Pergamon

Computers Math. Applic. Vol. 32, No. 12, pp. 1–14, 1996

0898-1221/96 $15.00 + 0.00

PII: S0898-1221(96)00203-9

Vectorized Algorithms for Solving Special Tridiagonal Systems

KUO-LIANG CHUNG*
Department of Information Management, National Taiwan Institute of Technology
No. 43, Section 4, Keelung Road, Taipei, Taiwan 10672, R.O.C.
klchung@cs.ntit.edu.tw

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

(Received November 1995; revised and accepted September 1996)

Abstract—Solving special tridiagonal systems often arise in the fields of engineering and science. This special tridiagonal system is diagonally dominant and circulant near-Toeplitz. This paper presents two fast vectorized algorithms for solving special tridiagonal systems. Both algorithms employ the matrix perturbation technique and have many computational advantages on vector supercomputer. The related error analysis are also given. Some experimental results are illustrated on vector uniprocessor of the CRAY X-MP EA/116se.

Keywords—Circulant near-Toeplitz systems, CRAY X-MP, Diagonally dominant, Error analyses, Tridiagonal matrices, Vectorized algorithms.

1. INTRODUCTION

In this paper, we are interested in the solution of the special tridiagonal system

$$A_n \mathbf{x} = \mathbf{b} \tag{1}$$

of order n on vector uniprocessor, where

$$A_n = \begin{pmatrix} \beta_1 & \beta_2 & & & & \beta_3 \\ \alpha & \beta & \gamma & & & \\ & & \cdot & \cdot & \cdot & \\ & & & \alpha & \beta & \gamma \\ \beta_3' & & & & \beta_2' & \beta_1' \end{pmatrix}$$

and $|\beta| > |\alpha + \gamma|$. Solving (1) arises in many computational problems [1–7], in which it is one of the most time-consuming elements. The availability of vector supercomputers has had a significant impact on scientific computations [8–10]. The motivation of this research is to design efficient vectorized algorithms for solving (1).

*Author to whom all correspondence should be addressed.

The authors would like to thank the help of the referees and Prof. E. Y. Rodin which led to an improved version of this paper. This research was supported in part by the National Science Council of R.O.C. under Grants NSC85-2213-E011-009, NSC85-2121-M011-002, and NCHC86-08-015.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

In this paper, two fast vectorized algorithms for solving (1) are presented. Both new algorithms consist of three phases and only differ in the second phase. The first phase is a Toeplitz factorization of a slightly perturbed matrix of A_n. The second phase is to solve the perturbed problem in a highly vectorized way, but only scale $\times$ vector operations are involved, hence, it leads to a great deal of computational saving. In the third phase, the solution to the original problem is recovered from the solution to the perturbed problem; this is called the update procedure. Some error analyses are also given. In addition, some experimental results are illustrated on CRAY X-MP EA/116se.

Section 2 presents our vectorized algorithms for solving (1) and the related error analyses. The implementations on the CRAY X-MP EA/116se are illustrated in Section 3. Section 4 gives the conclusions.

2. VECTORIZATIONS

2.1. Toeplitz Factorization

Throughout the remainder of this paper, matrices are represented by uppercase letters, vectors by bold lowercase letters, and scalars by plain lowercase letters. The superscript $^{\top}$ corresponds to the transpose operation; $\|\bullet\|$ denotes the sup-norm of one vector.

In order to avoid memory conflict in CRAY X-MP EA/116se [11], we enlarge (1), then perturb it to $A'_m\mathbf{x}' = \mathbf{b}'$, where $m = pq$ (p denotes the length of one register file in the vector computer, i.e., the vector length), $\mathbf{b}' = (\underbrace{b_1, b_2, \dots, b_n}_{n}, \underbrace{0, \dots, 0}_{n-m})^{\top}$, and

$$A' = \begin{pmatrix} a & \gamma & & & & \\ \alpha & \beta & \gamma & & & \\ & \cdot & \cdot & \cdot & & \\ & & & \alpha & \beta & \gamma \\ & & & & \alpha & \beta \end{pmatrix} = L'U', \tag{2}$$

$$L' = \begin{pmatrix} 1 & & & & \\ -d & 1 & & & \\ & \cdot & \cdot & & \\ & & -d & 1 & \\ & & & -d & 1 \end{pmatrix} \quad \text{and} \quad U' = \begin{pmatrix} a & \gamma & & & \\ & a & \gamma & & \\ & & \cdot & \cdot & \\ & & & a & \gamma \\ & & & & a \end{pmatrix},$$

which implies that $a - \gamma d = \beta$ and $-ad = \alpha$. This in turn implies that $d = -(\alpha/a)$ and $a = (\beta \pm \sqrt{\beta^2 - 4\gamma\alpha})/2$. Since we wish the matrices L' and U' to be diagonally dominant, we will select the sign so that the absolute value of a is greater than $\max(|\alpha|, |\gamma|)$. That is, when $\beta > |\alpha + \gamma|$, we choose $a = (\beta + \sqrt{\beta^2 - 4\gamma\alpha})/2$; when $\beta < -|\alpha + \gamma|$, we choose $a = (\beta - \sqrt{\beta^2 - 4\gamma\alpha})/2$. Since one of our choices always makes $|a| > \max(|\alpha|, |\gamma|)$, hence, the bidiagonal Toeplitz matrices L' and U' are diagonally dominant. The computation of a and d provides the Toeplitz factorization of the matrix A', which can be done in $O(1)$ time.

2.2. Solving the Perturbed System

2.2.1. The first method

In this section, our first vectorized method for solving $A'_m\mathbf{x}' = \mathbf{b}'$ in a highly vectorized way consists of two parts:

(1) vectorization of a lower bidiagonal Toeplitz system, $L'_m\mathbf{y}' = \mathbf{b}'$;
(2) vectorization of an upper bidiagonal Toeplitz system, $U'_m\mathbf{x}' = \mathbf{y}'$.

We would especially point out that due to our matrix perturbation technique, all the vector operations involved are scaled by a constant, which is very important for the efficient implementation on the vector computer.

For convenience, we first describe the vectorized method for solving the general lower bidiagonal Toeplitz system (LBTS), $L_m\mathbf{x} = \mathbf{y}$, where

$$L_m = \begin{pmatrix} r & & & & \\ s & r & & & \\ & \cdot & \cdot & & \\ & & s & r & \\ & & & s & r \end{pmatrix}_{m\times m}, \qquad \text{for } |r| > |s|$$

and $\mathbf{y} = (y_1, y_2, \dots, y_m)^\top$. In Section 2.1, we know that $m = pq$. Therefore, we partition the above LBTS into p LBTS's. Each LBTS can be written as

$$L_q\mathbf{x}'^{(i)} = \mathbf{y}^{(i)}, \qquad \text{for } i = 0, 1, \dots, p-1, \tag{3}$$

where $\mathbf{x}'^{(i)} = (x'_{iq+1}, x'_{iq+2}, \dots, x'_{iq+q})^\top$ and $\mathbf{y}^{(i)} = (y_{iq+1}, y_{iq+2}, \dots, y_{iq+q})^\top$. Our vectorized subroutine for solving these p smaller LBTS's is shown in Appendix 1, where Loop-5 and Loop-20 can be vectorized with vector length p. Specifically, only scalar $\times$ vector operations are involved in this subroutine.

