A fast and parallel inertia finder for Toeplitz expanded matrices

Evgenij E. Tyrtyshnikov
Institute of Numerical Mathematics
Russian Academy of Sciences
Leninskij Prosp.32-A
Moscow 117334, Russia

August 9, 2000

Abstract

An algorithm is proposed to find the inertia of Toeplitz expanded matrices, i.e. those expressed as a sum of two-term products of Toeplitz triangular matrices. Using elementary eliminations the algorithm spends only $(m-1)n^2$ multiplications and as many additions, and can be performed through $O(n)$ parallel steps, where m is the number of summands and n is matrices order. As to memory, it requires only m vectors of size n and one vector of size m to be retained. For $m = 2$ an ameliorated version is suggested along with one possible systolic algorithm. Special pivoting is advocated for numerical stability. Presented are the results of the roundoff analysis, that can be interpreted as an evidence that actually computed values form the true LDL^T decomposition of some matrix which differs from A, the original matrix , by a matrix whose norm is proportional to the unit rounding error and a condition number of A.

1 Introduction

While studying the spectra of large symmetric matrices one may be interested in the eigenvalue distribution rather than individual eigenvalues by themselves. To this end it is obviously sufficient to find the inertia indices for a sequence of shifted matrices. In the paper, we propose a fast and parallel algorithm to determine the inertia for symmetric Toeplitz expanded matrices, i.e. those expressed as a sum of products of Toeplitz matrices.

So, suppose a real symmetric matrix A of order n is given such that

$$A = L_1 D_1 L_1^T + \ldots + L_m D_m L_m^T \tag{1.1}$$

where L_j is Toeplitz lower triangular and D_j is diagonal:

$$L_j = [l_{p-q+1,j}]_{p,q=1}^n, \quad l_{p-q+1,j} = 0 \quad if\ p-q \le -1; \tag{1.2}$$

$$D_j = diag(d_j, \ldots, d_j).$$

A notable example of Toeplitz expansion like (1.1) is the Gohberg-Sementsul formula [7] for the inverses of Toeplitz matrices. Toeplitz expansion properties were by and large examined in [5, 9]. In particular, (1.1) takes place for any symmetric matrix, though m may be comparable with n. In what follows we assume that m is arbitrary. However, actual efficiency can be talked about only when m is small in regard to n.

We will assume that A is strongly regular, that is, all its leading minors are distinct from zero. It means that A admits the LDL^T factorization (see, for example, [8]):

$$A = LDL^T \tag{1.3}$$

where L is lower triangular and D is diagonal. By (1.3) A and D are congruent, and thence have the same inertia.

Some existing algorithms can be easily tailored to obtain the signs of A's leading minors, and hence the inertia. One such algorithm is that from [9] developed to calculate the Toeplitz expansion for A^{-1}. The derivation of a similar algorithm through a general bordering scheme is given in [6]. These algorithms require $O(mn^2)$ arithmetic operations, but have poor parallel properties.

In order to design a parallel inertia finder in this paper we adapt a technique advanced in [3]. However, unlike [3] we do without circular and hyperbolic rotations, opting for elementary transformations like those in the Gauss elimination method. Numerical stability is supported here by pivoting. We suggest a simple means to check whether the process can be continued or should be halted. Should the latter occurs, we have to settle for the inertia computed for some leading submatrix of the original matrix. In our numerical tests there has been no case of delivering wrong inertia (for the matrix or submatrix) in any variant of termination.

Our algorithm take $(m-1)n^2$ multiplications and as many additions, which is less than it could be got when dealing with algorithms from [6,9]. More significant still, in contrast with previous algorithms it grows possible to have only $c(m)n$ parallel steps, where $c(m) = O(m)$ or even less in some appropriate modifications.

The LDL^T decomposition of matrix A, in principle, is available via presented here algorithms. Columns of L are computed successively, and once a column is found, it is never touched again. This makes it possible to involve but a small amount of active memory. Though, in the context of finding the inertia, the LDL^T decomposition by itself is not the goal of computation. As a consequence, overall computation can be implemented within the basic memory whose size is rather moderate: only m vectors of size n and one vector of size m are needed to be stored.

In section 2 a theory is evolved necessary to construct our algorithms. It is summed up by Theorem 2.1. This theorem establishes the existence of "simple" transformations which eventually result in the LDL^T decomposition when starting from the given Toeplitz expansion of a real symmetric strongly regular matrix A.

In section 3 a general scheme is formulated to carry out these "simple" transformations so as to dispense with "redundant" arithmetic and memory.

