms have been proposed [6], including the well known recursive doubling reduction method (RCD) developed by Stone [2], and

* This research was supported in part by the National Science Fundation under NSF grant ASC-9720215 and CCR-9972251, and by National Aeronautics and Space Administration under NASA contract NAS1-1672.
# On leave from LSU

the cyclic reduction or odd-even reduction method (OER) developed by Hockney [1]. In general, parallel Toeplitz tridiagonal solvers require global communications which makes them inefficient on distributed-memory architectures. Recently, we have taken a new approach [3,4]: increasing parallel performance by introducing bounded numerical error. Two new algorithms, namely the Parallel Diagonal Dominant (PDD) algorithm [3] and the Simple Parallel Prefix (SPP) algorithm [4], have been proposed based on this approach for MIMD and SIMD machines respectively. These two algorithms take the advantage of that most of the tridiagonal systems arising in scientific applications are diagonal dominant. Backed by rigorous accuracy analysis, the algorithms truncate communication and computation without degrading the accuracy. Theoretical and experimental results have shown that these two algorithms are efficient, practical and applicable to many scientific applications.

In this paper, a new algorithm is introduced for solving periodic systems. The newly proposed algorithm has the same parallel computation count as that of the SPP algorithm. It requires only shift communications. Since the new algorithm is an extension of the SPP algorithm, it is well believed that the accuracy results regarding the SPP algorithm can be applied to the new algorithm as well. The discussion of the SPP algorithm is given in Section 2. The new algorithm for periodic symmetric Toeplitz tridiagonal systems is introduced in Section 3. The final section gives the conclusion.

2 The Simple Parallel Prefix Algorithm

The Simple Parallel Prefix (SPP) algorithm [4] is a "tearing" algorithm. It first solves a modified system, and, then, it corrects the intermedian result to get the final solution. The SPP algorithm is designed for fine-grain computing. With n processors, it solves an n-dimensional tridiagonal system with $2\lceil \lg(n) \rceil + 1$ AXPY operations. Two prefix communications are required in the computing (tearing) phase and one broadcast communication is needed in the correction phase. When the tridiagonal system is diagonally dominant, both the computing and correction phases can be truncated without influencing the accuracy. The truncation makes the SPP algorithm superior to other existing methods, even on vector machines.

2.1 The SPP Algorithm

We are interested in solving a linear system

$$Ax = d, \tag{1}$$

where A is a symmetric Toeplitz tridiagonal matrix of order n:

$$A = \begin{bmatrix} c & 1 & & & \\ 1 & c & 1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & 1 \\ & & & 1 & c \end{bmatrix} = [1, c, 1] \tag{2}$$

and $x = (x_1, \cdots, x_n)^T$ and $d = (d_1, \cdots, d_n)^T$ are n-dimensional vectors. We assume that matrix A is diagonally dominant (i.e., $|c| > 2$).

Computing Phase

The SPP consists of two phases, the computing phase and the correction phase. In the computing phase, the system

$$\tilde{A}\tilde{x} = d \tag{3}$$

is solved, where

$$\tilde{A} = \begin{pmatrix} a & 1 & & \\ 1 & c & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & c \end{pmatrix} = a \begin{pmatrix} 1 & & & \\ b & 1 & & \\ & \ddots & \ddots & \\ & & b & 1 \end{pmatrix} \begin{pmatrix} 1 & b & & \\ & \ddots & \ddots & \\ & & \ddots & b \\ & & & 1 \end{pmatrix} = a \cdot [b,1,0] \cdot [0,1,b]$$

a and b are the real solutions of:

$$\begin{cases} a + b = c \\ a \cdot b = 1 \end{cases} \tag{4}$$

Because $a \cdot b = 1$ and $|c| > 2$, we may further assume that $|a| > 1$ and $|b| < 1$. From Eq. (3), $\tilde{x} = a^{-1} \cdot [0,1,b]^{-1} [b,1,0]^{-1} d = b \cdot [0,1,b]^{-1} [b,1,0]^{-1} d$. Let $L = [-b,0,0]$. Then

$$[b,1,0] = [0,1,0] - [-b,0,0] = I - L$$

and

$$[0,1,b]^{-1} = (I + L^{2^{\lceil \lg n \rceil - 1}}) \cdots\cdots (I + L^2)(I + L). \tag{5}$$