After solving $\mathbf{x}'$ in a vectorized way, we have

$$L_m\mathbf{x}' = \mathbf{y} + s\sum_{i=1}^{p-1} x'_{iq}\mathbf{e}_{iq+1}, \tag{4}$$

where $e_k = (\underbrace{0, \dots, 0, 1}_{k}, \underbrace{0, \dots, 0}_{m-k})^\top$ for $1 \le k \le m$. Since

$$L_m\mathbf{e}_k = r\mathbf{e}_k + s\mathbf{e}_{k+1}, \qquad 1 \le k < m,$$

we have

$$\begin{aligned} L_m\left(\sum_{j=0}^{t-1}\frac{1}{r}\left(-\frac{s}{r}\right)^j \mathbf{e}_{k+j}\right) &= \sum_{j=0}^{t-1}\left(-\frac{s}{r}\right)^j \mathbf{e}_{k+j} - \sum_{j=0}^{t-1}\left(-\frac{s}{r}\right)^{j+1} \mathbf{e}_{k+j+1} \\ &= \mathbf{e}_k - \left(-\frac{s}{r}\right)^t \mathbf{e}_{k+t}. \end{aligned} \tag{5}$$

Let

$$\mathbf{x} = \mathbf{x}' - \frac{s}{r}\sum_{i=1}^{p-1} x'_{iq}\sum_{j=0}^{t-1}\left(-\frac{s}{r}\right)^j \mathbf{e}_{iq+j+1}, \tag{6}$$

where t denotes the length of the update vector which will be discussed in Section 2.3, the solution vector $\mathbf{x}$ of (6) can be computed by using the vectorized subroutine as shown in Appendix 2, where Loop-40 can be vectorized with vector length $p-1$. Only scalar $\times$ vector operations are involved in this subroutine.

By (5) and (6), we have

$$\begin{aligned} L_m\mathbf{x} &= \mathbf{y} + s\sum_{i=1}^{p-1} x'_{iq}\mathbf{e}_{iq+1} - s\sum_{i=1}^{p-1} x'_{iq}\left(\mathbf{e}_{iq+1} - \left(-\frac{s}{r}\right)^t \mathbf{e}_{iq+t+1}\right) \\ &= \mathbf{y} + s\sum_{i=1}^{p-1} x'_{iq}\left(-\frac{s}{r}\right)^t \mathbf{e}_{iq+t+1}, \end{aligned}$$

hence,

$$\|L_m\mathbf{x} - \mathbf{y}\| \le |s|\left|\frac{s}{r}\right|^t \|\mathbf{x}'\|. \tag{7}$$

To estimate $\|\mathbf{x}'\|$, we need the following lemma.

LEMMA 1. *If*

$$L = \begin{pmatrix} r & & & & \\ s & r & & & \\ & \cdot & \cdot & & \\ & & s & r & \\ & & & s & r \end{pmatrix}, \qquad |r| > |s|,$$

then $\|L^{-1}\mathbf{y}\| \leq 1/(|r| - |s|)\|\mathbf{y}\|$.

PROOF. See Appendix 3.

Using the above similar partition approach, it is easy to design our two vectorized subroutines, VUPPER1(r,s) (vs. VLOW1(r,s)) and UPDATEUPPER1(r,s,t) (vs. UPDATELOW1(r,s,t)), for solving the general upper bidiagonal Toeplitz system (UBTS), $U_m\mathbf{x} = \mathbf{y}$, where

$$U_m = \begin{pmatrix} r & s & & & \\ & r & s & & \\ & & \cdot & \cdot & \\ & & & r & s \\ & & & & r \end{pmatrix}, \qquad \text{for } |r| > |s|.$$

For saving space, we omit the pseudo codes for these two subroutines.

Return to solve $A'_m\mathbf{x}' = \mathbf{b}'$. Since $A'_m = L'_m U'_m$, we first solve $L'_m\mathbf{y}' = \mathbf{b}'$, where $\mathbf{b}' = (b_1, b_2, \ldots, b_n, \underbrace{0, \ldots, 0}_{m-n})^{\top}$, then solve $U'_m\mathbf{x}' = \mathbf{y}'$. Using our previous vectorized methods for solving general lower and upper bidiagonal systems, the following algorithm is used to solve $A'_m\mathbf{x}' = \mathbf{b}'$.

```
C******the entry array y represents vector b'
C******the exit array represents vector x'
       CALL VLOW1(1,-d)
       CALL UPDATELOW(1,-d,t1)
       CALL VUPPER1(a,gamma)
       CALL UPDATEUPPER(a,gamma,t2)
```

LEMMA 2. *Let* $c_1 = 1/(1 - |d|)$ *and* $c_2 = (1 + |d|)|\gamma|(1/(|a| - |\gamma|))((1 + |d|)/(1 - |d|))$, *then*

$$\|A'_m\mathbf{x}' - \mathbf{b}'\| \leq \left(c_1|d|^{t_1+1} + c_2\left|\frac{\gamma}{a}\right|^{t_2}\right)\|\mathbf{b}\|.$$

The term $c_1|d|^{t_1+1}$ *will be less than* ξ, *if* t_1 *is greater than* $(\log\xi - \log c_1)/\log|d| - 1$; *the term* $c_2|\gamma/a|^{t_2+1}$ *will be less than* ξ, *if* t_2 *is greater than* $((\log\xi - \log c_2)/(\log|\gamma/a|)) - 1$. *Let* $c_3 = ((1 + |\gamma/a|)/(|a| - |\gamma|))((1 + |d|)/(1 - |d|))$, *then it yields*

$$\|\mathbf{x}'\| \leq c_3\|\mathbf{b}\|.$$

PROOF. See Appendix 4.

Lemma 2 will be used in the analysis of the update phase in Section 2.3.

In the following section, based on the product expansion method [12,13], we present the second vectorized method for solving $A'_m\mathbf{x}' = \mathbf{b}'$, where $m = n$, since we do not need the partition approach as described in Section 2.2.1.

2.2.2. The second method

Now we describe our second vectorized method for solving $L_m\mathbf{x} = \mathbf{y}$. All the vector operations involved are scaled only by a constant.

