②

NASA Contractor Report 191431

ICASE Report No. 93-6

ICASE

20 Years of Excellence

APPLICATIONS AND ACCURACY OF THE PARALLEL DIAGONAL DOMINANT ALGORITHM

Xian-He Sun

NASA Contract No. NAS1-19480
February 1993

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association

28 pgs

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Applications and Accuracy
of
the Parallel Diagonal Dominant Algorithm *

Xian-He Sun

ICASE
Mail Stop 132C
NASA Langley Research Center
Hampton, VA 23681-0001
(804) 864-8018, sun@icase.edu

ABSTRACT

The Parallel Diagonal Dominant (PDD) algorithm is a highly efficient, ideally scalable tridiagonal solver. In this paper, a detailed study of the PDD algorithm is given. First the PDD algorithm is introduced. Then the algorithm is extended to solve periodic tridiagonal systems. A variant, the reduced PDD algorithm, is also proposed. Accuracy analysis is provided for a class of tridiagonal systems, the symmetric and anti-symmetric Toeplitz tridiagonal systems. Implementation results show that the analysis gives a good bound on the relative error, and the algorithm is a good candidate for the emerging massively parallel machines.

---

*This research was supported by the National Aeronautics and Space Administration under NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Application in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

Solving tridiagonal systems is one of the key issues in computational fluid dynamics (CFD) and many other scientific applications [21, 13]. Many methods used for the solution of partial differential equations (PDEs) rely on solving a sequence of tridiagonal systems. The alternating direction implicit (ADI) method, the most widely used implicit method for PDEs [17], solves PDEs by solving tridiagonal systems alternately in each coordinate direction. Discretization of partial differential equations by compact difference schemes also leads to a sequence of tridiagonal systems. Tridiagonal systems also arise in multigrid methods and in ADI or line-SOR preconditioners for conjugate gradient methods. In addition to solving PDE's, tridiagonal systems also arise in many other applications [1].

Solving tridiagonal systems is inexpensive on sequential machines. However, because of their serial nature, tridiagonal systems are difficult to solve efficiently on parallel computers. Thus intensive research has been done on the development of efficient parallel tridiagonal solvers. Many algorithms have been proposed [14, 8]. Among them, the recursive doubling reduction method (RCD), developed by Stone [16], and the cyclic reduction or odd-even reduction method (OER), developed by Hockney [9], are able to solve an n-dimensional tridiagonal system in O(log n) time using n processors. These are effective algorithms for fine grained computing. Later, several algorithms were proposed for median and coarse grain computing, i.e. for the case of $p < n$ or $p << n$, where p is the number of processors available [5, 11, 22]. The algorithm given by Lawrie and Sameh [11] and the algorithm given by Wang [22] can be considered substructured methods. These algorithms partition the original problem into sub-problems. The sub-problems are solved in parallel, and then the solutions of the sub-problems are combined to obtain the final solution. All of these parallel tridiagonal solvers increase parallelism by adding additional computation. They trade increased work for reduced communication overhead and better load balance and have a larger operation count than the best sequential algorithm.

Recently, Sun, Zhang, and Ni [21] have proposed three parallel algorithms for solving tridiagonal systems. All of these algorithms are based on Sherman-Morrison matrix modification formula [3]. The parallel partition LU (PPT) algorithm and the parallel hybrid (PPH) algorithm are fast and able to incorporate limited pivoting. The PPT algorithm is a good candidate when the number of processors, p, is small. The PPH algorithm is a better choice when p is large. Finally, for diagonal dominant problems, the (PPD) algorithm is the most efficient.

Compared with other tridiagonal solvers, which all have at least O(log p) communication cost, the PDD algorithm has only a small fixed communication cost and a small amount of additional computation. In fact, the PDD algorithm is perfectly scalable, in the the sense that the communication cost and the computation overhead do not increase with the problem size or with the number of processors available.

Modern technological advances have made it possible to build computers containing more and more processors. Multiprocessors with hundreds or thousands of processors are commercially available. Recent parallel computers, such as the Intel Paragon, Thinking Machine Corporations's CM-5, and Cray's MPP, highlight the use of high-density and high-speed processor and memory chips based on ultra-large-scale integration (ULSI) or very high-speed integrated circuits (VHSICs). With this new technology, 64-bit 150-MHz microprocessors, for example, are now available on a single chip having 1.6-million transistors [10]. The emerging parallel computers build on such microprocessors are noted for their scalable architecture and massively parallel processing. They are designed for grand challenge applications which could not otherwise be tackled.

Scalability has become an important metric of parallel algorithms [6, 20, 19]. Its perfect scalability and high efficiency make the PDD algorithm, when applicable, an ideal choice on these new machines. However, the PDD algorithm is relatively new and applicable only under certain conditions. In this paper we give a detailed study of the PDD algorithm. We study the applications of the PDD algorithm and provide a formal accuracy analysis for Toeplitz tridiagonal systems. The PDD algorithm proposed in this paper is slightly different from the algorithm proposed in [21]. Extended study is provided for different applications, such as periodic systems, and systems with multiple right-sides. The reduced PDD algorithm is also proposed. Simple formulas are provided for accuracy checking for symmetric and anti-symmetric Toeplitz tridiagonal systems.

This paper is organized as follows. Section 2 will introduce the sequential and parallel PDD algorithms. The applications of the PDD algorithm will be discussed in Section 3. This section will also give the variant of the PDD algorithm for periodic systems and the reduced PDD algorithm. Section 4 will give the accuracy study for the PDD algorithm and the reduced PDD algorithm. Experimental results on an Intel/860 multicomputer will be presented in Section 5. Finally, Section 6 gives the conclusion and final remarks.

2 The Parallel Diagonal Dominant Algorithm

We are interested in solving a tridiagonal linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{d} \tag{1}$$

where $\mathbf{A}$ is a tridiagonal matrix of order n

$$\mathbf{A} = \begin{bmatrix} b_0 & c_0 & & & & \\ a_1 & b_1 & c_1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & a_{n-2} & b_{n-2} & c_{n-2} \\ & & & & a_{n-1} & b_{n-1} \end{bmatrix} = [a_i, b_i, c_i] \tag{2}$$

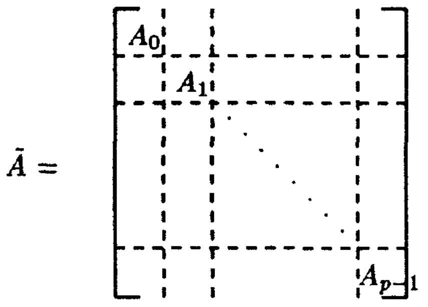
$\mathbf{x} = (x_0, \cdots, x_{n-1})^T$ and $\mathbf{d} = (d_0, \cdots, d_{n-1})^T$. We assume that $\mathbf{A}, \mathbf{x}$, and $\mathbf{d}$ have real coefficients. Extension to the complex case is straightforward.

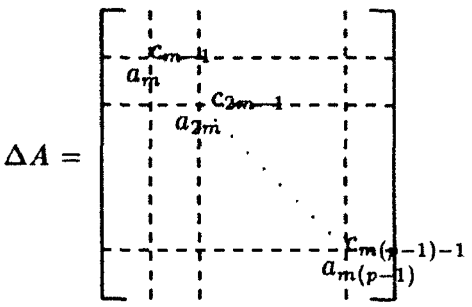
2.1 The Matrix Partition Technique

To solve Eq. (1) efficiently on parallel computers, we partition $\mathbf{A}$ into submatrices. For convenience we assume that $n = pm$. The matrix $\mathbf{A}$ in Eq. (2) can be written as

$$\mathbf{A} = \tilde{\mathbf{A}} + \Delta\mathbf{A}, \tag{3}$$

where





The submatrices $\mathbf{A}_i (i = 0, \cdots, p-1)$ are $m \times m$ tridiagonal matrices. Let e_i be a column vector with its ith $(0 \le i \le n-1)$ element being one and all the other entries being zero. We have

$$\Delta\mathbf{A} = [a_m e_m, c_{m-1}, e_{m-1} a_{2m} e_{2m}, c_{2m-1} e_{2m-1}, \cdots, c_{(p-1)m-1} e_{(p-1)m-1}] \begin{bmatrix} e_{m-1}^T \\ e_m^T \\ \cdot \\ \cdot \\ e_{(p-1)m-1}^T \\ e_{(p-1)m}^T \end{bmatrix} = \mathbf{V}\mathbf{E}^T \tag{4}$$

where both $\mathbf{V}$ and $\mathbf{E}$ are $n \times 2(p-1)$ matrices. Thus, we have

$$\mathbf{A} = \tilde{\mathbf{A}} + \mathbf{V}\mathbf{E}^T. \tag{5}$$

Based on the matrix modification formula originally defined by Sherman and Morrison [15] for rank-one changes and generalized by Woodbury [23], and assuming that all A_i's are invertible, Eq. (1) can be solved by

$$x = \mathbf{A}^{-1}\mathbf{d} = (\tilde{\mathbf{A}} + \mathbf{V}\mathbf{E}^{\mathrm{T}})^{-1}\mathbf{d} \tag{6}$$

$$x = \tilde{\mathbf{A}}^{-1}\mathbf{d} - \tilde{\mathbf{A}}^{-1}\mathbf{V}(\mathbf{I} + \mathbf{E}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\mathbf{V})^{-1}\mathbf{E}^{\mathrm{T}}\tilde{\mathbf{A}}^{-1}\mathbf{d}. \tag{7}$$