In section 4, a pruned down version of the general scheme is proposed (Algorithm 4.1) to specially handle Toeplitz matrices, and matrices (1.1) with $m = 2$. Also here described is a systolic array with very simple processor elements to implement Algorithm 4.1. The corresponding systolic algorithm is couched in the spirit of algorithmic descriptions from Chapter 6 of the book [8].

Section 5 is devoted to the analysis of rounding errors in Algorithm 4.1. The prime result can be interpreted as that showing that the actually computed LDL^T decomposition may differ from A by a matrix whose norm is proportional to the unit roundoff and a condition number of A. Finally, we demonstrate that pivoting is essential for numerical stability.

2 Theoretical background

Everything will be based on the next simple lemma.

Lemma 2.1 *Let P be a real nonsingular matrix of order mn, and suppose that*

$$[L_1, \ldots, L_m]P = [\hat{L}_1, \ldots, \hat{L}_m] \tag{2.1}$$

$$[L_1 D_1, \ldots, L_m D_m]P^{-T} = [\hat{L}_1 \hat{D}_1, \ldots, \hat{L}_m \hat{D}_m] \tag{2.2}$$

there $L_i, \hat{L}_i, D_i, \hat{D}_i \in \mathbf{R}^{n\times n}$. Then

$$L_1 D_1 L_1^T + \ldots + L_m D_m L_m^T = \hat{L}_1 \hat{D}_1 \hat{L}_1^T + \ldots + \hat{L}_m \hat{D}_m \hat{L}_m^T \tag{2.3}$$

Proof. The right -hand side of (2.3) equals

$$[\hat{L}_1, \ldots, \hat{L}_m][\hat{L}_1 \hat{D}_1, \ldots, \hat{L}_m \hat{D}_m]^T$$

while the left-hand side is

$$[L_1, \ldots, L_m][L_1 D_1, \ldots, L_m D_m]^T = ([L_1, \ldots, L_m]P)([L_1 D_1, \ldots, L_m D_m]P^{-T})^T. \quad \square$$

The main idea we are going to exploit consists in choosing P so as to have $\hat{L}_2 = \ldots = \hat{L}_m = 0$. Important is that P can be built up by piecemeal, that means that it will be expressed as a product of "simple" matrices. Specifically, if

$$P = P_1 \ldots P_N \tag{2.4}$$

then

$$P^{-T} = P_1^{-T} \ldots P_N^{-T} \tag{2.5}$$

Every P_i may account for obtaining new zeroes in $\hat{L}_2, \ldots, \hat{L}_m$ so that all preceding zeroes are kept unchanged.

Lemma 2.2 *Suppose real numbers p, q and vectors $u = [u_1 \ldots u_n]^T, v = [v_1 \ldots v_n]^T$ are given such that*

$$u_1 \neq 0, \tag{2.6}$$

$$u_1^2 p + v_1^2 q \neq 0. \tag{2.7}$$

Then there exists a nonsingular matrix

$$E = \begin{bmatrix} 1 & r_1 \\ r_2 & 1 \end{bmatrix} \tag{2.8}$$

such that

$$[u, v]E = [\hat{u}, \hat{v}], \tag{2.9}$$

$$[up, vq]E^{-T} = [\hat{u}\hat{p}, \hat{v}\hat{q}], \tag{2.10}$$

where $\hat{u} = [\hat{u}_1, \ldots, \hat{u}_n]^T, \hat{v} = [\hat{v}_1, \ldots, \hat{v}_n]^T$ and

$$\hat{v}_1 = 0; \tag{2.11}$$

$$\hat{p} = \frac{p}{g}, \hat{q} = \frac{q}{g}, g = 1 - r_1 r_2. \tag{2.12}$$

Proof. Due to (2.6), (2.7) $p \neq 0$, so set

$$r_1 = -\frac{v_1}{u_1}, \tag{2.13}$$

$$r_2 = (\frac{v_1}{u_1})(\frac{q}{p}). \tag{2.14}$$

Then

$$u_1 r_1 + v_1 = 0, \tag{2.15}$$

$$-u_1 p r_2 + v_1 q = 0. \tag{2.16}$$

At the same time,

$$g = det\ E = 1 - r_1 r_2 = 1 + \frac{v_1^2}{u_1^2}\frac{q}{p} = \frac{1}{u_1^2 p}(u_1^2 p + v_1^2 q) \neq 0.$$