Since $L_m = r(I - E)$ with

$$E = \begin{pmatrix} 0 & & & & \\ -\frac{s}{r} & 0 & & & \\ & \ddots & & & \\ & & -\frac{s}{r} & 0 & \\ & & & -\frac{s}{r} & 0 \end{pmatrix},$$

the system $L_m\mathbf{x} = \mathbf{y}$ is equal to $(I - E)\mathbf{x} = (1/r)\mathbf{y}$. $\mathbf{x}$ can be obtained by computing $\mathbf{x} = (I + E^{2^k}) \cdots (I + E^2)(I + E)((1/r)\mathbf{y})$ because

$$\begin{aligned} L_m\mathbf{x} &= r(I - E)\mathbf{x} \\ &= r(I - E)(I + E)\left(I + E^2\right) \cdots \left(I + E^{2^k}\right)\left(\frac{1}{r}\mathbf{y}\right) \\ &= \left(I - E^{2^{k+1}}\right)\mathbf{y}. \end{aligned}$$

Since

$$E^{2^{k+1}} = \begin{pmatrix} \left(-\frac{s}{r}\right)^{2^{k+1}} & & & \\ & \ddots & & \\ & & \left(-\frac{s}{r}\right)^{2^{k+1}} & \\ & & & \left(-\frac{s}{r}\right)^{2^{k+1}} \end{pmatrix},$$

we then have

$$\|L_m\mathbf{x} - \mathbf{y}\| \le \left|\frac{s}{r}\right|^{2^{k+1}} \|\mathbf{y}\|. \tag{8}$$

The relative residual $(\|L_m\mathbf{x} - \mathbf{y}\|)/\|\mathbf{y}\|$ will be less than ξ, if k is greater than $\log(\log\xi/(\log|s/r|))$ -1. Therefore, the computation of $\mathbf{x}$ $(= \mathbf{x}^{(k+1)})$ can be accomplished by the following iterative formula:

$$\mathbf{x}^{(i+1)} = \left(I + E^{2^i}\right)\mathbf{x}^{(i)} = \mathbf{x}^{(i)} + E^{2^i}\mathbf{x}^{(i)}, \qquad 0 \le i \le k, \tag{9}$$

with the initial assignment $\mathbf{x}^{(0)} = (1/r)\mathbf{y}$.

From (8), we have $\|\mathbf{x}^{(i+1)}\| \le (1 + |s/r|^{2^i})\|\mathbf{x}^{(i)}\|$. Thus,

$$\begin{aligned} \left\|\mathbf{x}^{(k+1)}\right\| &\le \left(1 + \left|\frac{s}{r}\right|^{2^k}\right)\left(1 + \left|\frac{s}{r}\right|^{2^{k-1}}\right) \cdots \left(1 + \left|\frac{s}{r}\right|\right)\left|\frac{1}{r}\right| \|\mathbf{y}\| \\ &= \frac{1 - |s/r|^{2^{k+1}}}{1 - |s/r|}\left|\frac{1}{r}\right| \|\mathbf{y}\| \\ &\le \frac{1}{|r| - |s|}\|\mathbf{y}\|. \end{aligned} \tag{10}$$

The formal vectorized subroutine for computing $\mathbf{x}^{(k+1)}$ is shown in Appendix 5, where mainly scalar $\times$ vector operations are involved in this subroutine.

Similarly, it is easy to design our vectorized subroutine for solving the general UBTS system, $U_m\mathbf{x} = \mathbf{y}$. The corresponding vectorized subroutine is shown in Appendix 6, where mainly scalar $\times$ vector operations are involved in this subroutine.

Using our previous two vectorized subroutines shown in Appendices 5 and 6 for solving general lower and upper bidiagonal systems, respectively, the following procedure is used to solve $A'_m\mathbf{x}' = \mathbf{b}'$.

```
CALL VLOW2(1,-d,k1)
CALL VUPPER2(a,gamma,k2)
```

LEMMA 3. *Let $c'_1 = 1$ and $c'_2 = (1+|d|)/(1-|d|)$, then we have*

$$\|A'_m\mathbf{x}' - \mathbf{b}'\| \le \left(c'_1|d|^{2^{k_1+1}} + c'_2\,|d'|^{2^{k_2+1}}\right)\|\mathbf{b}\|.$$

The term $c'_1|d|^{2^{k_1+1}}$ will be less than ξ, if k_1 is greater than $\log(\log\xi/\log|d|) - 1$; the term $c'_2|d'|^{2^{k_2+1}}$ will be less than ξ, if k_2 is greater than $\log((\log\xi - \log c'_2)/\log|d|) - 1$. Let $c'_3 = (1/(|a|-|\gamma|)(1/(1-|d|))$, then it follows that

$$\|\mathbf{x}'\| \le c'_3\|\mathbf{b}\|.$$

PROOF. See Appendix 7.

Lemma 3 will be used in the following section.

2.3. Update

After solving the perturbed system $A'_m\mathbf{x}' = \mathbf{b}'$ approximately, the approximate solution of $\mathbf{x}$ will be recovered from the perturbed system in this section.

Let $\mathbf{z}' = A'_m\mathbf{x}' - \mathbf{b}'$, $\bar{\mathbf{z}} = (z'_1, z'_2, \dots, z'_n)^\top$, $\bar{\mathbf{x}} = (x'_1, x'_2, \dots, x'_n)^\top$, and $\mathbf{w} = A_n\bar{\mathbf{x}} - \mathbf{b}$. Since

$$\begin{aligned} w_i &= \alpha x'_{i-1} + \beta x'_i + \gamma x'_{i+1} - b_i = z'_i, \qquad \text{for } 1 < i < n,\\ w_1 &= \beta_1 x'_1 + \beta_2 x'_2 + \beta_3 x'_n - b_1,\\ w_n &= \beta'_3 x'_1 + \beta'_2 x'_{n-1} + \beta_1 x'_n - b_n, \end{aligned} \tag{11}$$

we have $\|\mathbf{w} - w_1\mathbf{e}_1 - w_n\mathbf{e}_n\| \le \|\bar{\mathbf{z}}\| \le \|\mathbf{z}'\|$. By Lemma 2, it follows that

$$\|A_n\bar{\mathbf{x}} - \mathbf{b} - w_1\mathbf{e}_1 - w_n\mathbf{e}_n\| \le \|\mathbf{z}'\| \le \left(c_1|d|^{t_1+1} + c_2\left|\frac{\gamma}{a}\right|^{t_2}\right)\|\mathbf{b}\|. \tag{12}$$