Note that I is an identity matrix and $\mathbf{Z} = \mathbf{I} + \mathbf{E}^{\mathrm{T}} \tilde{\mathbf{A}}^{-1} \mathbf{V}$ is a pentadiagonal matrix of order $2(p-1)$. Let

$$\begin{aligned} \tilde{A}\tilde{x} &= d && (8)\\ \tilde{A}Y &= V && (9)\\ h &= E^T \tilde{x} && (10)\\ Z &= I + E^T Y && (11)\\ Zy &= h && (12)\\ \Delta x &= Yy. && (13) \end{aligned}$$

Equation (7) becomes

$$x = \tilde{x} - \Delta x. \tag{14}$$

In Eq.s (8) and (9), $\tilde{x}$ and Y are solved by the LU decomposition method. By the structure of $\tilde{A}$ and V, this is equivalent to solve

$$A_i[\tilde{x}^{(i)}, v^{(i)}, w^{(i)}] = [d^{(i)}, a_{im} e_0, c_{(i+1)m-1} e_{m-1}], \tag{15}$$

$i = 0, \cdots, p-1$. Here $\tilde{x}^{(i)}$ and $d^{(i)}$ are the ith block of $\tilde{x}$ and d, respectively, and $v^{(i)}$, $w^{(i)}$ are possible nonzero column vectors of the ith row block of Y. Equation (15) implies that we only need to solve three linear systems of order m with the same LU decomposition for each i ($i = 0, \cdots, p-1$). In addition, we can skip the first $m-1$ forward substitutions for the third system since the first $m-1$ components of the vector at the right-hand side are all zeros. There is no computation or communication involved in computing h and Z.

2.2 The PDD Algorithm

Solving Eq. (12) is the major computation involved in the conquer part of our algorithms. Different approaches have been proposed for solving Eq.(12), which results different algorithms for solving tridiagonal systems [21]. The matrix Z in Eq. (12) has the form

$$\mathbf{Z} = \begin{bmatrix} 1 & w_{m-1}^{(0)} & 0 & & & & \\ v_0^1 & 1 & 0 & w_0^{(1)} & & & \\ v_{m-1}^{(1)} & 0 & 1 & w_{m-1}^{(1)} & & & \\ & \cdot & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \cdot & \\ & & & \cdot & 1 & 0 & w_0^{(p-2)} \\ & & & & \cdot & 1 & w_{m-1}^{(p-2)} \\ & & & & & v_0^{(p-1)} & 1 \end{bmatrix} \tag{16}$$

where $v^{(i)}, w^{(i)}$ for $i = 0, \cdots, p-1$ are solutions of Eq. (15) and the 1's come from the identity matrix I. In practice, especially for a diagonal dominant tridiagonal system, the magnitude of the last component of $v^{(i)}$, $v_{m-1}^{(i)}$, and the first component of $w^{(i)}$, $w_0^{(i)}$, may be smaller than machine accuracy when $p << n$. In this case, $w_0^{(i)}$ and $v_{m-1}^{(i)}$ can be dropped, and Z becomes a diagonal block system consisting of $(p-1)$ 2×2 independent blocks. Thus, Eq.(12) can be solved efficiently on parallel computers, which leads to the highly efficient *parallel diagonal dominant* (PDD) algorithm.

In the sequential PDD algorithm, since Y has at most two nonzero entries in every row, and Z is a diagonal block matrix with 1's as diagonal elements, (12) takes five arithmetic operations per row, and the evaluation of (13) takes four operations per row. Based on the above observations, and together with a careful scaling process, we conclude that the sequential PDD algorithm takes $17n - 9\frac{n}{p} - 4p - 9$ arithmetic operations.

Using p processors, the PDD algorithm consists of the following steps:

Step 1. Allocate $A_i, d^{(i)}$, and elements $a_{im}, c_{(i+1)m-1}$ to the ith node, where $0 \leq i \leq p-1$.

Step 2. Solve (15). All computations can be executed in parallel on p processors.

Step 3. Send $\tilde{x}_0^{(i)}, v_0^{(i)}$ from the ith node to the $(i-1)$th node, for $i = 1, \cdots, p-1$.

Step 4. Solve

$$\begin{bmatrix} 1 & w_{m-1}^{(i)} \\ v_0^{(i+1)} & 1 \end{bmatrix} \begin{pmatrix} y_{2i} \\ y_{2i+1} \end{pmatrix} = \begin{pmatrix} \tilde{x}_{m-1}^{(i)} \\ \tilde{x}_0^{(i+1)} \end{pmatrix} \tag{17}$$

in parallel on the ith node for $0 \leq i \leq p-2$. Then send y_{2i} from the ith node to the $(i+1)$th node, for $i = 0, \cdots, p-2$.

Step 5. Compute (13) and (14). We have

$$\Delta x^{(i)} = [v^{(i)}, w^{(i)}] \begin{pmatrix} y_{2(i-1)} \\ y_{2i} \end{pmatrix} \tag{18}$$

$$x^{(i)} = \tilde{x}^{(i)} - \Delta x^{(i)} \tag{19}$$

In all of these, one has only two neighboring communications.

Communication cost is an overhead of parallelism. Recent advanced communication mechanisms, such as circuit switching and wormhole routing, have reduced communication delay considerably. However, compared with the improvement of processing speed, the improvement of communication speed is relatively small. Communication cost has a great impact on overall performance. Empirically, for most distributed-memory computers, the communication time for a neighboring communication is a linear function of the problem size [4]. Let S be the number of bytes to be transferred. Then the transfer time of a neighboring communication can be expressed as $\alpha + S\beta$, where α represents a fixed startup overhead and β is the incremental transmission time per byte. Assuming 4 bytes are used for each real number, Step 3 and Step 4 take $\alpha + 8\beta$ and $\alpha + 4\beta$

communication respectively. The parallel PDD algorithm needs $17\frac{n}{p} - 4$ parallel computation and $2(\alpha + 6\beta)$ communication.

2.3 Scalability Analysis

As parallel machines have been built with more and more processors, the performance metric *scalability* becomes more and more important. Thus, the question is how an algorithm will perform when the problem size is scaled up linearly with the number of processors. Let $T(p, W)$ be the execution time for solving a system with W work (problem size) on p processors. The ideal situation would be when both the number of processors and the amount of work are scaled up N times, the execution time remains unchanged:

$$T(N \times p, N \times W) = T(p, W) \tag{20}$$

How one should define problem size, in general, is a style under debate. However, it is commonly agreed that the floating point (flop) operation count is a good estimate of problem size for scientific computations. To eliminate the effect of numerical inefficiencies in parallel algorithms, in practice the flop count is based upon some practical optimal sequential algorithm. In our case, the LU decomposition has chosen as the sequential algorithm. It takes $8n - 7$ floating point operations, where 7 is a negligible constant number when n is large. As the problem size W increases N times to W', we have

$$\begin{aligned} W' &= N \times 8n = 8n' \\ n' &= N \cdot n. \end{aligned} \tag{21}$$

Let τ_{comp} represent the unit of a computation operation normalized to the communication time. The time required to solve (1) by the PDD algorithm with p processors is

$$T(p, W) = (17\frac{n}{p} - 4)\tau_{comp} + 2(\alpha + 6\beta), \tag{22}$$

and

$$\begin{aligned} T(N \times p, N \times W) &= (17\tfrac{n'}{N \cdot p} - 4)\tau_{comp} + 2(\alpha + 6\beta) \\ &= (17\tfrac{N \cdot n}{N \cdot p} - 4)\tau_{comp} + 2(\alpha + 6\beta) \\ &= (17\tfrac{n}{p} - 4)\tau_{comp} + 2(\alpha + 6\beta) \\ &= T(p, W). \end{aligned} \tag{23}$$

The PDD algorithm has the ideal scalability. Similar arguments could be applied to periodic systems (see Section 3) and the same result would be obtained.

Using the isospeed approach, scalability has been formally defined in [20]. The average unit speed is defined as the quotient of the achieved speed of the given computing system and the number of processors. Since Eq.(20) is true if and only if the average unit speed of the given computing system is a constant, the scalability is defined as the ability to maintain the average unit speed [20]. Let W be the amount of work of an algorithm when p processors are employed in a machine, and let W' be the amount of work of the algorithm when $p' = N \cdot p$ processors are employed to maintain the average speed, then the scalability from system size p to system size $N \cdot p$ of the algorithm-machine combination is defined as

$$\psi(p, N \times p) = \frac{N \cdot p \cdot W}{p \cdot W'} = \frac{N \cdot W}{W'}. \tag{24}$$

The average unit speed can be represented as

$$A\_S(p, W) = \frac{W}{p \cdot T(p, W)}, \tag{25}$$

where W is the problem size, p is the number of processors, and $T(p, W)$ is the corresponding execution time. From our early discussion, for the PDD algorithm, when $W' = N \cdot W$, we have $T(N \times p, W') = T(p, W)$. Therefore

$$A\_S(N \times p, W') = \frac{W'}{N \cdot T(N \times p, W')} = \frac{W'}{N \cdot T(p, W)} = \frac{N \cdot W}{N \cdot T(p, W)} = \frac{W}{T(p, W)}. \tag{26}$$

That is $W' = N \cdot W$ has maintained the average unit speed, and the scalability is

$$\psi(p, N \times p) = \frac{N \cdot W}{W'} = \frac{N \cdot W}{N \cdot W} = 1. \tag{27}$$

It is the ideal scalability.