According to (2.8) and (2.12) we have

$$E^{-T} = \frac{1}{g}\begin{bmatrix} 1 & -r_2 \\ -r_1 & 1 \end{bmatrix}. \tag{2.17}$$

Hence, (2.10) is equivalent to the matrix equality

$$\begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}\begin{bmatrix} 1 & -r_2 \\ -r_1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & r_1 \\ r_2 & 1 \end{bmatrix}\begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}, \tag{2.18}$$

which obviously emanates from (2.13), (2.14). □

From now on, by a simple matrix $P(k,l,E)$, used as P_i, will be meant a matrix which coincides with the identity matrix everywhere, except four positions $(k,k),(k,l),(l,k),(l,l), k < l$, housing 2×2 matrix E being either of the form (2.8) or the permutation matrix

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

If E is determined in accordance with Lemma 2.2 by values u_1, v_1, p, q then we will write $E = E(u_1, v_1, p, q)$.

Lemma 2.3 *Let A be a real symmetric strongly regular matrix of order n, given by Toeplitz expansion (1.1) where L_j, D_j are of the form (1.2). Then there is a product P of simple matrices, which obeys (2.1) and (2.2) where $\hat{D}_j$ are scalar matrices, $\hat{L}_j$ are Toeplitz lower triangular, and $\hat{L}_2, \ldots, \hat{L}_m$ have only zeroes along the main diagonal.*

Moreover, matrix

$$\tilde{A} \equiv \tilde{L}_1 \tilde{D}_1 \tilde{L}_1^T + \ldots + \tilde{L}_m \tilde{D}_m \tilde{L}_m^T \tag{2.19}$$

of order $n-1$ where

$$\tilde{L}_1 \equiv \begin{bmatrix} \hat{l}_{11} & & & 0 \\ \hat{l}_{21} & \hat{l}_{11} & & \\ \ldots & & \ddots & \\ \hat{l}_{n-1,1} & \hat{l}_{n-2,1} & \ldots & \hat{l}_{11} \end{bmatrix};$$

$$\tilde{L}_j \equiv \begin{bmatrix} \hat{l}_{2j} & & & 0 \\ \hat{l}_{3j} & \hat{l}_{2j} & & \\ \ldots & & \ddots & \\ \hat{l}_{n,j} & \hat{l}_{n-1,j} & \ldots & \hat{l}_{2j} \end{bmatrix}, j = 2, \ldots, m; \tag{2.20}$$

$$\tilde{D}_j \equiv diag(\hat{d}_j, \ldots, \hat{d}_j), \quad j = 1, \ldots, m,$$

is strongly regular.

Proof. Since A is strongly regular, we find

$$a_{11} = l_{11}^2 d_1 + \ldots + l_{1m}^2 d_m \neq 0. \tag{2.21}$$

If it were for all $1 \leq i < j \leq m$ that

$$l_{1i}^2 d_i + l_{1j}^2 d_j = 0, \tag{2.22}$$

then by summing these equations we would obtain

$$\sum_{1 \leq i < j \leq m} (l_{1i}^2 d_i + l_{1j}^2 d_j) = (m-1) a_{11}, \tag{2.23}$$

and therefore $a_{11} = 0$. Thus, it follows from the strong regularity that there exist indices i, j such that

$$l_{1i}^2 d_i + l_{1j}^2 d_j \neq 0. \tag{2.24}$$

Assume that $l_{1i} \neq 0$. Then in chime with Lemma 2.2 we may bring in matrix

$$E = E(l_{1i}, l_{1j}, d_i, d_j), \tag{2.25}$$

and take up

$$P_1 = \prod_{k=1}^{n} P((i-1)m + k, (j-1)m + k, E). \tag{2.26}$$

Set

$$[L_1^{(1)}, \ldots, L_m^{(1)}] \equiv [L_1, \ldots, L_m] P_1; \tag{2.27}$$

$$d_i^{(1)} = \frac{d_i}{det\ E}, d_j^{(2)} = \frac{d_j}{det\ E}; d_k^{(1)} = d_k, k \in \{1, \ldots, m\} \backslash \{i, j\}. \tag{2.28}$$