The value of $\|\mathbf{z}'\|$ will be very small when t_1 and t_2 are large enough. Therefore, the approximate solution of $\mathbf{x}$ to be determined equals $\bar{\mathbf{x}} - \mathbf{p}$, where

$$A\mathbf{p} \doteq w_1\mathbf{e}_1 + w_n\mathbf{e}_n. \tag{13}$$

To solve $\mathbf{p}$, we try to ignore the first and last equalities of the system $A\mathbf{p} \doteq w_1\mathbf{e}_1 + w_n\mathbf{e}_n$, then we must solve the recurrence relation: $\alpha p_{i-1} + \beta p_i + \gamma p_{i+1} = 0$ for $2 \le i \le n-1$. From $a - \gamma d = \beta$, $-ad = \alpha$, it follows that $\alpha + \beta d + \gamma d^2 = 0$ and $\gamma + \beta d' + \alpha d'^2 = 0$, where $d' = -\gamma/a$. Naturally, if we try $\mathbf{p}_1 = (\underbrace{d, d^2, \dots, d^{t_3}}_{t_3}, \underbrace{0, \dots, 0}_{n-t_3})^\top$, then

$$\begin{aligned} A_n\mathbf{p}_1 &= \left(\underbrace{\beta_1 d + \beta_2 d^2, \alpha d + \beta d^2 + \gamma d^3, \dots, \alpha d^{t_3-1} + \beta d^{t_3}}_{t_3}, \underbrace{\alpha d^{t_3}, 0, \dots, 0, \beta'_3 d}_{n-t_3}\right)^\top \\ &= \left(\underbrace{\beta_1 d + \beta_2 d^2, 0, \dots, 0, \alpha d^{t_3-1} + \beta d^{t_3}}_{t_3}, \underbrace{\alpha d^{t_3}, 0, \dots, 0, \beta'_3 d}_{n-t_3}\right)^\top \\ &= u\mathbf{e}_1 + v\mathbf{e}_n + \alpha d^{t_3}\left(-d\frac{\gamma}{\alpha}\mathbf{e}_{t_3} + \mathbf{e}_{t_3+1}\right), \end{aligned} \tag{14}$$

where $u = (\beta_1 d + \beta_2 d^2)$ and $v = \beta_3' d$. Similarly, if we try $\mathbf{p}_2 = (\underbrace{0,\dots,0}_{n-t_4},\underbrace{d'^{t_4},\dots,d'^2,d'}_{t_4})^{\top}$, then

$$A_n \mathbf{p}_2 = u'\mathbf{e}_1 + v'\mathbf{e}_n + \gamma d'^{t_4}\left(-d'\frac{\alpha}{\gamma}\mathbf{e}_{n-t_4+1} + \mathbf{e}_{n-t_4}\right), \tag{15}$$

where $v' = (\beta_1' d' + \beta_2' d'^2)$ and $u' = \beta_3 d'$.

Let

$$\mathbf{p} = s\mathbf{p}_1 + s'\mathbf{p}_2, \tag{16}$$

then by (14) and (15), we have

$$A_n \mathbf{p} = (su + s'u')\,\mathbf{e}_1 + (sv + s'v')\,\mathbf{e}_n + \mathbf{r}, \tag{17}$$

where $\mathbf{r} = s\alpha d^{t_3}(-d(\gamma/\alpha)\mathbf{e}_{t_3} + \mathbf{e}_{t_3+1}) + s'\gamma d'^{t_4}(-d'(\alpha/\gamma)\mathbf{e}_{n-t_4+1} + \mathbf{e}_{n-t_4})$.

Comparing with (13), we let $su + s'u' = w_1$ and $sv + s'v' = w_n$, and it follows that

$$\begin{aligned} s &= \frac{w_1 v' - w_n u'}{uv' - vu'}, \\ s' &= \frac{u w_n - v w_1}{uv' - vu'}, \\ A_n \mathbf{p} &= w_1 \mathbf{e}_1 + w_n \mathbf{e}_n + \mathbf{r}. \end{aligned} \tag{18}$$

After determining $\mathbf{p}$, the subroutine for computing $\mathbf{x}$ $(= \bar{\mathbf{x}} - \mathbf{p})$ is shown in Appendix 8, where the above concerning operations are the well-known prefix-product operations.

Furthermore, combining the vectorized subroutines described in Section 2.2.1, and the above subroutine shown in Appendix 8, our first vectorized algorithm for solving (1) is constituted by the following five subroutines.

```
CALL VLOW1(1,-d)
CALL UPDATELOW(1,-d,t1)
CALL VUPPER1(a,gamma)
CALL UPDATEUPPER(a,gamma,t2)
CALL FINAL(t3,t4)
```

Similarly, combining the vectorized subroutines described in Section 2.2.2, and the subroutine FINAL(t_3, t_4), our second vectorized algorithm for solving (1) is shown below.

```
CALL VLOW2(1,-d,k1)
CALL VUPPER2(a,gamma,k2)
CALL FINAL(t3,t4)
```

2.4. Error Analyses

2.4.1. For the first method

The following theorem gives the error analysis of our first vectorized algorithm for solving (1).

THEOREM 4. *Let* $c_4 = 1 + (|\beta_1| + |\beta_2| + |\beta_3|)c_3$, $c_5 = 1 + (|\beta_1'| + |\beta_2'| + |\beta_3'|)c_3$, $c_6 = (c_4|v'| + c_5|u'|)/(|uv' - vu'|)$, *and* $c_7 = (c_5|u| + c_4|v|)/(|uv' - vu'|)$, *we have*

$$\frac{\|A_n\mathbf{x} - \mathbf{b}\|}{\|\mathbf{b}\|} \le \left(c_1|d|^{t_1+1} + c_2\,|d'|^{t_2} + c_8|d|^{t_3} + c_9\,|d'|^{t_4}\right),$$

where $c_8 = c_6|\alpha|$ *and* $c_9 = c_7|\gamma|$.

PROOF. See Appendix 9.

For simplicity, by Theorem 4 we define the following notations: $t_1(\xi) = \min\{t \in N : c_1|d|^{t+1} < \xi/4\}$; $t_2(\xi) = \min\{t \in N : c_2|d'|^t < \xi/4\}$; $t_3(\xi) = \min\{t \in N : c_8|d|^t < \xi/4\}$; $t_4(\xi) = \min\{t \in N : c_9|d'|^t < \xi/4\}$, where ξ is the upper bound of the relative residual. If $\max(t_1(\xi), t_2(\xi)) > n/p$ or $\max(t_3(\xi), t_4(\xi)) > n$, then our first vectorized algorithm will break down when the required relative tolerance is sufficiently small and/or the diagonal dominance ratio is sufficiently close to 2.