3 Special Applications

In this section, we first discuss some tridiagonal systems arising in CFD applications, the *symmetric* and *anti-symmetric Toeplitz tridiagonal systems.* Then two variants of the PDD algorithm, the *reduced PDD algorithm* and the PDD algorithm for periodic systems, will be presented.

3.1 Toeplitz Tridiagonal Systems

A Toeplitz tridiagonal matrix has the form

$$\mathbf{A} = \begin{bmatrix} b & c & & & \\ a & b & c & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & c \\ & & & a & b \end{bmatrix} = [a, b, c]. \tag{28}$$

Symmetric Toeplitz tridiagonal systems are often arise in solving partial differential equations and in other scientific applications. Compact finite difference scheme [12] is a relative new scheme for solving PDE's. Because of its simplicity and high accuracy, it has been widely used in practice. Using the compact scheme, the general approximation of a first derivative has the form:

$$\beta f'_{i-2} + \alpha f'_{i-1} + f'_i + \alpha f'_{i+1} + \beta f'_{i+2} = c\frac{f_{i+3} - f_{i-3}}{6h} + b\frac{f_{i+2} - f_{i-2}}{4h} + a\frac{f_{i+1} - f_{i-1}}{2h}$$

Letting

$$\alpha = \frac{1}{3}, \beta = 0, a = \frac{14}{9}, b = \frac{1}{9}, c = 0, \tag{29}$$

the scheme becomes formally sixth order accurate and the resulting system is $[\frac{1}{3}, 1, \frac{1}{3}]$, a symmetric Toeplitz tridiagonal system. Similarly, the general approximation of a second derivative has the form

$$\beta f''_{i-2} + \alpha f''_{i-1} + f''_i + \alpha f''_{i+1} + \beta f''_{i+2} = c\frac{f_{i+3} - 2f_i + f_{i-3}}{9h^2} + b\frac{f_{i+2} - 2f_i + f_{i-2}}{4h^2} + a\frac{f_{i+1} - 2f_i + f_{i-1}}{h^2}.$$

For

$$\alpha = \frac{2}{11}, \beta = 0, a = \frac{12}{11}, b = \frac{3}{11}, c = 0, \tag{30}$$

a sixth order difference scheme is obtained, and the tridiagonal system is symmetric and Toeplitz, $[\frac{2}{11}, 1, \frac{2}{11}]$. Discretized in time, the one dimensional wave equation $u_t = a \cdot u_x$ and the heat equation $u_t = a \cdot u_{xx}$ can be represented as

$$u^{n+1} = u^n + \Delta t \cdot a \cdot u^n_x, \tag{31}$$

and

$$u^{n+1} = u^n + \Delta t \cdot a \cdot u^n_{xx} \tag{32}$$

respectively.

Using the compact scheme, u^n_x and u^n_{xx} are defined by symmetric Toeplitz tridiagonal systems. Therefore, the solutions can be obtained by solving a sequence of symmetric Toeplitz tridiagonal systems. Using ADI methods [17], parabolic and hyperbolic systems can be solved by solving a sequence of symmetric Toeplitz tridiagonal systems.

Anti-symmetric Toeplitz tridiagonal systems also arise in solving PDEs [17]. For instance, to solve the wave equation $u_t + a \cdot u_x = f$, we begin with the formula

$$u_t = \frac{u(t+k, x) - u(t.x)}{k} + O(k^2) \tag{33}$$

for u_t evaluated at $(t + \frac{1}{2}k, x)$. We also use the relation

$$\begin{aligned} u_x(t + \tfrac{1}{2}k, x) &= \tfrac{u_x(t+k,x)+u_x(t,x)}{2} + O(k^2) \\ &= \tfrac{1}{2}\left[\tfrac{u(t+k,x+h)-u(t+k,x-h)}{2h} + \tfrac{u(t,x+h)-u(t,x-h)}{2h}\right] + O(k^2) + O(h^2). \end{aligned} \tag{34}$$

Using these approximations for $u_t + a \cdot u_x = f$ about $(t + \frac{1}{2}k, x)$, we obtain

$$\frac{v_m^{n+1} - v_m^n}{k} + a\frac{v_{m+1}^{n+1} - v_{m-1}^{n+1} + v_{m+1}^n - v_{m-1}^n}{4h} = \frac{f_m^{n+1} + f_m^n}{2} \tag{35}$$

or, equivalently,

$$\frac{a\lambda}{4}v_{m+1}^{n+1} + v_m^{n+1} - \frac{a\lambda}{4}v_{m-1}^{n+1} = -\frac{a\lambda}{4}v_{m+1}^n + v_m^n + \frac{a\lambda}{4}v_{m-1}^n + \frac{k}{2}(f_m^{n+1} + f_m^n). \tag{36}$$

The left side is an *anti-symmetric Toeplitz tridiagonal matrix*, $A = [\frac{a\lambda}{4}, 1, \frac{-a\lambda}{4}]$.

3.2 Periodic Tridiagonal Systems

Many PDE's arisen in real applications have periodic boundary conditions. For instance, to study a physical phenomenon of a large object, we often simulate only a small portion of it and then apply periodic boundary conditions on each of the portions. The resulting linear systems have the form of

$$\mathbf{A} = \left[\begin{array}{ccccccc} b_0 & c_0 & & & & & a_0 \\ a_1 & b_1 & c_1 & & & & \\ \cdot & \cdot & \cdot & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & \cdot & \cdot & \cdot & & \\ & & & a_{n-2} & b_{n-2} & c_{n-2} & \\ c_{n-1} & & & & a_{n-1} & b_{n-1} & \end{array}\right], \tag{37}$$

and are called ***periodic tridiagonal systems***. On sequential machine, periodic tridiagonal systems are solved by combining the solutions of two different right-sides [7], which increases the operation count from $8n - 7$ to $14n - 16$.

The PDD algorithm can be extended to periodic tridiagonal systems. The difference is that, after dropping $w_0^{(i)}$, and $v_{m-1}^{(1)}$, the matrix Z becomes a periodic system of order $2p$:

$$\mathbf{Z} = \left[\begin{array}{cccccc} 1 & w_{(m-1)}^{(0)} & & & & v_0^{(0)} \\ v_0^{(1)} & 1 & 0 & & & \\ & 0 & 1 & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & w_{m-1}^{(p-2)} \\ w_{m-1}^{(p-1)} & & & & v_0^{(p-1)} & 1 \end{array}\right] \tag{38}$$

The dimension of **Z** is slightly higher than in the non-periodic case, which simply makes the load on the 0th and (p-1)th processor identical to load on all of the other processors. The parallel computation time remains the same. For periodic systems, the communication at step 3 and 4 changes from one dimensional array communication to ring communication. The communication time is also unchanged. Figure 1 shows the communication pattern of the PDD algorithm for periodic systems.

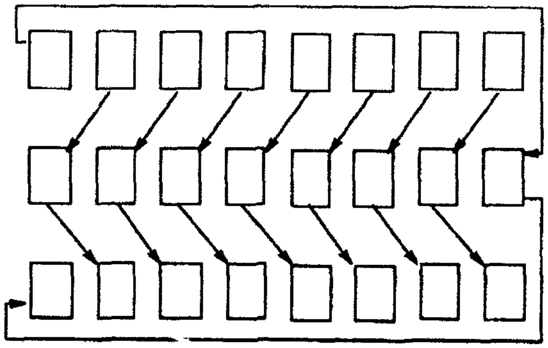


Figure 1. Communication Pattern for Solving Periodic Systems.

3.3 The Reduced PDD Algorithm

In the last step, Step 5, of the PDD algorithm, the final solution, x, is computed by combining the intermediate results concurrently on each processor,

$$x^{(i)} = \tilde{x}^{(i)} - y_{2(i-1)}v^{(i)} - y_{2i}w^{(i)}, \tag{39}$$

which requires $4(n-1)$ operations in total and $4m$ parallel operations, if $p = n/m$ processors are used. The PDD algorithm drops off the the first element of w, w_0, and the last element of v, v_{m-1}, in solving Eq. (12). In Section 4.1 - 4.2, we will show that, for symmetric and anti-symmetric Toeplitz tridiagonal systems, the w_0 and v_{m-1} can be dropped when m is large with the accuracy of the final solution unaffected. Further more, we have

$$v = \frac{1}{\lambda(a + b\sum_{i=0}^{m-1} b^{2i})}\left(\sum_{i=0}^{m-1} b^{2i}, \sum_{i=0}^{m-1} b^{2i}/(-b), \cdots, (-b)^{m-1}\right)^T. \tag{40}$$

So, when m is large enough, we may drop off $v_i, i = \frac{m}{2}, \cdots, m-1$, and $w_i, i = 0, 1, \cdots, \frac{m}{2} - 1$, while maintaining the same accuracy. If we replace v_i by $\tilde{v}_i$, where $\tilde{v}_i = v_i$ for $i = 0, 1, \cdots, \frac{m}{2} - 1$, $\tilde{v}_i = 0$, for $i = \frac{m}{2}, \cdots, m-1$; and replace w by $\tilde{w}$, where $\tilde{w}_i = w_i$ for $i = \frac{m}{2}, \cdots, m-1$, and $\tilde{w}_i = 0$, for $i = 0, 1, \cdots, \frac{m}{2} - 1$; and use $\tilde{v}, \tilde{w}$ in step 5, we have

Step 5'

$$\Delta x^{(i)} = [\tilde{v}, \tilde{w}] \begin{pmatrix} y_{2(i-1)} \\ y_{2i} \end{pmatrix} \tag{41}$$

$$x^{(i)} = \tilde{x}^{(i)} - \Delta x^{(i)}. \tag{42}$$

It requires $2\frac{n}{p}$ parallel operation. Replacing Step 5 of the PDD algorithm by Step 5', we get the reduced PDD algorithm which requires $15\frac{p}{n} - 4$ parallel computations.