By Lemma 2.2

$$A = L_1^{(1)} D_1^{(1)} (L_1^{(1)})^T + \ldots + L_m^{(1)} D_m^{(1)} (L_m^{(1)})^T, \tag{2.29}$$

where

$$D_j^{(1)} = diag(d_j^{(1)}, \ldots, d_j^{(1)}).$$

By virtue of our definition of P_1 matrices $L_1^{(1)}, \ldots, L_m^{(1)}$ remain Toeplitz lower triangular, and L_j^1 thus acquires the zeroed main diagonal and will be ignored in subsequent transformations. Next, we again seek for indices i, j such that

$$l_{1i}^{(1)} \neq 0, (l_{1i}^{(1)})^2 d_i^{(1)} + l_{1j}^{(1)})^2 d_j^{(1)} \neq 0,$$

then construct the corresponding matrix E and define P_2 by (2.26). Clearly, it will take $t \leq m-1$ steps to arrive at

$$[\overline{L}_1, \ldots, \overline{L}_1] \equiv [L_1, \ldots, L_m] P_1 \ldots P_t,$$

where for some i $\overline{l}_{1i} \neq 0$, and for all $j \in \{1, \ldots, m\} \backslash \{i\}$ $\overline{l}_{1j} = 0$. If $i = 1$ then we have already achieved the new Toeplitz expansion of A we are after. If $i \neq 1$ then setting

$$P_{t+1} \equiv \prod_{k=1}^{n} P(k, (i-1)m + k, J), \tag{2.30}$$

we obtain

$$[\hat{L}_1, \ldots, \hat{L}_1] \equiv [L_1, \ldots, L_m] P_1 \ldots P_{t+1}, \tag{2.31}$$

where matrices $\hat{L}_1, \ldots, \hat{L}_1$ possess all desired properties, and each of $P_1, \ldots, P_{t+1}$ is a product of simple matrices.

It only remains to verify that $\tilde{A}$ of the form (2.19) is strongly regular. If we write down

$$A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{21} & & & \\ \cdots & & B & \\ a_{n1} & & & \end{bmatrix}, \tag{2.32}$$

then in accordance with (2.3) and (2.19)

$$B = \hat{d}_1 \begin{bmatrix} \hat{l}_{21} \\ \cdots \\ \hat{l}_{n1} \end{bmatrix} [\hat{l}_{21} \ldots \hat{l}_{n1}] + \tilde{A}. \tag{2.33}$$

Further,

$$\begin{bmatrix} 1 & & 0 \\ -\frac{a_{21}}{a_{11}} & & \\ \cdots & \ddots & \\ -\frac{a_{n1}}{a_{11}} & 0 & 1 \end{bmatrix} A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ 0 & & & \\ \cdots & & \hat{B} & \\ 0 & & & \end{bmatrix} \tag{2.34}$$

where

$$\hat{B} = B - \frac{1}{a_{11}} \begin{bmatrix} a_{21} \\ \cdots \\ a_{n1} \end{bmatrix} [a_{21} \ldots a_{n1}]. \tag{2.35}$$

On the strength of (2.34) it is clear that A's strong regularity entails strong regularity of $\hat{B}$. At the same time, by (2.19) the first column of A is of the form

$$[\hat{l}_{11} \ldots \hat{l}_{n1}]^T \hat{d}_1 \hat{l}_{11},$$

and hence

$$\frac{1}{a_{11}} \begin{bmatrix} a_{21} \\ \cdots \\ a_{n1} \end{bmatrix} [a_{21} \ldots a_{n1}] = d_1 \begin{bmatrix} \hat{l}_{21} \\ \cdots \\ \hat{l}_{n1} \end{bmatrix} \hat{l}_{21} \ldots \hat{l}_{n1}]. \tag{2.36}$$

Combining (2.33), (2.35), and (2.36) we conclude that $\tilde{A} = \tilde{B}$, and that completes the proof. □

Theorem 2.1 *Suppose A is a real symmetric strongly regular matrix of order n, given by Toeplitz expansion (1.1) where L_j, D_j are of the form (1.2). Then there exist simple matrices whose product P is such that (2.1) and (2.2) hold with $\hat{L}_2 = \ldots = \hat{L}_n = 0$, and $\hat{L}_1$ is lower triangular, $\hat{D}_1$ is diagonal.*

Proof. We will use the induction on n. Matrix $\tilde{A}$, obtained with the help of Lemma 2.3, enjoys all hypotheses of the theorem, but is of order $n-1$. Assume there is a product $\tilde{Q} = \tilde{Q}_1 \ldots \tilde{Q}_N$ of simple matrices of order $m(n-1)$, such that

$$[\tilde{L}_1, \ldots, \tilde{L}_m]\tilde{Q} = [\tilde{L}, 0, \ldots, 0], [\tilde{L}_1\tilde{D}_1, \ldots, \tilde{L}_m\tilde{D}_m]\tilde{Q}^{-T} = [\tilde{L}\tilde{D}, 0, \ldots, 0].$$