However, for some cases, our first vectorized algorithm works well. For example, for conventional B-spline curve fitting [14,15], the corresponding system is of (1) with $\beta_1 = 5$, $\beta_2 = 1$, $\beta = 4$, $\beta_3 = 0$, $\alpha = 1$, $\gamma = 1$, $\beta_1' = 5$, $\beta_2' = 1$, and $\beta_3' = 0$. We have $t_1(\xi) = 11$, $t_2(\xi) = 12$, $t_3(\xi) = 13$, and $t_4(\xi) = 13$ for $\xi = 10^{-6}$; we have $t_1(\xi) = 13$, $t_2(\xi) = 14$, $t_3(\xi) = 15$, and $t_4(\xi) = 15$ for $\xi = 10^{-7}$. For closed B-spline curve fitting [14], the corresponding system is of (1) with $\beta_1 = 4$, $\beta_2 = 1$, $\beta = 4$, $\beta_3 = 1$, $\alpha = 1$, $\gamma = 1$, $\beta_1' = 4$, $\beta_2' = 1$, and $\beta_3' = 1$. We have $t_1(\xi) = 11$, $t_2(\xi) = 12$, $t_3(\xi) = 14$, and $t_4(\xi) = 14$ for $\xi = 10^{-6}$; we have $t_1(\xi) = 13$, $t_2(\xi) = 14$, $t_3(\xi) = 15$, and $t_4(\xi) = 15$ for $\xi = 10^{-7}$. Furthermore, let us examine the other two cases. If the system is of (1) with $\beta_1 = 3$, $\beta_2 = 1$, $\beta = 3$, $\beta_3 = 1$, $\alpha = 1$, $\gamma = 1$, $\beta_1' = 3$, $\beta_2' = 1$, and $\beta_3' = 1$, we have $t_1(\xi) = 16$, $t_2(\xi) = 17$, $t_3(\xi) = 19$, and $t_4(\xi) = 19$ for $\xi = 10^{-6}$. If the system with small diagonal dominance ratio is of (1) with $\beta_1 = 2.1$, $\beta_2 = 1$, $\beta = 2.1$, $\beta_3 = 1$, $\alpha = 1$, $\gamma = 1$, $\beta_1' = 2.1$, $\beta_2' = 1$, and $\beta_3' = 1$, we have $t_1(\xi) = 52$, $t_2(\xi) = 60$, $t_3(\xi) = 68$, and $t_4(\xi) = 68$ for $\xi = 10^{-6}$.

2.4.2. For the second method

In what follows, we will discuss the error analysis of our second vectorized algorithm for solving (1).

By Lemma 2 and (11), we have

$$|w_1| \le c_4' \|b\| \quad \text{and} \quad |w_n| \le c_5' \|b\|, \tag{19}$$

where $c_4' = 1 + (|\beta_1| + |\beta_2| + |\beta_3|)c_3'$ and $c_5' = 1 + (|\beta_1'| + |\beta_2'| + |\beta_3'|)c_3'$. By (18) and (19), we have

$$|s| \le c_6' \|\mathbf{b}\| \quad \text{and} \quad |s'| \le c_7' \|\mathbf{b}\|, \tag{20}$$

where $c_6' = (c_4'|v'| + c_5'|u'|)/(|uv' - vu'|)$ and $c_7' = (c_5'|u| + c_4'|v|)/(|uv' - vu'|)$. By the variation of (9a) and (20), we have the following result.

THEOREM 5.
$$\frac{\|A_n\mathbf{x} - \mathbf{b}\|}{\|\mathbf{b}\|} \le \left(|d|^{2^{k_1+1}} + c_2'\,|d'|^{2^{k_2+1}} + c_8'|d|^{t_3} + c_9'\,|d'|^{t_4}\right),$$

where $c_8' = c_6'|\alpha|$ and $c_9' = c_7'|\gamma|$.

For simplicity, by (20) we define the following notations: $t_1(\xi) = \min\{t \in N : |d|^{2^{t+1}} < \xi/4\}$; $t_2(\xi) = \min\{t \in N : c_2'|d'|^{2^{t+1}} < \xi/4\}$; $t_3(\xi) = \min\{t \in N : c_8'|d|^t < \xi/4\}$; $t_4(\xi) = \min\{t \in N : c_9'|d'|^t < \xi/4\}$, where ξ is the upper bound of the relative residual.

If $\max(t_1(\xi), t_2(\xi)) > n$ or $\max(t_3(\xi), t_4(\xi)) > n$, then our second vectorized algorithm will break down when the required relative tolerance is sufficiently small and/or the diagonal dominance ratio is sufficiently close to 2.

However, for some cases, our second vectorized algorithm works well. For the conventional B-spline curve fitting, we have $t_1(\xi) = 3$, $t_2(\xi) = 3$, $t_3(\xi) = 13$, and $t_4(\xi) = 13$ for $\xi = 10^{-6}$; we have $t_1(\xi) = 3$, $t_2(\xi) = 3$, $t_3(\xi) = 15$, and $t_4(\xi) = 15$ for $\xi = 10^{-7}$. For closed B-spline curve fitting, we have $t_1(\xi) = 3$, $t_2(\xi) = 3$, $t_3(\xi) = 13$, and $t_4(\xi) = 13$ for $\xi = 10^{-6}$; we have $t_1(\xi) = 3$, $t_2(\xi) = 3$, $t_3(\xi) = 15$, and $t_4(\xi) = 15$ for $\xi = 10^{-7}$. Furthermore, let us examine the previous two cases with diagonal dominance ratios 3/2 and 2.1/2, respectively. For the first case, we have $t_1(\xi) = 3$, $t_2(\xi) = 4$, $t_3(\xi) = 19$, and $t_4(\xi) = 19$ for $\xi = 10^{-6}$. For the second case, we have $t_1(\xi) = 5$, $t_2(\xi) = 5$, $t_3(\xi) = 65$, and $t_4(\xi) = 65$ for $\xi = 10^{-6}$.

3. EXPERIMENTAL RESULTS

The machine used in the numerical experiments is the CRAY X-MP EA/116se. The machine has a register-register architecture without the cache memory and has one vector processor which contains eight 64-bit vector registers of length 64. Memory is divided into 16 banks and each bank contains 1 M 64-bit words. Each bank requires 14 cycle time (one cycle time needs 8.5 nanoseconds) before it is ready for another request. The peak performance is 235 MFLOPS (millions of floating-point operations per second). In order to avoid memory conflicts for some m, the value of m (≥ 512) can be selected as $64(2z+1)$, where z is the smallest positive integer such that it satisfies that $64(2z+1) \geq n$. Here $m = 64(2\lceil n/128 \rceil + 1)$.

All of our testing data, **b**'s are generated by a random number generator, a function call **ranf()**, and each entry of **b** is ranged from 0 to 1. Our two vectorized algorithms are coded by CRAY Fortran 77 language. The operating system used here is UNICOS 6.1.6 and the compiler is called CF77.