4 Accuracy Analysis

The PDD algorithm is highly efficient, perfectly scalable, but it is only applicable when the intermediate results $v_{m-1}^{(i)}, w_0^{(i)}, 0 \leq i \leq p-1$, can be dropped out. However this dropping may lead to inaccurate or even wrong solution. Thus an accuracy study is essential for applying the PDD algorithm. Some study have been done for the accuracy of the PDD algorithm. Sufficient conditions have been given [24, 2]. However, the study is for the general case. The conditions given in [24] are difficult to verify and the accuracy bound is large. In this section we focus on a particular class of tridiagonal systems, *symmetric and anti-symmetric Toeplitz tridiagonal systems*. Our analysis is four fold. First, we give the decay rate of $w_0^{(i)}, v_{m-1}^{(i)}, i = 0, \cdots, p-1$. They are the entries treated as zeros by the PDD algorithm. Second, the accuracy of the PDD algorithm is studied. Then, we analyze the accuracy of the reduced PDD algorithm. All of the above three analysis are for symmetric Toeplitz tridiagonal systems. Finally, we extend the results to anti-symmetric Toeplitz tridiagonal systems.

4.1 The Decay Rate of $\mathbf{v_{m-1}}$ and $\mathbf{w_0}$

Symmetric Toeplitz tridiagonal systems have the form $A = [\lambda, \beta, \lambda] = \lambda[1, c, 1]$, where $c = \beta/\lambda$. We assume the matrix A is diagonal dominant. That is $|c| > 2$. To study the accuracy of the solution of $Ax = b$, we first study the matrix

$$\tilde{\mathbf{B}} = \begin{pmatrix} a & 1 & & & \\ 1 & c & 1 & & \\ & 1 & \cdot & \cdot & \\ & & \cdot & \cdot & 1 \\ & & & 1 & c \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ b & 1 & & & \\ & b & \cdot & & \\ & & \cdot & \cdot & \\ & & & b & 1 \end{pmatrix} \begin{pmatrix} a & 1 & & & \\ & a & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & 1 \\ & & & & a \end{pmatrix}$$

where a and b are the real solutions of

$$b + a = c, \quad b \cdot a = 1. \tag{43}$$

Since $a \cdot b = 1$ and $|c| > 2$, we may further assume that $|a| > 1$, and $|b| < 1$.

The LDL^T decomposition of $\tilde{\mathbf{B}}$ is

$$\tilde{\mathbf{B}} = [b,1,0] \times [0,a,0] \times [0,1,b].$$

Thus

$$\tilde{\mathbf{B}}^{-1} = [0,1,b]^{-1} \times [0,a,0]^{-1} \times [b,1,0]^{-1}$$

$$= \begin{pmatrix} 1 & -b & b^2 & . & (-b)^{n-1} \\ & 1 & -b & . & (-b)^{n-2} \\ & & . & . & . \\ & & & 1 & -b \\ & & & & 1 \end{pmatrix} \begin{pmatrix} a^{-1} & & & & \\ & a^{-1} & & & \\ & & . & & \\ & & & . & \\ & & & & a^{-1} \end{pmatrix} \begin{pmatrix} 1 & & & & \\ -b & 1 & & & \\ b^2 & -b & 1 & & \\ . & . & . & . & \\ (-b)^{n-1} & . & . & -b & 1 \end{pmatrix}$$

Let $d = (1,0,\cdots,0)^T$, then

$$\tilde{\mathbf{B}}^{-1}d = \frac{1}{a}\left(\sum_{i=0}^{n-1} b^{2i}, \sum_{i=1}^{n-1} b^{2i}/(-b), \sum_{i=2}^{n-1} b^{2i}/b^2, \ldots, \sum_{i=n-1}^{n-1} b^{2i}/(-b)^{n-1}\right)^T$$

Let

$$\mathbf{\Delta B} = \begin{pmatrix} b & 0 & . & . & . & 0 \\ 0 & 0 & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & . & . & . & 0 \end{pmatrix} = \begin{pmatrix} b \\ 0 \\ . \\ . \\ 0 \end{pmatrix} (1,0,\cdots,0) = \tilde{V}\tilde{E}^T, \tag{44}$$

and

$$\mathbf{B} = \tilde{\mathbf{B}} + \mathbf{\Delta B} = [1,c,1] \tag{45}$$

Then, by the matrix modification formula (7), the solution of $By = d$ is

$$\begin{aligned} y &= B^{-1}d = (\tilde{B} + \tilde{V}\tilde{E}^T)^{-1}d \\ &= \tilde{B}^{-1}d - \tilde{B}^{-1}\tilde{V}(I + \tilde{E}^T\tilde{B}^{-1}\tilde{V})^{-1}\tilde{E}^T\tilde{B}^{-1}d \end{aligned} \tag{46}$$

where

$$(I + \tilde{E}^T\tilde{B}^{-1}\tilde{V})^{-1} = \frac{a}{a + b\sum_{i=0}^{n-1} b^{2i}},$$

$$\tilde{E}^T\tilde{B}^{-1}d = \frac{\sum_{i=0}^{n-1} b^{2i}}{a},$$

$$\tilde{B}^{-1}\tilde{V} = \frac{b}{a}\left(\sum_{i=0}^{n-1} b^{2i}, \sum_{i=1}^{n-1} b^{2i}/(-b), \cdots, \sum_{i=n-1}^{n-1} b^{2i}/(-b)^{n-1}\right)^T,$$

$$\tilde{B}^{-1}\tilde{V}(I+\tilde{E}^T\tilde{B}^{-1}\tilde{V})^{-1}\tilde{E}^T\tilde{B}^{-1}d = \frac{b}{a}\cdot\frac{\sum_{i=0}^{n-1} b^{2i}}{a+b\sum_{i=0}^{n-1} b^{2i}}\begin{pmatrix}\sum_{i=0}^{n-1} b^{2i}\\ \cdot\\ \cdot\\ (-b)^{n-1}\end{pmatrix}$$

The last element of y is

$$\begin{aligned} y_{n-1} &= \frac{(-b)^{n-1}}{a} - \frac{(-b)^{n-1}}{a}\cdot\frac{b\sum_{i=0}^{n-1} b^{2i}}{a+b\sum_{i=0}^{n-1} b^{2i}} = \frac{(-b)^{n-1}}{a}\left(\frac{a}{a+b\sum_{i=0}^{n-1} b^{2i}}\right) && (47)\\ &= \frac{(-b)^{n-1}}{a}\left(\frac{1}{1+b^2\sum_{i=0}^{n-1} b^{2i}}\right) \qquad (\textit{note } a\cdot b = 1). && (48)\end{aligned}$$

Thus:

$$|y_{n-1}| \le \frac{|b|^{n-1}}{|a|} = |b|^n \tag{49}$$

The first element of y is

$$y_0 = \frac{\sum_{i=0}^{n-1} b^{2i}}{a}\left(\frac{1}{1+b^2\sum_{i=0}^{n-1} b^{2i}}\right) = \frac{b(1-b^{2n})}{1-b^{2(n+1)}}$$

$$|y_0| = \left|\frac{b(1-b^{2n})}{1-b^{2(n+1)}}\right| < |b|. \tag{50}$$

For the original system $Ax = d, A = \lambda[1, c, 1]$, the first element of x is

$$x_0 = \frac{y_0}{\lambda}. \tag{51}$$

The last element of x is

$$x_{n-1} = \frac{y_{n-1}}{\lambda} \tag{52}$$

Since for Toeplitz tridiagonal systems, each submatrix $A_i, i = 0, \cdots, p-1$, has the same structure as A, we have the following lemma:

Lemma 1 *If* $\frac{b^{m-1}}{\lambda a}$*, where* $m = n/p$*, is less than machine accuracy, then* $v_{m-1}^{(i)}, i = 0, \cdots, p-1$*, can be replaced by zero without affecting the accuracy of the final solution of* $Ax = d$*.*

With similar arguments, we can prove that for $d = (0, \cdots 0, 1)^T$, $Ax = d$ has solution

$$x_i = \frac{y_{n-(i+1)}}{\lambda}. \tag{53}$$

In particular

$$\begin{aligned} x_{n-1} &= \tfrac{y_0}{\lambda} \\ x_0 &= \tfrac{y_{n-1}}{\lambda} \end{aligned}$$

Combining with Lemma 1, we have:

Theorem 1 *If $\frac{b^{m-1}}{\lambda a}$, $m = n/p$, is less than machine accuracy, then the PDD algorithm gives an approximation to the true solution within machine accuracy.*

4.2 Accuracy of the PDD Algorithm

Theorem 1 says that if v_{m-1}, w_0 are less than machine accuracy, the PDD algorithm gives a satisfactory solution. In most scientific applications, the accuracy requirement is much weaker than machine accuracy. We now study how the decay rate of v_{m-1}, w_0 influences the accuracy of the final solution. Our study starts at the matrix partition formula (7).