The superscripts of matrix L represent matrix multiplication. Similarly, let $U = [0,0,-b]$. Then,

$$[b,1,0]=[0,1,0]-[-b,0,0]=I-U$$

and

$$[0,1,b]^{-1}=(I+U^{2^{\lceil \lg n\rceil -1}})\cdots\cdots(I+U^2)(I+U).$$

Thus, the solution of Eq. (3) is

$$\tilde{x}=b\cdot(I+U^{2^{\lceil \lg n\rceil -1}})\cdots\cdots(I+U^2)(I+U)(I+L^{2^{\lceil \lg n\rceil -1}})\cdots\cdots(I+L^2)(I+L)d. \quad (6)$$

Let $v=(v_1,\cdots,v_n)^T$ be an n-dimensional vector. Given the special structure of L^i, we find $(I+L^i)v=v+(-b)^i v_{(i)}$, where $v_{(i)}=(0,\cdots 0,v_1,\cdots,v_{n-i})^T$ and v_1 is the $i+1$ element of $v_{(i)}$. Similarly, for U^i, we find $(I+U^i)v=v+(-b)^i v^{(i)}$, where $v^{(i)}=(v_{i+1},\cdots,v_n,0,\cdots 0)^T$. The operation that a vector plus a multiple of another vector is called the AXPY operation. It can be implemented efficiently on SIMD and vector machines. Equation (6) indicates that Eq. (3) can be solved in $2\cdot\lceil \lg n\rceil$ AXPY operations. Because $|b|<1$, and both $\|L^i\|\to 0, \|U^i\|\to 0$, when $n\to\infty$, the AXPY operations can be truncated without influencing the accuracy.

Correction Phase

Our goal is to find the solution of Eq. (1). Modification is needed to convert the solution of Eq. (3) to the final solution. The relation between Eq. (1) and Eq. (3) is given by

$$x=\tilde{x}-\tilde{A}^{-1}V(I+E^T\tilde{A}^{-1}V)^{-1}\tilde{x}_1,$$

where $\tilde{x}_1$ is the first element of vector $\tilde{x}$, and

$$\tilde{A}^{-1}V(I+E^T\tilde{A}^{-1}V)^{-1}=b^2\left(\frac{1-b^{2n}}{1-b^{2(n+1)}},\cdots,(-b)^i\frac{1-b^{2(n-i)}}{1-b^{2(n+1)}},\cdots,(-b)^{n-1}\frac{1-b^2}{1-b^{2(n+1)}}\right)^T. \quad (7)$$

The final solution is

$$x=\tilde{x}-\tilde{x}_1 z, \quad (8)$$

where vector z is the right side of Eq. (7). Because $|b|<1$, z can be truncated at some integer k_1 without affecting the accuracy. Furthermore, when n is large, $b^{2(n-i+1)}, i=0,1,\cdots,k_1$, will be less than machine accuracy, and z is reduced to

$\tilde{z} = ((-b)^2, (-b)^3, \cdots, (-b)^{k_1+2}, 0, \cdots, 0)^T$. Totally, without truncation, $2 \cdot \lceil \lg n \rceil + 1$ AXPY operations are needed to get the final solution.

2.2 Accuracy Analysis

If the Toeplitz matrix under consideration is diagonal dominant, the $2 \cdot \lceil \lg n \rceil$ AXPY operations in the computing phase can be truncated to *2k* AXPY operations without influencing the accuracy, where *k* is an integer less than $\lceil \lg n \rceil$. Let *x* be the solution of the SPP algorithm without truncation and x^* be the corresponding solution with truncation, the accuracy analysis in [4] shows that the relative error under l_1 norm is bounded by

$$\frac{\|x-x^*\|}{\|x\|} \le \frac{|b|^k (1+|b|^{n-k})}{1-|b|} \left(1+\frac{(1-|b|^k)(1+|b|)}{1-|b|}\right)\left(1+\frac{b^2(1+|b|^{n+1})(1-|b|^n)}{(1-b^2)(1-|b|)}\right)$$

When the order of the tridiagonal system *n* is large, $|b|^n$ may be smaller than the machine accuracy. In this case, the inequality (9) becomes

$$\frac{\|x-x^*\|}{\|x\|} \le \frac{|b|^k (1+|b|^{n-k})}{1-|b|} \left(1+\frac{(1-|b|^k)(1+|b|)}{1-|b|}\right)\left(1+\frac{b^2}{(1-b^2)(1-|b|)}\right)$$

In general the integer *k* is quite small. For instance, if the diagonal dominancy, $|c|$, is larger than or equal to *3*, *k* equals *4* and *6* for single precision (10^{-7}) and double precision (10^{-14}) respectively.