Denote by Q_j the matrix of order mn which is the same as the identity matrix everywhere, save for $m(n-1)$ positions $(k,l) \in M$,

$$M \equiv \{2, \ldots, km\} \setminus \bigcup_{i=2}^{m} \{in-1\},$$

housing $\tilde{Q}_j$. Matrices $Q_1, \ldots, Q_N$ are clearly simple. If $\overline{P}$ designates the product of simple matrices raised when applying Lemma 2.3 to A, then with $Q = Q_1 \ldots Q_N$

$$[L_1, \ldots, L_m]\overline{P}Q = [L, 0, \ldots, 0], [L_1 D_1, \ldots, L_m D_m]\overline{P}^{-T} Q^{-T} = [LD, 0, \ldots, 0].$$

where

$$L = \begin{bmatrix} \hat{l}_{11} & 0 & \cdots & 0 \\ \cdots & & & \\ \hat{l}_{n1} & & \tilde{L} & \end{bmatrix}, D = \begin{bmatrix} \hat{d}_1 & 0 \\ 0 & \tilde{D} \end{bmatrix}.$$

Here $\hat{l}_{11}, \ldots, \hat{l}_{n1}$ and $\hat{d}_1$ are values emerged in the course of application of Lemma 2.3 to A. The proof is completed. □

3 General scheme

We are now in a position to present our algorithm for finding inertia. Theorem 2.1, in fact, indicates a way to calculate all components of the LDL^T decomposition of A.Apparently, consecutive steps should be performed, each having the Toeplitz expansion of some new matrix of decreased by 1 order to be computed. We need not therefore compute and store all elements of dense matrices, and matrix operations described in the preceding section should be carried out, no doubt, implicitly. To determine inertia we need only signs of the entries of $D = diag(d_1, \ldots, d_n)$. Following Theorem 2.1 we can get $d_1, \ldots, d_n$ successively. To do this and avoid superfluous memory traffic we should erase those already found LDL^T components which would not be refered to in the sequel. Among other things, we incorporate in the algorithm some pivoting for the sake of numerical stability.

Thus, we introduce two-dimensional array $L(1:n, 1:m)$ and one-dimensional array $D(1:m)$. No other memory is needed.

Algorithm 3.1 *Given the components of the Toeplitz expansion (1.1), (1.2) of matrix $A \in \mathbf{R}^{n\times n}$ ($L(:,j)$ is the first column of L_j, and $D(j) = d_j, j = 1, \ldots, m$) suppose that $d_j \neq 0$ for all j. The algorithm computes the order n_u of the biggest strongly regular leading submatrix in A and the number n_e of negative eigenvalues for this submatrix.*

$n_e \leftarrow 0,\ n_u \leftarrow n$
FOR $k = 1 : n$
  $m_1 \leftarrow$ the number of nonzero components among $L(k, 1:m)$
  IF $m_1 > 1$ THEN
    FOR $j = 2, m_1$
      Find indices i_1, j_1 such that
      $|L(k, i_1)| \geq |L(k, j_1)| \geq 0$,
      $L(k, i_1), L(k, j_1), D(i_1), D(j_1) \neq 0$;
      if there is no such indices, then set $n_u \leftarrow k - 1$ and quit.
      $r_1 \leftarrow -L(k, j_1)/L(k, i_1)$
      $r_2 \leftarrow -r_1 D(j_1)/D(i_1)$
      $g \leftarrow 1 - r_1 r_2$
      $[L(k:n, i_1), L(k:n, j_1)] \leftarrow [L(k:n, i_1), L(k:n, j_1)] \begin{bmatrix} 1 & r_1 \\ r_2 & 1 \end{bmatrix}$
      $D(i_1) \leftarrow D(i_1)/g$
      $D(j_1) \leftarrow D(j_1)/g$
    END
  ENDIF

IF $(D(i_1) < 0)$ $n_e \leftarrow n_e + 1$
IF $k < n$ THEN
$U(k+1:n, i_1) \leftarrow U(k:n-1, i_1)$
ENDIF
END

The correctness of Algorithm 3.1 follows from Theorem 2.1. The arithmetic work does not exceed $2(m-1)n^2$, with equal parts for multiplications and additions. Algorithm 3.1 has a salient vectorized structure, and can be implemented through cn parallel steps, where $c = O(m)$. Obviously, there is an option to deal with independent pairs $(i_1, j_1), (i_1', j_1'), (i_1'', j_1'')$, and so on, allowing the concurrent treatment of corresponding columns.