Let

$$y = (I + E^T \tilde{A}^{-1} V)^{-1} E^T \tilde{A}^{-1} d. \tag{54}$$

Substitute y into equation (7), we have

$$\begin{aligned} x &= \tilde{A}^{-1} d - \tilde{A}^{-1} V y \\ E^T x &= E^T \tilde{A}^{-1} d - E^T \tilde{A}^{-1} V \cdot y \\ &= (I + E^T \tilde{A}^{-1} V) y - E^T \tilde{A}^{-1} V \cdot y = y. \end{aligned} \tag{55}$$

Let y^* be the corresponding solution of the PDD algorithm,

$$y^* = (I + E^T \tilde{A}^{-1} V - D)^{-1} E^T \tilde{A}^{-1} d,$$

where D is the $2(p-1) \times 2(p-1)$ matrix which contains all the $v_{m-1}^{(i)}, w_0^{(i)}$ elements. Combined with Eq.(54) we have

$$(I + E^T \tilde{A}^{-1} V) y - (I + E^T \tilde{A}^{-1} V - D) y^* = 0,$$

That is

$$(y^* - y) = (I + E^T \tilde{A}^{-1} V - D)^{-1} D \cdot y.$$

Let x^* be the corresponding final solution of the PDD algorithm. Then

$$\begin{aligned} x^* &= \tilde{A}^{-1} d - \tilde{A}^{-1} V \cdot y^* \\ x - x^* &= \tilde{A}^{-1} V (y^* - y) \\ &= \tilde{A}^{-1} V (I + E^T \tilde{A}^{-1} V - D)^{-1} D \cdot y \\ &= \tilde{A}^{-1} V (I + E^T \tilde{A}^{-1} V - D)^{-1} D \cdot E^T x. \end{aligned}$$

Thus,

$$\frac{||x - x^*||}{||x||} \le ||\tilde{A}^{-1}V(I + E^T\tilde{A}^{-1}V - D)^{-1}DE^T||. \tag{56}$$

The inequality (56) holds for general tridiagonal systems. In the following we assume the special structure of symmetric Toeplitz tridiagonal system to compute the norm of its right side. We use the 1-norm in our study. As discussed in the last section,

$$(I + E^T\tilde{A}^{-1}V - D)^{-1} = \begin{pmatrix} Z_0^{-1} & & & & \\ & Z_1^{-1} & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & Z_{p-1}^{-1} \end{pmatrix}, \tag{57}$$

where Z_i are 2×2 matrices:

$$Z_i = \begin{pmatrix} 1 & w_{m-1}^{(i)} \\ v_0^{(i)} & 1 \end{pmatrix} \tag{58}$$

For symmetric Toeplitz tridiagonal systems $v_0^{(i)} = w_{m-1}^{(i)} = v_0^{(0)} = \tilde{a}$, and $v_{m-1}^{(i)} = w_0^{(i)} = v_{m-1}^{(0)} = \tilde{b}$, for $i = 0, \cdots, p-1$. So, for our applications,

$$Z_i = Z_1 = \begin{pmatrix} 1 & \tilde{a} \\ \tilde{a} & 1 \end{pmatrix}, \tag{59}$$

$$Z_1^{-1} = \frac{1}{1 - \tilde{a}^2}\begin{pmatrix} 1 & -\tilde{a} \\ -\tilde{a} & 1 \end{pmatrix}. \tag{60}$$

$D \cdot E^T$ stretchs D from a $2(p-1) \times 2(p-1)$ matrix to a $2(p-1) \times n$ matrix. Each column of $D \cdot E^T$ is either a zero column or contains only one possible non-zero element, $\tilde{b}$. $(I + E^T\tilde{A}^{-1}V - D)^{-1}DE^T$ is a $2(p-1) \times n$ matrix. Each of its column either is a zero column or contains only two possible non-zero elements c_1, c_2, where

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = Z_1^{-1}\begin{pmatrix} 0 \\ w_0^{(i)} \end{pmatrix} = Z_1^{-1}\begin{pmatrix} 0 \\ \tilde{b} \end{pmatrix} = \frac{\tilde{b}}{1 - \tilde{a}^2}\begin{pmatrix} -\tilde{a} \\ 1 \end{pmatrix}, \tag{61}$$

and

$$\begin{pmatrix} c_2 \\ c_1 \end{pmatrix} = Z_1^{-1}\begin{pmatrix} v_{m-1}^{(i)} \\ 0 \end{pmatrix} = Z_1^{-1}\begin{pmatrix} \tilde{b} \\ 0 \end{pmatrix} = \frac{\tilde{b}}{1 - \tilde{a}^2}\begin{pmatrix} 1 \\ -\tilde{a} \end{pmatrix}. \tag{62}$$

For our application $A_i = A_1$, and $a_0^{(i)} = c_{m-1}^{(i)} = \lambda, i = 0, \cdots, p-1$. So, $v^{(i)} = v, w^{(i)} = w, i = 0, \cdots, p-1$ (see Eq. (15)). $(\tilde{A}^{-1}V)(I + E^T\tilde{A}^{-1}V - D)^{-1}D \cdot E^T$ is an $n \times n$ matrix, with each column is either zero column or contains only $c_1 w, c_2 v$ or $c_2 w, c_1 v$ respectively. Thus,

$$\begin{aligned} &||\tilde{A}^{-1}V(I + E^T\tilde{A}^{-1}V - D)^{-1}DE^T|| \leq max\{||c_2 v|| + ||c_1 w||, ||c_1 v|| + ||c_2 w||\} \\ &\leq |c_2|||w|| + |c_1|||v|| \qquad (note\ ||w|| = ||v||,\ Eq.(53)) \\ &= (|c_2| + |c_1|)||v|| = \tfrac{|\tilde{b}|}{|1-\tilde{a}^2|}(1 + |\tilde{a}|)||v|| = \tfrac{|\tilde{b}|}{|1-|\tilde{a}||}||v||. \end{aligned} \tag{63}$$

From our results given in Section 4.1,

$$|\tilde{a}| = \left|\frac{b(1 - b^{2m})}{\lambda(1 - b^{2(m+1)})}\right| \leq |\frac{b}{\lambda}|, \qquad |\tilde{b}| \leq \left|\frac{b^m}{\lambda}\right|, \tag{64}$$

and

$$\begin{aligned} v &= \tfrac{1}{\lambda a}\left(1 - \tfrac{b\sum_{i=0}^{m-1} b^{2i}}{a+b\sum_{i=0}^{m-1} b^{2i}}\right)\left(\sum_{i=0}^{m-1} b^{2i}, \sum_{i=1}^{m-1} b^{2i}/(-b), \cdots, (-b)^{m-1}\right)^T \\ &= \tfrac{1}{\lambda(a+b\sum_{i=0}^{m-1} b^{2i})}\left(\sum_{i=0}^{m-1} b^{2i}, \cdots, (-b)^{m-1}\right)^T. \end{aligned}$$

We have

$$\begin{aligned} ||v|| &= \left|\tfrac{1-b^2}{\lambda(a-b^{2m+1})}\right| \sum_{i=0}^{m-1} \left|\tfrac{(-b)^i(1-b^{2(m-i)})}{1-b^2}\right| \\ &\leq \left|\tfrac{1-b^2}{\lambda(a-b^{2m+1})}\right| \tfrac{(1+|b|^{m+1})(1-|b|^m)}{(1-b^2)(1-|b|)} \\ &\leq \tfrac{1}{|\lambda a|}\left|\tfrac{(1+|b|^{m+1})}{1-b^{2(m+1)}}\right|\left|\tfrac{(1-|b|^m)}{(1-|b|)}\right| \\ &\leq \tfrac{1}{|\lambda a|} \cdot \tfrac{1-|b|^m}{1-|b|^{m+1}} \cdot \tfrac{1}{1-|b|} \qquad (note\ |a| > 1,\ |b| < 1) \\ &\leq \tfrac{1}{|\lambda|(|a|-1)} \end{aligned} \tag{65}$$

Combining the inequalities (56) and (63) we obtain the final results

$$\frac{||x - x^*||}{||x||} \leq \frac{|\tilde{b}|}{|\lambda(1 - |\tilde{a}|)| \times (|a| - 1)} \tag{66}$$

$$\begin{aligned} \tfrac{||x-x^*||}{||x||} &\leq \tfrac{|b|^m}{|\lambda^2(1-|\tilde{a}|)(|a|-1)} \\ &= \tfrac{|b|^m}{|\lambda|\left(|\lambda|-\left|\frac{b(1-b^{2m})}{1-b^{2(m+1)}}\right|\right)(|a|-1)} \end{aligned} \tag{67}$$