Let $A_{i1} = A_i(2.5)$, $A_{i2} = A_i(2.7)$, and $A_{i3} = A_i(3)$ for $i = 1, 2$, where

$$A_1(\beta) = \begin{pmatrix} \beta & 1 & & & & \\ 1 & \beta & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & 1 & \beta & 1 \\ & & & & 1 & \beta \end{pmatrix}$$

and

$$A_2(\beta) = \begin{pmatrix} \beta & 1 & & & & 1 \\ 1 & \beta & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & 1 & \beta & 1 \\ 1 & & & & 1 & \beta \end{pmatrix}.$$

In addition, we let

$$A_{14} = \begin{pmatrix} 5 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 5 \end{pmatrix}$$

and

$$A_{24} = \begin{pmatrix} 5 & 1 & & & & 1 \\ 1 & 4 & 1 & & & \\ & \cdot & \cdot & \cdot & & \\ & & & 1 & 4 & 1 \\ 1 & & & & 1 & 5 \end{pmatrix},$$

which are derived from the conventional B-spline curve fitting problem and have been discussed in Section 2.4. When running our two algorithms on CRAY X-MP EA/116se, the performance is illustrated in Table 1 and Table 2, respectively.

Table 1. First algorithm's performance on CRAY X-MP EA/116se.

n	p	q	time	r_{11}	r_{12}	r_{13}	r_{14}	r_{21}	r_{22}	r_{23}	r_{24}
2048	64	33	0.18	10^{-8}	10^{-10}	10^{-11}	10^{-11}	10^{-8}	10^{-10}	10^{-11}	10^{-11}
4096	64	65	0.27	10^{-8}	10^{-10}	10^{-11}	10^{-11}	10^{-8}	10^{-10}	10^{-11}	10^{-11}
8192	64	129	0.45	10^{-8}	10^{-10}	10^{-11}	10^{-11}	10^{-8}	10^{-10}	10^{-11}	10^{-11}
16384	64	257	0.81	10^{-8}	10^{-10}	10^{-11}	10^{-11}	10^{-8}	10^{-10}	10^{-11}	10^{-11}

Table 2. Second algorithm's performance on CRAY X-MP EA/116se.

n	time	r_{11}	r_{12}	r_{13}	r_{14}	r_{21}	r_{22}	r_{23}	r_{24}
128	0.043	10^{-10}	10^{-12}	10^{-14}	10^{-14}	10^{-10}	10^{-12}	10^{-14}	10^{-14}
256	0.066	10^{-10}	10^{-12}	10^{-14}	10^{-14}	10^{-10}	10^{-12}	10^{-14}	10^{-14}
512	0.11	10^{-10}	10^{-12}	10^{-14}	10^{-14}	10^{-10}	10^{-12}	10^{-14}	10^{-14}
1024	0.20	10^{-10}	10^{-12}	10^{-14}	10^{-14}	10^{-10}	10^{-12}	10^{-14}	10^{-14}

In Tables 1 and 2, the symbols n, p, and $q \times q$ denote the size of $\mathbf{b}$, the number of the blocks, the size of one block. The symbol 'time' in terms of millisecond, represents the time spent in the first vectorized algorithm. The symbol r_{ij} for $1 \le i \le 2$ and $1 \le j \le 4$ represents the sup-norm of the relative residual corresponding to A_{ij}. In Table 1, by Theorem 4, we select $t_1 = t_2 = t_3 = t_4 = 30$; in Table 2, by Theorem 5, we select $k_1 = k_2 = 5$ and $t_3 = t_4 = 30$.

4. CONCLUSIONS

This paper has presented two fast vectorized algorithms for solving special tridiagonal systems and has analyzed the error analyses. Due to our matrix perturbation technique, all the vector operations involved in the two algorithms are scaled by a constant, which is very important for the efficient implementation on the CRAY X-MP. Some experimental results demonstrate the performance of our two vectorized algorithms. Our results can be applied to solve the quadratic B-spline curve fitting problem [16,17], the parabolic PDE [4] problem, and so on, since these problems belong to the type of special tridiagonal Toeplitz systems.

Using the same matrix perturbation method proposed in this paper, the parallel algorithms for solving (1) on hypercubes [18] and the B-spline surface fitting [19], respectively, have been developed. In addition, the results of this paper can also be applied to solve the diagonally dominant block tridiagonal system to achieve better performance. It is interesting to employ the other parallel tridiagonal solvers [20,21].

APPENDIX 1

```
      SUBROUTINE VLOW1(r,s)
      real r,s
C*****the entry array y represents vector y
C*****the exit array y represents vector x'
      do 5 i=0, p-1
        y(i*q+1)=(1/r)*y(i*q+1)
5     continue
      do 10  j=2,q
         do 20 i=0,p-1
            y(i*q+j)=(1/r)*(y(i*q+j)-s*y(i*q+j-1))
20       continue
10    continue
```

APPENDIX 2

```
      SUBROUTINE UPDATELOW1(r,s,t)
      real r,s
      integer t
C******the entry array y represents vector x'
C******the exit array y represents vector x
      temp=1
      do 30 j=0,t-1
         temp=temp*(-s/r)
```

```
        do 40 i=1,p-1
            y(i*q+j+1)=y(i*q+j+1)+y(iq)*temp
40      continue
30    continue
```

APPENDIX 3

Let $L^{-1}\mathbf{y} = \mathbf{x}$, then $L\mathbf{x} = \mathbf{y}$, i.e.,

$$rx_1 = y_1, \tag{3a}$$

$$sx_{i-1} + rx_i = y_i, \qquad i > 1. \tag{3b}$$

If $\|\mathbf{x}\| = |x_1|$, then by (3a), we have $\|\mathbf{y}\| \geq |y_1| = |rx_1| = |r|\,\|\mathbf{x}\|$. If $\|\mathbf{x}\| = |x_i|$ for some $i > 1$, then by (3b) and the triangular inequality, we have

$$\begin{aligned}\|\mathbf{y}\| \geq |y_i| &\geq |rx_i| - |sx_{i-1}| \\ &\geq |r|\,\|\mathbf{x}\| - |s|\,\|\mathbf{x}\| \\ &= (|r| - |s|)\|\mathbf{x}\|.\end{aligned}$$

We complete the proof and have $\|\mathbf{x}\| \leq (1/(|r| - |s|))\|\mathbf{y}\|$.