The inequality

$$|L(k, i_1)| \geq |L(k, j_1)| \tag{3.1}$$

serves as a criterion for pivoting. Of course, the choice of i_1, j_1 may be subjected to some additional requirements.

4 Toeplitz matrix case

Algorithm 3.1 becomes especially elegant when A is Toeplitz, or when $m = 2$, that is,

$$A = d_1 L_1 L_1^T + d_2 L_2 L_2^T. \tag{4.1}$$

If d_1 and d_2 take on the same sign then A is sign-definite, and so the case is trivial from the inertia finding point. Therefore, later on we assume that signs of d_1 and d_2 differ. Without loss of generality let us agree that

$$d_2 = -d_1. \tag{4.2}$$

Lemma 4.1 *Suppose A is a real symmetric strongly regular matrix with Toeplitz expansion (4.1) which is subject to (4.2). Then on each step of Algorithm 3.1 the following is fulfilled:*

$$D(2) = -D(1); \tag{4.3}$$

$$r_2 = r_1; \tag{4.4}$$

$$|r_1| < 1, \quad 0 < g < 1; \tag{4.5}$$

$$|L(k,1)| \neq |L(k,2)|; \tag{4.6}$$

$$|L(k,1)| > |L(k,2)| \Rightarrow i_1 = 1, j_1 = 2; \tag{4.7}$$

$$|L(k,1)| < |L(k,2)| \Rightarrow i_1 = 2, j_1 = 1; \tag{4.8}$$

Proof. The inequality (4.3) is maintained via assignments $D(i_1) \leftarrow D(i_1)/g$, $D(j_1) \leftarrow D(j_1)/g$; (4.3) directly implies (4.4). Next, by our principle of pivoting $|r_1| \leq 1$; using (4.4) we can write $g = 1 - r_1^2$, and so $0 \leq g \leq 1$. With this,

$$g = 1 - \frac{L(k, j_1)^2}{L(k, i_1)^2},$$

and hence $g = 0$ is equivalent to $|L(k, i_1)| = |L(k, j_1)|$. We thus obtain (4.6) and, as a consequence, (4.5), (4.7), and (4.8). □

Algorithm 4.1 *Given the components of Toeplitz expansion (4.1) of matrix $A \in \mathbf{R}^{n\times n}$ with (4.2) being held, suppose that first columns of L_1 and L_2 reside in $L(1:n,1)$ and $L(1:n,2)$. Algorithm 4.1 computes the order n_u of the biggest strongly regular leading submatrix in A and the number of negative eigenvalues in this submatrix.*

$n_e \leftarrow 0, \quad n_u \leftarrow n$
FOR $k = 1 : n$
  IF $|L(k,1)| = |L(k,2)|$ THEN
    $n_u \leftarrow k-1$
    RETURN
  ENDIF
  IF $|L(k,1)| > |L(k,2)|$ THEN
    $i_k \leftarrow 1; \quad j_k \leftarrow 2$
  ELSE
    $i_k \leftarrow 2; \quad j_k \leftarrow 1; \quad n_e \leftarrow n_e + 1$
  ENDIF
  $r_k \leftarrow -L(k,j_k)/L(k,i_k)$
  $[L(k:n,i_k), L(k:n,j_k)] \leftarrow [L(k:n,i_k), L(k:n,j_k)] \begin{bmatrix} 1 & r_k \\ r_k & 1 \end{bmatrix}$
  IF $k < n$ THEN
    $L(k+1:n,i_k) \leftarrow L(k:n-1,i_k)$
  ENDIF
END

This algorithm requires n^2 multiplications and as many additions. It also calls for only $O(n)$ parallel steps. All needed memory is two vectors of order n.

Algorithm 4.1 has an entirely regular structure, and thus can be well executed in parallel. It can be easily implemented on diverse systolic arrays. Bellow described will be one such array consisting of fairly simple processor elements $P_1, \ldots, P_n$ of two kinds: C and B. Processor element P_1 is of kind C, and the others are of kind B.