Inequality (66) shows how the values of v_{m-1} and w_0 influence the accuracy of the final results. Inequality (67) gives an error bound of the PDD algorithm. When $\left|\frac{b}{\lambda}\right| < 1$, inequality (67) can be simplified to

$$\frac{||x - x^*||}{||x||} \leq \frac{|b|^m}{|\lambda|(|\lambda| - |b|)(|a| - 1)}$$

4.3 The Accuracy of the Reduced PDD Algorithm

For the sake of writing, in this section and next section we assume $m = n/p$ is an even integer. Let $\tilde{V}$ be the matrix corresponding to V in Eq.(9) such that $\tilde{A}^{-1}\tilde{V}$ results the vectors $\tilde{v}, \tilde{w}$ (see Section

3.3), and let x' be the solution of the reduced PDD algorithm. Then

$$x' = \tilde{A}^{-1}d - \tilde{A}^{-1}\tilde{V}(I + E^T\tilde{A}^{-1}V)E^T\tilde{A}^{-1}d. \tag{68}$$

As in Section 4.2, we let $y = (I + E^T\tilde{A}^{-1}V)E^T\tilde{A}^{-1}d$. Notice that x^* is the solution of the PDD algorithm (see Section 4.2). By Eq. (7) and (55),

$$x' - x^* = (\tilde{A}^{-1}\tilde{V} - \tilde{A}^{-1}V)y = (\tilde{A}^{-1}\tilde{V} - \tilde{A}^{-1}V)E^Tx,$$

Therefore,

$$\begin{aligned}
\tfrac{||x'-x^*||}{||x||} &\leq ||(\tilde{A}^{-1}\tilde{V} - \tilde{A}^{-1}V)|| \cdot ||E^T|| = ||\tilde{A}^{-1}\tilde{V} - \tilde{A}^{-1}V|| = ||\tilde{v} - v|| \\
&= \left|\tfrac{1}{\lambda(a+b\sum_{i=0}^{m-1} b^{2i})}\right| \sum_{i=\frac{m}{2}}^{m-1} \left|\tfrac{(-b)^i(1-b^{2(m-i)})}{1-b^2}\right| \\
&\leq \left|\tfrac{1}{\lambda a}\right| \left|\tfrac{1-b^2}{1-b^{2(m+1)}}\right| \left|\tfrac{(1+|b|^{m+1})(|b|^{\frac{m}{2}}(1-|b|^{\frac{m}{2}}))}{(1-b^2)(1-|b|)}\right| \\
&\leq \left|\tfrac{b^{\frac{m}{2}}}{\lambda a}\right| \left|\tfrac{1-|b|^{\frac{m}{2}}}{(1-|b|^{m+1})(1-|b|)}\right| \\
&\leq \tfrac{|b^{\frac{m}{2}}|}{|\lambda|(|a|-1)}
\end{aligned}$$

Equation (69) gives the accuracy of the reduced PDD algorithm.

$$\frac{||x - x'||}{||x||} \leq \frac{||x - x^*||}{||x||} + \frac{||x^* - x'||}{||x||} \tag{69}$$

$$\leq \frac{|b|^m}{|\lambda|\left(|\lambda| - \left|\frac{b(1-b^{2m})}{1-b^{2(m+1)}}\right|\right)(|a| - 1)} + \frac{|b^{\frac{m}{2}}|}{|\lambda|(|a| - 1)} \tag{70}$$

4.4 Anti-Symmetric Toeplitz Tridiagonal Systems

The accuracy analysis given by Sections 4.1 - 4.3 is for symmetric Toeplitz tridiagonal systems. In this section we extend the results to anti-symmetric Toeplitz tridiagonal systems. We assume that $m = n/p$ is an even number.

An anti-symmetric Toeplitz tridiagonal matrix A has the form $A = [-\lambda, \beta, \lambda] = \lambda \cdot [-1, c, 1]$. Let $B = [-1, c, 1]$. Then, for the corresponding matrix $\tilde{B}$ (see Section 4.1)

$$\tilde{B} = [b, 1, 0] \times [0, a, 1] \times [0, 1, -b],$$

where a, b are the solutions of

$$b + c = c, \quad b \cdot a = -1. \tag{71}$$

Comparing with symmetric case, the only difference are $-b$ in matrix $[0, 1, -b]$ and $b \cdot a = -1$ in Eq. (71). Following the steps given in the study of symmetric systems, we have computed the vectors of v and w in Eq. (15),

$$\begin{aligned} v &= \tfrac{1}{\lambda a}\left(1-\tfrac{b\sum_{i=0}^{m-1}(-1)^i b^{2i}}{a+b\sum_{i=0}^{m-1}(-1)^i b^{2i}}\right)\left(\sum_{i=0}^{m-1}(-1)^i b^{2i}, \sum_{i=1}^{m-1}(-1)^i b^{2i}/(b), \cdots, (-b)^{m-1}\right)^T \\ &= \tfrac{1}{\lambda(a+b\sum_{i=0}^{m-1}(-1)^i b^{2i})}\left(\sum_{i=0}^{m-1}(-1)^i b^{2i}, \cdots, (-b)^{m-1}\right)^T; \end{aligned} \tag{72}$$

$$w = \tfrac{1}{\lambda(a+b\sum_{i=0}^{m-1}(-1)^i b^{2i})}\left((-1)^{m-1}(-b)^{m-1}, (-1)^{m-2}\sum_{i=m-2}^{m-1}(-1)^i b^{2i}/b^i, \cdots, \sum_{i=0}^{m-1}(-1)^i b^{2i}\right)^T.$$

We can see for anti-symmetric Toeplitz tridiagonal systems $v_0^{(i)} = w_{m-1}^{(i)} = v_0^{(0)} = \tilde{a}$, and $v_{m-1}^{(i)} = -w_0^{(i)} = v_{m-1}^{(0)} = \tilde{b}$, for $i = 0, \cdots, p-1$. Thus, the inequality (63) remains true for anti-symmetric cases.

By Eq. (72), we have

$$\tilde{b} = v_{m-1}^{(i)} = \frac{(-b)^{m-1}}{\lambda(a+b\sum_{i=0}^{m-1}(-1)^i b^{2i})} = \frac{(-b)^m\cdot(1+b^2)}{\lambda(1+b^{2(m+1)})},$$

$$|\tilde{b}| \leq \frac{|b^m(1+b^2)|}{|\lambda|};$$

and

$$\tilde{a} = v_0^{(i)} = \frac{\sum_{i=0}^{m-1}(-1)^i b^{2i}}{\lambda(a+b\sum_{i=0}^{m-1}(-1)^i b^{2i})} = \frac{-b\cdot(1-b^{2m})}{\lambda(1+b^{2(m+1)})},$$

$$|\tilde{a}| = \left|\frac{-b\cdot(1-b^{2m})}{\lambda(1+b^{2(m+1)})}\right| \leq \frac{|b|}{\lambda}.$$

For the bound of the norm of vector v (see Eq. (65)). When $b \cdot a = -1$,

$$\begin{aligned} ||v|| &\leq \left|\tfrac{1}{\lambda a}\left(\tfrac{a}{a+b\sum_{i=0}^{m-1}(-1)^i b^{2i}}\right)\right|\left|\tfrac{(1+|b|)^{m+1}(1-|b|^{2m})}{(1+b^2)(1-|b|)}\right| \\ &\leq \tfrac{1}{|\lambda a(1+b^{2(m+1)})|}\cdot\tfrac{1-|b|^{2(m+1)}}{1-|b|} \leq \tfrac{1}{|\lambda(|a|-1)|}. \end{aligned}$$

The corresponding relative error

$$\frac{||x-x^*||}{||x||} \leq \frac{|\tilde{b}|}{|\lambda(1-|\tilde{a}|)(|a|-1)|} \tag{73}$$

in terms of $\tilde{a}$ and $\tilde{b}$; and

$$\frac{||x-x^*||}{||x||} \leq \frac{|b|^m(1+b^2)}{|\lambda^2(1-|\tilde{a}|)(|a|-1)|} = \frac{|b|^m(1+b^2)}{|\lambda(|\lambda| - \left|\frac{b(1-b^{2m})}{1+b^{2(m+1)}}\right|(|a|-1)|} \tag{74}$$

System	Matrix	Best sequential	the PDD Computation	the PDD Communication
Single System	Non-periodic	8n-7	$17\frac{n}{p}-4$	$2\alpha+12\beta$
	Periodic	14n-16	$17\frac{n}{p}-4$	$2\alpha+12\beta$
Multiple right-side	Non-periodic	$(5n-3)*nl$	$(9\frac{n}{p}+1)*nl$	$(2\alpha+8\beta)*nl$
	Periodic	$(7n-1)*nl$	$(9\frac{n}{p}+1)*nl$	$(2\alpha+8\beta)*nl$

Table 1. Computation and Communication Counts of the PDD Algorithm

in terms of a and b. When $\frac{|b|}{|\lambda|} < 1$, we have

$$\frac{||x-x^*||}{||x||} \leq \frac{|b|^m(1+b^2)}{|\lambda(|\lambda|-|b|)(|a|-1)|}$$