3 The New Algorithm

When the boundary conditions are periodic, the discretization matrices of PDE's and other scientific applications are periodic systems. For periodic systems, the matrix *A* in the equation

$$Ax = d, \tag{11}$$

has the form

$$A=\begin{pmatrix} c & 1 & & 1 \\ 1 & c & 1 & \\ & \ddots & \ddots & 1 \\ 1 & & 1 & c \end{pmatrix} = a\begin{pmatrix} 1 & & & b \\ b & 1 & & \\ & \ddots & \ddots & \\ & & b & 1 \end{pmatrix}\begin{pmatrix} 1 & b & & \\ & \ddots & \ddots & \\ & & \ddots & b \\ b & & & 1 \end{pmatrix}, \quad (12)$$

where a and b are the solutions of the linear system (4). Let

$$L_P=\begin{pmatrix} 0 & & & -b \\ -b & \ddots & & \\ & \ddots & \ddots & \\ & & -b & 0 \end{pmatrix}.$$

Then,

$$I-L_P=\begin{pmatrix} 1 & & & b \\ b & 1 & & \\ & \ddots & \ddots & \\ & & b & 1 \end{pmatrix}$$

It is easy to verify that the inverse of $I-L_P$ is

$$(I-L_P)^{-1}=\frac{1}{1-(-b)^n}(I+L_P+L_P^2+\cdots+L_P^{n-1}). \quad (13)$$

The i-th power of L_P has the form

$$L_P^i=\begin{pmatrix} & & (-b)^i & & \\ (-b)^i & & & \ddots & \\ & \ddots & & & (-b)^i \\ & & (-b)^i & & \end{pmatrix},$$

where the first non-zero element in first column is the $(i+1, 1)$-th entry, and the first non-zero element in first row is the $(1, n-i+1)$-th entry. The inverse of matrix $I-L_P$ can be computed through $2^{\lceil \lg(n) \rceil}-1$ matrix mulplications using the following equality:

$$(I+L_P+L_P^2+\cdots+L_P^{n-1})=(I+L^{2^{\lceil \lg n \rceil -1}})(I+L^{2^{\lceil \lg n \rceil -2}})\cdots(I+L) \quad (14)$$

Similarly, let

$$U_P = \begin{pmatrix} 0 & -b & & \\ & \ddots & \ddots & \\ & & \ddots & -b \\ -b & & & 0 \end{pmatrix}.$$

Then

$$(I - U_P)^{-1} = \frac{1}{1-(-b)^n}(I + U_P + U_P^2 + \cdots + U_P^{n-1}) \tag{15}$$

$$= \frac{1}{1-(-b)^n}(I + U_P^{2^{\lceil \lg n \rceil - 1}})(I + U_P^{2^{\lceil \lg n \rceil - 2}}) \cdots (I + U_P) \tag{16}$$

Based on Eq. (12), (14) and Eq. (16), the solution of (11) can be obtained by

$$x = A^{-1}d = a^{-1}(I - U_P)^{-1}(I - L_P)^{-1}d$$
$$= \frac{b}{(1-(-b)^n)^2}(I + U_P^{2^{\lceil \lg n \rceil - 1}}) \cdots (I + U_P)(I + L_P^{2^{\lceil \lg n \rceil - 1}}) \cdots (I + L_P)d$$