APPENDIX 4

Taking the triangular inequality on both sides of (6), since $|s/r| < 1$, it yields

$$\|\mathbf{x}\| \leq \left(1 + \left|\frac{s}{r}\right|\right) \|\mathbf{x}'\|. \tag{4a}$$

Applying Lemma 1 to (3), we have $\|\mathbf{x}'^{(i)}\| \leq 1/(|r| - |s|)\|\mathbf{y}^{(i)}\|$. Thus, we obtain

$$\begin{aligned}\|\mathbf{x}'\| &= \max_{1\leq i\leq p} \left\|\mathbf{x}'^{(i)}\right\| \\ &\leq \max_{1\leq i\leq p} \frac{1}{|r| - |s|} \left\|\mathbf{y}^{(i)}\right\| \\ &\leq \frac{1}{|r| - |s|}\|\mathbf{y}\|.\end{aligned}$$

Furthermore, by (7) we have

$$\|L_m\mathbf{x} - \mathbf{y}\| \leq |s| \left|\frac{s}{r}\right|^t \frac{1}{|r| - |s|}\|\mathbf{y}\|; \tag{4b}$$

by (4a), we have

$$\|\mathbf{x}\| \leq \frac{1 + |s/r|}{|r| - |s|}\|\mathbf{y}\|. \tag{4c}$$

By (4b), (4c), and their variations, the bound of the sup-norm of the residual vector $A'_m\mathbf{x}' - \mathbf{b}'$ is derived by

$$\begin{aligned}\|A'_m\mathbf{x}' - \mathbf{b}'\| &\leq \|L'_m(U'_m\mathbf{x}' - \mathbf{y}')\| + \|L'_m\mathbf{y}' - \mathbf{b}'\| \\ &\leq (1 + |d|)\,\|U'_m\mathbf{x}' - \mathbf{y}'\| + \|L'_m\mathbf{y}' - \mathbf{b}'\| \\ &\leq (1 + |d|)|\gamma| \left|\frac{\gamma}{a}\right|^{t_2} \frac{1}{|a| - |\gamma|}\,\|\mathbf{y}'\| + |d|^{t_1+1}\frac{1}{1 - |d|}\,\|\mathbf{b}'\| \\ &\leq \left((1 + |d|)|\gamma| \left|\frac{\gamma}{a}\right|^{t_2} \frac{1}{|a| - |\gamma|}\,\frac{1 + |d|}{1 - |d|} + |d|^{t_1+1}\frac{1}{1 - |d|}\right) \|\mathbf{b}\|.\end{aligned}$$

Let $c_1 = 1/(1-|d|)$ and $c_2 = (1+|d|)|\gamma|(1/(|a|-|\gamma|))((1+|d|)/(1-|d|))$, the above bound is simplified by

$$\|A'_m\mathbf{x}' - \mathbf{b}'\| \le \left(c_1|d|^{t_1+1} + c_2\left|\frac{\gamma}{a}\right|^{t_2}\right)\|\mathbf{b}\|. \tag{4d}$$

The term $c_1|d|^{t_1+1}$ will be less than ξ, if t_1 is greater than $(\log\xi - \log c_1)/\log|d| - 1$; the term $c_2|\gamma/a|^{t_2+1}$ will be less than ξ, if t_2 is greater than $(\log\xi - \log c_2)/(\log|\gamma/a|) - 1$.

By (4c) and its variation, we have

$$\begin{aligned}\|\mathbf{x}'\| &\le \frac{1+|\gamma/a|}{|a|-|\gamma|}\|\mathbf{y}'\| \\ &\le \frac{1+|\gamma/a|}{|a|-|\gamma|}\frac{1+|d|}{1-|d|}\|\mathbf{b}'\|.\end{aligned}$$

Let $c_3 = ((1+|\gamma/a|)/(|a|-|\gamma|))((1+|d|)/(1-|d|))$, then it yields

$$\|\mathbf{x}'\| \le c_3\|\mathbf{b}\|.$$

APPENDIX 5

```
      SUBROUTINE VLOW2(r,s,k)
      integer k
      real r,s
      p=1
      temp=-s/r
C*****Vector operation
      x[1:n]=(1/r)*x[1:n]
      do 10 i=0,k
C*****Compute (15)  in a vectorized way
         x[1+p:n]=x[1+p:n]+temp*x[1:n-p]
         temp=temp*temp
         p=p*2
10    continue
```

APPENDIX 6

```
      SUBROUTINE VUPPER2(r,s,k)
      integer k
      real r,s
      p=1
      temp=-s/r
      x[1:n]=(1/r)*x[1:n]
      do 10 i=0,k
         x[1:n-p]=x[1:n-p]+temp*x[1+p:n]
         temp=temp*temp
         p=p*2
10    continue
```

APPENDIX 7

By (8), (10), and their variations, the bound of the sup-norm of the residual vector $A'_m\mathbf{x}' - \mathbf{b}'$ isderived by

$$
\begin{aligned}
\|A'_m\mathbf{x}' - \mathbf{b}'\| &\le \|L'_m\,(U'_m\mathbf{x}' - \mathbf{y}')\| + \|L'_m\mathbf{y}' - \mathbf{b}'\| \\
&\le (1+|d|)\,\|U'_m\mathbf{x}' - \mathbf{y}'\| + \|L'_m\mathbf{y}' - \mathbf{b}'\| \\
&\le (1+|d|)\,|d'|^{2^{k_1+1}}\,\|\mathbf{y}'\| + |d|^{2^{k_2+1}}\,\|\mathbf{b}'\| \\
&\le (1+|d|)\,|d'|^{2^{k_1+1}}\,\frac{1}{1-|d|}\,\|\mathbf{b}'\| + |d|^{2^{k_2+1}}\,\|\mathbf{b}'\| \\
&\le \left((1+|d|)\,|d'|^{2^{k_1+1}}\,\frac{1}{1-|d|} + |d|^{2^{k_2+1}}\right)\|\mathbf{b}\|.
\end{aligned}
$$

Let $c'_1 = 1$ and $c'_2 = (1+|d|)/(1-|d|)$, then the bound is simplified by

$$
\|A'_m\mathbf{x}' - \mathbf{b}'\| \le \left(c'_1|d|^{2^{k_1+1}} + c'_2\,|d'|^{2^{k_2+1}}\right)\|\mathbf{b}\|. \tag{7a}
$$

The term $c'_1|d|^{2^{k_1+1}}$ will be less than ξ, if k_1 is greater than $\log(\log\xi/\log|d|) - 1$; the term $c'_2|d'|^{2^{k_2+1}}$ will be less than ξ, if k_2 is greater than $\log((\log\xi - \log c'_2)/\log|d|) - 1$.