For the reduced PDD algorithm, when the system is anti-symmetric, we have

$$\begin{aligned} \frac{||x^*-x'||}{||x||} &\leq ||\tilde{v}-v|| \\ &= \left|\frac{1}{\lambda}\cdot\frac{1+b^2}{1+b^{2(m+1)}}\right|\left|\frac{(1+|b|^{2(m+1)})(|b|^{m/2}(1-|b|^{m/2}))}{(1+b^2)(1-|b|)}\right| \\ &\leq \frac{|b|^{m/2}}{|\lambda|(|a|-1)}. \end{aligned}$$

and

$$\frac{||x-x'||}{||x||} \leq \frac{|b|^m(1+b^2)}{|\lambda(|\lambda|-\left|\frac{b(1-b^{2m})}{1+b^{2(m+1)}}\right|(|a|-1)|} + \frac{|b|^{m/2}}{|\lambda|(|a|-1)}.$$

5 Experimental Results

Table 1 gives the computation and communication count of the PDD algorithm. Since the tridiagonal systems arising in both ADI and in the compact scheme method are multiple right-side systems, the computation and communication count of solving multiple right-side systems is also listed in Table 1, where the factorization of matrix A is not considered and nl is the number of right-sides. Note for multiple right-side systems, the communication cost increases with the number of right-sides. Table 2 gives the computation and communication counts of the reduced PDD algorithm. As the PDD algorithm, it has the same parallel computation and communication counts for periodic and non-periodic systems.

A sample matrix is chosen to illustrate and verify the algorithm and theoretical results given in previous sections. The sample matrix A is a resulting matrix of the compact scheme,

$$A = [\frac{1}{3}, 1, \frac{1}{3}]. \tag{75}$$

System	the Reduced PDD	
	Computation	Communication
Single system	$15\frac{n}{p} - 4$	$2\alpha + 12\beta$
Multiple right-side	$(7\frac{n}{p} + 1) * n1$	$(2\alpha + 8\beta) * n1$

Table 2. Computation and Communication Counts of the Reduced PDD

For matrix A,

$$A = [\frac{1}{3}, 1, \frac{1}{3}] = \frac{1}{3} \cdot [1, 3, 1] = \frac{1}{3} \cdot ([b, 1, 0] \times [0, a, 0] \times [0, 1, b] - \Delta B),$$

where ΔB is given by Eq.(44), and

$$\lambda = \frac{1}{3}, c = 3, a = \frac{3 + \sqrt{5}}{2}, b = \frac{3 - \sqrt{5}}{2}. \tag{76}$$

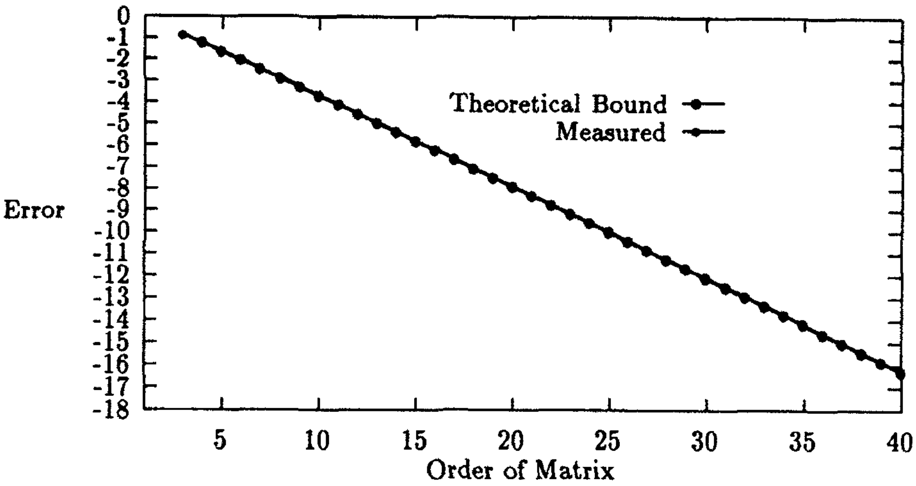


Figure 2. Measured and Predicted Decay Rate.

The PDD algorithm was first implemented on a X11r4 terminal to solve the corresponding periodic system of $Ax = d$ for accuracy checking. Then the algorithm was implemented on a 32-node Intel/860 to measure the speedup over Thomas algorithm [7], a commonly used practical sequential algorithm for periodic tridiagonal systems. For accuracy checking, all the measured and predicted data have been converted by a logarithm function with base ten to make the difference visible. Figure 2 depicts the decay rate of v_{m-1} of matrix A, where the x-coordinate is the order of the sub-system A_i and the y-coordinate is the value of v_{m-1}. We can see that the theoretical bound given in Section 4.1 coincides with the measured value.

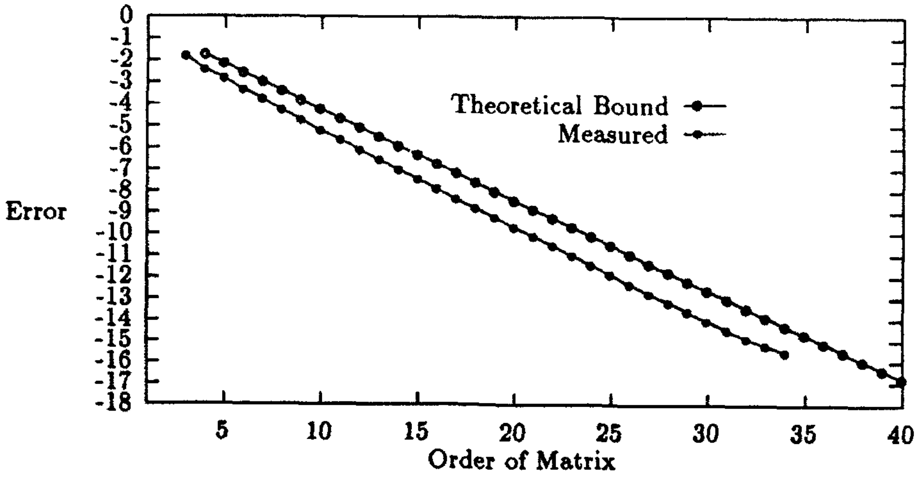


Figure 3. Measured and Predicted Accuracy of the PDD Algorithm.

Accuracy comparisons of the PDD and the reduced PDD algorithms are given in Fig. 3 and Fig. 4 respectively. For the accuracy comparisons, the right-side vector, d, was randomly generated. The x-coordinate is the order of matrix A, and the y-coordinate is the relative error in the 1-norm. These two figures show that our accuracy analysis provides a very good bound.

Figure 5 and 6 give the speedup of the PDD algorithm over Thomas algorithm. For single system, the order of matrix A is limited by the machine memory for $n = 6400$. For multiple right-sides, the system is limited for $n = 128$ and $n1 = 4096$. From Fig. 5 we can see that the speedup of solving a single system increases linearly with the number of processors. Figure 6 shows that the linear increasing property does not hold for multiple right-side systems. The lower speedup is due to the increase of communication cost. Since the Intel/860 has a very high (communication speed)/(computation speed) ratio, we can expected a better speedup on an Intel Paragon or even on an Intel/iPSC2 [18] multicomputer.

6 Conclusion

A detailed study has been given for the efficient tridiagonal solver, the Parallel Diagonal Dominant (PDD) algorithm. The presented PDD algorithm is slightly different from the originally proposed version [21] and is also extended to periodic systems. A variant, the reduced PDD algorithm, was also introduced. Accuracy analysis is provided for a class of tridiagonal systems, the symmetric

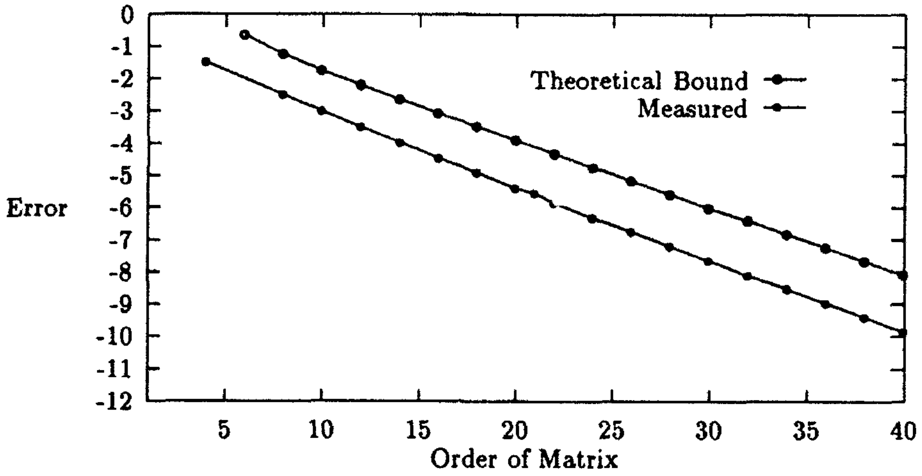


Figure 4. Measured and Predicted Accuracy of the Reduced PDD .