All B-like elements must have memory to retain three real numbers and one logical value. Apart from some logic, each B-like element must perform the following action:

B: IF $logic = .TRUE.$ THEN
    $[u_{loc}, v_{loc}] \leftarrow [v_{loc}, u_{loc}]$
  ENDIF
  $[u_{loc}, v_{loc}] \leftarrow [u_{loc}, v_{loc}] \begin{bmatrix} 1 & r \\ r & 1 \end{bmatrix}$

while P_1 is prescribed to additionally store two integer numbers and execute the next statements:

C: IF $|u_{loc}| = |v_{loc}|$ THEN
    quit
  ELSE
    $n_u \leftarrow n_u + 1$
  ENDIF
  IF $|u_{loc}| > |v_{loc}|$ THEN
    $logic \leftarrow .FALSE.$
  ELSE
    $logic \leftarrow .TRUE.$
    $[u_{loc}, v_{loc}] \leftarrow [v_{loc}, u_{loc}]$
    $n_e \leftarrow n_e + 1$
  ENDIF

$r \leftarrow -\frac{v_{loc}}{u_{loc}}$, $u_{loc} \leftarrow u_{loc} + v_{loc} r$

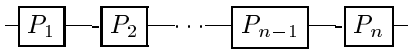
In terms of actions C and B, Algorithm 4.1 can be encapsulated by a graph. Below this is done for $n = 6$:

$$
\begin{array}{ccccccccccc}
C_1 & \rightarrow & B_2 & \rightarrow & B_3 & \rightarrow & B_4 & \rightarrow & B_5 & \rightarrow & B_6 \\
 & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow \\
 & & C_3 & \rightarrow & B_4 & \rightarrow & B_5 & \rightarrow & B_6 & \rightarrow & B_7 \\
 & & & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow \\
 & & & & C_5 & \rightarrow & B_6 & \rightarrow & B_7 & \rightarrow & B_8 \\
 & & & & & \searrow & \downarrow & \searrow & \downarrow & \searrow & \downarrow \\
 & & & & & & C_7 & \rightarrow & B_8 & \rightarrow & B_9 \\
 & & & & & & & \searrow & \downarrow & \searrow & \downarrow \\
 & & & & & & & & C_9 & \rightarrow & B_{10} \\
 & & & & & & & & & \searrow & \downarrow \\
 & & & & & & & & & & C_{11}
\end{array}
$$

Subscripts indicate the points in time at which the corresponding action starts running. To design a systolic array we, in effect, regard the projection of the above graph along the "main diagonal" direction :

$$C \rightleftharpoons B \rightleftharpoons B \rightleftharpoons B \rightleftharpoons B \rightleftharpoons B$$

Thus, processor elements will be aligned in a chain:



Let the global clock be set with ticks t. Then, pursuing the terminology and style of the book [8], we suggest the following

Algorithm 4.2 *Suppose processor elements $P_1, \ldots, P_n$ are connected to form a systolic chain. If each processor executes the following node program, then upon completion P_1 houses the order n_u of A's biggest strongly regular leading submatrix and the number n_e of its negative eigenvalues.*

loc. init.$[n, \mu = my.id, right, left,\ u_{loc} = L(\mu, 1),$
$v_{loc} = L(\mu, 2),$
$r = 0,$
$logic_{loc} = .FALSE.;$
if $\mu = 1$ then $\{n_e = 0, n_u = 0\}]$
FOR $t = 1 : 2n - 1$
IF $\mu = 1$ THEN
IF i is odd THEN
IF $t \neq 1$ RECEIVE$(v_{loc}, right)$
PERFORM ACTION C
ELSE
SEND$(\{r_{loc}, logic_{loc}, right\})$
ENDIF
ELSE
IF $\mu \leq t \leq 2n - \mu$ THEN
IF $\mu + t$ is even THEN
RECEIVE$(\{r_{loc}, logic_{loc}\}, left)$

```
                    IF (t > μ and μ ≠ n) RECEIVE(v_loc, right)
                    PERFORM ACTION B
                ELSE
                    SEND({r_loc, logic_loc}, right)
                    SEND(v_loc, left)
                ENDIF
            ENDIF
        ENDIF
    END
    quit
```

The understanding of how it works can be alleviated by observing the following chart which shows the functioning of processor elements with time ($n = 6$) :

t	P_1	P_2	P_3	P_4	P_5	P_6
1	C					
2		B				
3	C		B			
4		B		B		
5	C		B		B	
6		B		B		B
7	C		B		B	
8		B		B		
9	C		B			
10		B				
11	C					

Here the blank cells match the ticks when the corresponding processor is idle.