It requires $2 \cdot \lceil \lg n \rceil + 1$ matrix-vector mulplications. In general, matrix-vector mulplication needs n^2 operations. Since the matrices L_P^i and U_P^i are special Toeplitz matrices, the required matrix-vector multiplications can be done with AXPY operations. For an n-dimensional vector, $v = (v_1, \cdots, v_n)^T$,

$$(I + L^i)v = v + (-b)^i v_{(i)}, \tag{17}$$

with $v_{(i)} = (v_{n-i+1}, v_{n-i+2} \cdots, v_n, v_1, \cdots, v_{n-i})^T$ and

$$(I + U_P^i)v = v + (-b)^i v^{(i)},$$

with $v^{(i)} = (v_{i+1}, v_{i+2} \cdots, v_n, v_1, v_2, \cdots, v_i)^T$. Notice that for matrices L_P^i and U_P^i, the second vector in the AXPY operation is a shift of the first vector along opposite directions. The shift of vector requires "shift" communications on distributed-memory machines. Figure 1 gives the proposed algorithm for solving periodic symmetric Toeplitz tridiagonal systems. With n processors the algorithm finds the solution in $2 \cdot \lceil \lg n \rceil + 1$ AXPY operations for a matrix of order n, Since $|b| < 1$, we may assume that $|b|^n$ is less than machine accuracy and assume that the AXPY operation can be truncated from $2 \cdot \lceil \lg n \rceil + 1$ to $2k + 1$ without degrading the accuracy. The integer k is independent of the order of matrix. Interested readers may refer [4] for detailed accuracy study. Using n processors, each of the three j loops in Figure 1 can be executed in one AXPY operation

for $i \leftarrow 0$ ***to*** $k - 1$ ***do***
    for $j \leftarrow 1$ ***to*** n ***do***
        $d_j = d_j + (-b)^{2^i} d_{(|j-2^i|)\bmod(n)}$
        $j \leftarrow j + 1$
    $i \leftarrow i + 1$

for $i \leftarrow 0$ ***to*** $k - 1$ ***do***
    for $j \leftarrow 1$ ***to*** n ***do***
        $d_j = d_j + (-b)^{2^i} d_{(j+2^i)\bmod(n)}$
        $j \leftarrow j + 1$
    $i \leftarrow i + 1$

for $j \leftarrow 1$ ***to*** n ***do***
    $x_j = b \cdot d_j$
    $j \leftarrow j + 1$

Figure 1. The Proposed Algorithm for Periodic Systems.

4 CONCLUSION

A scalable parallel algorithm has been proposed for solving periodic symmetric Toeplitz Tridiagonal systems for fine-grain computing. With n processors, the proposed algorithm needs $2 \cdot \lceil \lg n \rceil + 1$ AXPY operations for a system of n equations. Most tridiagonal systems arising in scientific computing are diagonal dominant. The proposed algorithm takes the advantage of diagonal dominancy and truncate the AXPY operations from $2 \cdot \lceil \lg n \rceil + 1$ to $2k + 1$ without influencing the accuracy. The proposed algorithm is an extension of the SPP algorithm [4]. Based on the accuracy analysis and experimental results given in [4], k is quite small. In general, k is equal to *4* and *6* for single and double precision respectively. The truncation of computation and communication makes the proposed algorithm efficient on vector machines, where AXPY operations can be implemented efficiently, as well as on SIMD machines. Since the integer k is independent of the order of matrix, the truncation also makes the algorithm scalable. It is perfectly scalable in terms of *isospeed* scalability [5], that is when problem size increases linearly with the number of processors, the achieved average speed of the algorithm will maintain the same.

REFERENCES

[1] Hockney, R. A fast direct solution of Poisson's equation using Fourier analysis. *J. of ACM* 12 (1965), 95-113.

[2] Stone, H. An efficient parallel algorithm for the solution of a tridiagonal linear system of equations. *J. of ACM* 20, 1 (Jan. 1973), 27-38.

[3] Sun, X.-H. Application and accuracy of the parallel diagonal dominant algorithm. *Parallel Computing* (Aug. 1995), 1241-1267.

[4] Sun, X.-H., and Joslin, R. A simple parallel prefix algorithm for almost Toeplitz tridiagonal systems. *International Journal of High Speed Computing* (Dec. 1995).

[5] Sun, X.-H., and Rover, D. Scalability of parallel algorithm-machine combinations. *IEEE Transactions on Parallel and Distributed Systems* (June 1994), 599-613.

[6] Sun, X.-H., Zhang, H., and Ni, L. Efficient tridiagonal solvers on multicomputers. *IEEE Transactions on Computers* 41, 3 (1992), 286-296.

[7] Taha, T. R., and Jiang, P. A parallel algorithm for solving periodic tridiagonal Toeplitz linear systems. In *Proc. of the Sixth SIAM Conf. on Parallel Processing for Scientific Computing* (March 1993), 491-496.