From (10) and its variation, we have

$$
\begin{aligned}
\|\mathbf{x}'\| &\le \frac{1}{|a|-|\gamma|}\,\|\mathbf{y}'\| \\
&\le \frac{1}{|a|-|\gamma|}\,\frac{1}{1-|d|}\|\mathbf{b}\|.
\end{aligned}
$$

Let $c'_3 = (1/(|a|-|\gamma|))(1/(1-|d|))$, then the simplified bound is given by

$$
\|\mathbf{x}'\| \le c'_3\|\mathbf{b}\|. \tag{7b}
$$

APPENDIX 8

```
      SUBROUTINE FINAL(t3,t4)
      integer t3,t4
      temp=d
      do 10 i=1,t3
        x[i]=x[i]-temp*s
        temp=temp*d
10    continue
C*****dp represents d' in the context
      temp=dp
      do 20 i=n,n+1-t4
C*****sp represents s' in the context
        x[i]=x[i]-temp*sp
        temp=temp*dp
20    continue
```

APPENDIX 9

By $\mathbf{x} = \bar{\mathbf{x}} - \mathbf{p}$, (16), (17), and (18), we have

$$
\begin{aligned}
A_n\mathbf{x} - \mathbf{b} &= A_n\bar{\mathbf{x}} - \mathbf{b} - A_n\mathbf{p} \\
&= A_n\bar{\mathbf{x}} - \mathbf{b} - w_1\mathbf{e}_1 - w_n\mathbf{e}_n \\
&\quad - s\alpha d^{t_3}\left(-d\frac{\gamma}{\alpha}\mathbf{e}_{t_3} + \mathbf{e}_{t_3+1}\right) - s'\gamma d'^{t_4}\left(-d'\frac{\alpha}{\gamma}\mathbf{e}_{n-t_3+1} + \mathbf{e}_{n-t_4}\right).
\end{aligned}
$$

Immediately, it yields

$$
\begin{aligned}
\|A_n\mathbf{x} - \mathbf{b}\| &\le \|\mathbf{z}'\| + \left\|s\alpha d^{t_3}\left(-d\frac{\gamma}{\alpha}\mathbf{e}_{t_3} + \mathbf{e}_{t_3+1}\right) + s'\gamma d'^{t_4}\left(-d'\frac{\alpha}{\gamma}\mathbf{e}_{n-t_4+1} + \mathbf{e}_{n-t_4}\right)\right\| \\
&\le \left(c_1|d|^{t_1+1} + c_2\left|\frac{\gamma}{a}\right|^{t_2}\right)\|\mathbf{b}\| + \left|s\alpha d^{t_3}\right| + \left|s'\gamma d'^{t_4}\right|.
\end{aligned} \tag{9a}
$$

By (11) and Lemma 2, we have

$$|w_1| \leq c_4\|b\| \quad \text{and} \quad |w_n| \leq c_5\|b\|, \tag{9b}$$

where $c_4 = 1 + (|\beta_1| + |\beta_2| + |\beta_3|)c_3$ and $c_5 = 1 + (|\beta_1'| + |\beta_2'| + |\beta_3'|)c_3$.

By (18) and (9b), we have

$$|s| \leq c_6\|\mathbf{b}\| \quad \text{and} \quad |s'| \leq c_7\|\mathbf{b}\|, \tag{9c}$$

where $c_6 = (c_4|v'| + c_5|u'|)/(|uv' - vu'|)$ and $c_7 = (c_5|u| + c_4|v|)/(|uv' - vu'|)$. By (9a) and (9c), it follows that

$$\frac{\|A_n\mathbf{x} - \mathbf{b}\|}{\|\mathbf{b}\|} \leq \left(c_1|d|^{t_1+1} + c_2\left|d'\right|^{t_2} + c_8|d|^{t_3} + c_9\left|d'\right|^{t_4}\right), \tag{9d}$$

where $c_8 = c_6|\alpha|$ and $c_9 = c_7|\gamma|$.

REFERENCES

1. J.J. Dongarra, C.B. Moler, J.R. Bunch and G.W. Stewart, *LINPACK User's Guide*, SIAM Press, Pennsylvania, (1979).
2. D. Fischer, G. Golub, O. Hald, C. Levia and O. Widlund, On Fourier-Toeplitz methods for separable elliptic problems, *Mathematics of Computation* **28** (126), 349–368 (1974).
3. O.B. Widlund, On the use of fast methods for separable finite difference equations for the solution of general elliptic problems, In *Sparse Matrices and Their Applications*, (Edited by D.J. Rose and R.A. Willoughby), pp. 121–131, Plenum Press, New York, (1972).
4. G.D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, (1985).
5. R. Roberts and C.T. Mullis, *Digital Signal Processing*, Addison-Wesley, Reading, MA, (1987).
6. R.W. Hockney and C.R. Jesshope, *Parallel Computers 2*, Adam Hilger, Bristol, (1988).
7. R.F. Boisvert, Algorithms for special tridiagonal systems, *SIAM J. Sci. Stat. Comput.*, **12**, 423–442 (1991).
8. J. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York, (1988).
9. J.J. Dongarra, I.S. Duff, D.C. Sorensen and H.A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM Press, Pennsylvania, (1991).
10. G. Golub and J. Ortega, *Scientific Computing: An Introduction to Parallel Computing*, Academic Press, New York, (1993).
11. *Supercomputer Programming (I): Advanced Fortran: Architecture, Vectorization, and Parallel Computing*, Working manual for CRAY X-MP EA/116se, (1991).
12. O. Axelsson and V. Eijkhout, A note on the vectorization of scalar recursions, *Parallel Comput.* **3**, 73–83 (1986).
13. H.S. Stone, An efficient parallel algorithm for the solution of a tridiagonal linear system of equations, *J. ACM.* **20**, 27–38 (1973).
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, California, (1987).
15. K.L. Chung and L.J. Shen, Vectorized algorithm for B-spline curve fitting on CRAY X-MP EA/16se, Presented at the *IEEE Conf. on Supercomputing*, November 1992, pp. 166–169.
16. B. Pham, Quadratic B-splines for automatic curve and surface fitting, *Comput. & Graphics* **13**, 471–475 (1989).
17. K.L. Chung and W.M. Yan, Computing quadratic B-spline curve fitting on CRAY X-MP, In *Proc. National Computer Symposium*, Chiayi, R.O.C., December 1993, pp. 401–407.
18. K.L. Chung, W.M. Yan and J.G. Wu, A parallel algorithm for solving special tridiagonal systems on ring networks, *Computing* **56** (4), 385–395 (1996).
19. K.L. Chung and W.M. Yan, Parallel B-spline surface fitting on mesh-connected computers, *J. Parallel and Distributed Computing* **35**, 205–210 (1996).
20. R.W. Hockney, A fast direct solution of Poisson's equation using Fourier analysis, *J. ACM.* **12**, 95–113 (1965).
21. H.H. Wang, A parallel method for tridiagonal equations, *ACM Trans. Math. Software* **7** (2), 170–183 (1981).