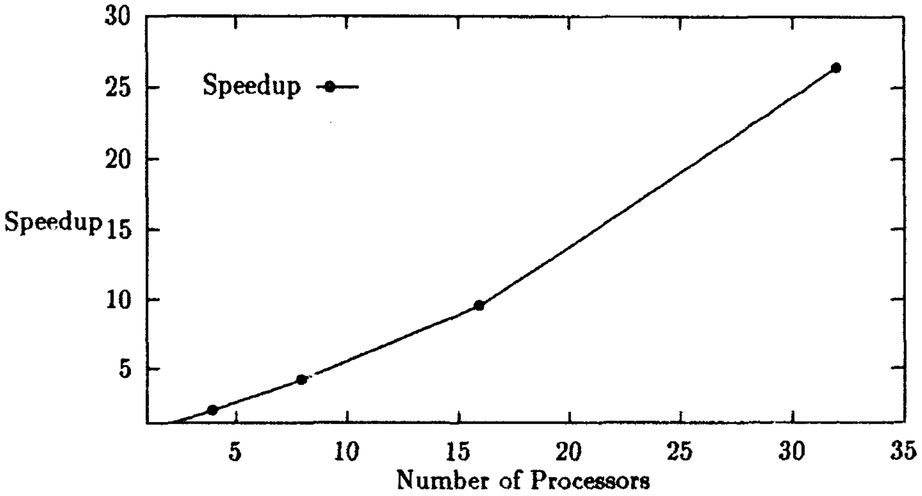


Figure 5. Measured Speedup Over Thomas Algorithm.

Single System of Order 6400

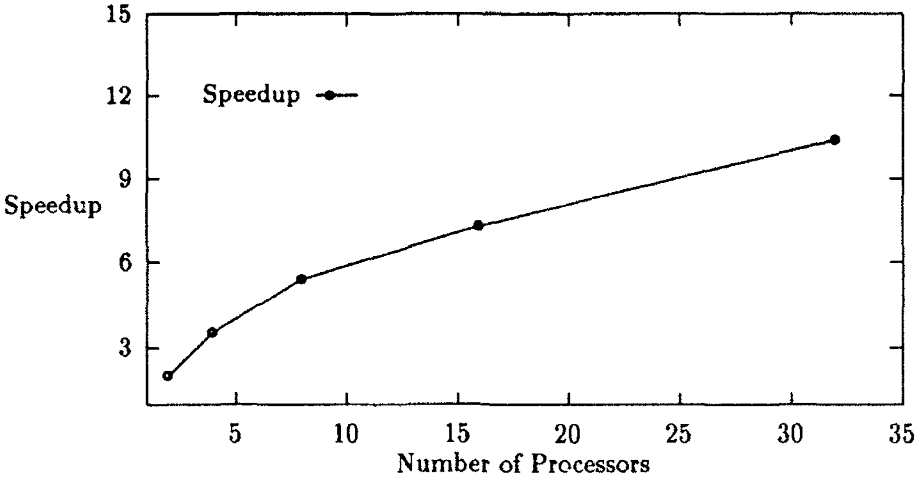


Figure 6. Measured Speedup Over Thomas Algorithm.

4096 Systems of Order 128, Factorization Time Not Included

and anti-symmetric Toeplitz tridiagonal systems. Implementation results were provided for both accuracy analysis and for the proposed algorithm. They showed that the accuracy analysis provides a very good theoretical bound and that the algorithm is highly efficient for both single and multiple right-side systems. The algorithm is a good candidate for large scale computing, where the number of processors and the problem size are large. It is a good choice for the newly emerged massively parallel machines, such as Thinking Machine Corporations's CM-5 and Intel's Paragon. The discussion is based on distributed-memory machines. The result can be easily applied to shared-memory machines as well.

The PDD algorithm and the reduced PDD algorithm proposed in this paper can be extended to band systems and block tridiagonal systems. The accuracy analysis, which gives a good, simple relative error bound, is for symmetric and anti-symmetric Toeplitz tridiagonal systems only. It is unlikely that the analysis can be extended for general case with the same technique.

Acknowledgements

I am indebted to several individuals at ICASE for their help with this research. In particular, Gordon Erlebacher brought the issue of designing fast symmetric Toeplitz tridiagonal solvers for compressible and uncompressible flow. John R. Van_Rosendale offered valuable suggestions and comments which improved the technical quality and presentation of this paper.

References

[1] CHUNG, K.-L., AND SHEN, L.-J. Vectorized algorithm for b-spline curve fitting on cary x-mp ea/16se. In *Proc. of Supercomputing'92* (1992), pp. 166–169.

[2] DEMKO, S., MOSS, W., AND SMITH, P. Decay rates for inverses of band matrices. *Mathemat. Comput. 43*, 168 (1984), 491–499.

[3] DUFF, I., ERISMAN, A., AND REID, J. *Direct Methods for Sparse Matrices.* Clarendon Press, Oxford, 1986.

[4] DUNIGAN, T. Performance of the intel ipsc/860 and ncube 6400 hypercubes. *Parallel Computing* (Dec. 1991), 1285–1302.

[5] EGECIOGLU, O., KOC, D., AND LAUB, A. A recursive doubling algorithm for solution of tridiagonal systems on hypercube multiprocessors. *J. of Comp. and Appl. Math. 27* (1989).

[6] GUSTAFSON, J., MONTRY, G., AND BENNER, R. Development of parallel methods for a 1024-processor hypercube. *SIAM J. on SSTC 9*, 4 (July 1988).

[7] HIRSCH, C. *Numerical Computation of Internal and External Flows.* John Wiley & Sons, 1988.

[8] HO, C., AND JOHNSSON, S. Optimizing tridiagonal solvers for alternating direction methods on boolean cube multiprocessors. *SIAM J. Sci. Stat. Comput. 11*, 3 (1990), 563–592.

[9] HOCKNEY, R. A fast direct solution of poisson's equation using fourier analysis. *J. ACM 12* (1965), 95–113.

[10] HWANG, K. *Advanced Computer Architectures with Parallel Programming.* McGraw-Hill Book Co., 1992.

[11] LAWRIE, D., AND SAMEH, A. The computation and communication complexity of a parallel banded system solver. *ACM Trans. Math. Soft. 10*, 2 (June 1984), 185–195.

[12] LELE, S. Compact finite difference schemes with spectral-like resolution. *J. Comp. Phys. 103*, 1 (1992), 16–42.

[13] LI, T., ZHANG, H., AND SUN, X.-H. Parallel homotopy algorithm for symmetric tridiagonal eigenvalue problem. *SIAM J. of Sci. Stat. Comput. 12* (May 1991).

[14] ORTEGA, J., AND VOIGT, R. Solution of partial differential equations on vector and parallel computers. *SIAM Review* (June 1985).

[15] SHERMAN, J., AND MORRISON, W. Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix. *Ann. Math. Stat. 20*, 621 (1949).

[16] 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.

[17] STRIKWERDA, J. C. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth & Brooks/Cole, Mathematics Series, 1989.

[18] SUN, X.-H., AND GUSTAFSON, J. Toward a better parallel performance metric. *Parallel Computing 17* (Dec 1991), 1093–1109.

[19] SUN, X.-H., AND NI, L. Another view on parallel speedup. In *Proc. of Supercomputing'90* (NY, NY, 1990), pp. 324–333.

[20] SUN, X.-H., AND ROVER, D. Scalability of parallel algorithm-machine combinations. Technical Report, IS-5057, UC-32, Ames Laboratory, U.S. Department of Energy, 1991.

[21] SUN, X.-H., ZHANG, H., AND NI, L. Efficient tridiagonal solvers on multicomputers. *IEEE Transactions on Computers 4*, 3 (1992), 286–296.

[22] WANG, H. A parallel method for tridiagonal equations. *ACM Trans. Math. Software 7* (June 1981), 170–183.

[23] WOODBURY, M. Inverting modified matrices. Memorandum 42, Statistics Research Group, Princeton University, 1950.

[24] ZHANG, H. On the accuracy of the parallel diagonal dominat algorithm. *Parallel Computing 17* (1991), 265–272.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
February 1993

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
APPLICATIONS AND ACCURACY OF THE PARALLEL DIAGONAL DOMINANT ALGORITHM

5. FUNDING NUMBERS
C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)
Xian-He Sun

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 93-6

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-191431
ICASE Report No. 93-6

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Michael F. Card
Final Report

Submitted to the International Conference on Parallel Processing

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 59, 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Parallel Diagonal Dominant (PDD) algorithm is a highly efficient, ideally scalable tridiagonal solver. In this paper, a detailed study of the PDD algorithm is given. First the PDD algorithm is introduced. Then the algorithm is extended to solve periodic tridiagonal systems. A variant, the reduced PDD algorithm, is also proposed. Accuracy analysis is provided for a class of tridiagonal systems, the symmetric and anti-symmetric Toeplitz tridiagonal systems. Implementation results show that the analysis gives a good bound on the relative error, and the algorithm is a good candidate for the emerging massively parallel machines.

14. SUBJECT TERMS
parallel processing; parallel numerical algorithm; scalable computing; tridiagonal system; Toeplitz systems

15. NUMBER OF PAGES
27

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

20. LIMITATION OF ABSTRACT

NASA-Langley, 1993