5 Roundoff analysis

Here we will try to assess how roundoffs affect Algorithm 4.1, the starting point being that the algorithm under study can be modified so as to be vied as that for computing the LDL^T decomposition of A. For definiteness, assume that $d_1 = 1$ and $d_2 = -1$, that is,

$$A = L_1 L_1^T - L_2 L_2^T. \tag{5.1}$$

It is more convenient to work with the matrix counterpart of Algorithm 4.1. So we set

$$L^{(0)} \equiv [L_1, L_2]. \tag{5.2}$$

Then, in harmony with the theory of Section 2, mapping the k-loop iterations onto matrix manipulations we have :

$$L^{(k)} = L^{(k-1)} P_k, \qquad k = 1, \ldots, n, \tag{5.3}$$

where

$$L^{(n)} = [\hat{L}, 0]. \tag{5.4}$$

Here, $\hat{L}$ is a lower triangular matrix whose k-th column captures the contents of one of the columns of array L before shifting at the end of k-th iteration :

$$\hat{L}(k : n, k) = L(k : n, i_k) \tag{5.5}$$

The following relationships are also valid in conjunction with (5.3) :

$$L^{(k)}D^{(k)} = L^{(k-1)}D^{(k-1)}P_k^{-T}, \quad k = 1, \ldots, n, \tag{5.6}$$

where $D^0 = diag(1, \ldots, 1, -1, \ldots, -1)$, and further

$$D^k = diag(d_1^{(k)}, \ldots, d_n^{(k)}, \overline{d}_1^{(k)}, \ldots, \overline{d}_n^{(k)}), \tag{5.7}$$

$$d_i^{(k)} = \begin{cases} d_i^{(k-1)} & , \ 1 \le i \le k-1, \\ d_k & , \ k \le i \le n; \end{cases} \tag{5.8}$$

$$d_k = (-1)^{i_k - 1} \prod_{l=1}^{k} \frac{1}{1 - r_l^2}, \quad k = 1, \ldots, n; \tag{5.9}$$

$$\overline{d}_i^{(k)} = \begin{cases} -d_k & , \ 1 \le k \le n-k+1, \\ d_i^{(k-1)} & , \ n-k+2 \le i \le n. \end{cases} \tag{5.10}$$

Thus,

$$A = \hat{L} diag(d_1, \ldots, d_n) \hat{L}^T. \tag{5.11}$$

Rounding errors result in that instead of r_k some other values $\tilde{r}_k$ will be obtained, and thus instead of $L^{(k)}$ and P_k some other $\tilde{L}^{(k)}$ and $\tilde{P}_k$ will come about, no longer satisfying (5.3). All the same , if we write

$$\tilde{L}^{(k)} = \tilde{L}^{(k-1)}\tilde{P}_k + F^{(k)}, \quad k = 1, \ldots, n, \tag{5.12}$$

then $F^{(k)}$'s elements will appear "sufficiently small".

Lemma 5.1 *Let η specify the unit roundoff. Then*

$$\|F^{(k)}\|_{1,\infty} \le \eta(1 + |\tilde{r}_k|)\|\tilde{L}^{(k-1)}\|_{1,\infty}. \tag{5.13}$$

Proof. Consider what happens with two $\tilde{L}^{(k-1)}$'s columns:

$$[\tilde{u}_{new}, \tilde{v}_{new}] = fl\left([\tilde{u}_{old}, \tilde{v}_{old}] \begin{bmatrix} 1 & \tilde{r}_k \\ \tilde{r}_k & 1 \end{bmatrix} \right).$$

This means that

$$\|fl(\tilde{u}_{old} + \tilde{v}_{old}\tilde{r}_k) - (\tilde{u}_{old} + \tilde{v}_{old}\tilde{r}_k)\|_1 \le \eta \ max(\|\tilde{u}_{old}\|_1, \|\tilde{v}_{old}\|_1)(1 + |\tilde{r}_k|),$$

$$\|fl(\tilde{u}_{old}\tilde{r}_k + \tilde{v}_{old}) - (\tilde{u}_{old}\tilde{r}_k + \tilde{v}_{old})\|_1 \le \eta \ max(\|\tilde{u}_{old}\|_1, \|\tilde{v}_{old}\|_1)(1 + |\tilde{r}_k|),$$

which leads to (5.13) as to the 1-norm case. The ∞-norm case is treated analogously. □

Corollary.

$$\|F^{(k)}\|_{1,\infty} \le \eta \prod_{l=1}^{k}(1 + |\tilde{r}_l|)\|L^{(0)}\|_{1,\infty} + O(\eta^2) \tag{5.14}$$

Lemma 5.2 *If*

$$\tilde{L}^{(n)} \equiv L^{(0)}\tilde{P}_1 \ldots \tilde{P}_n + F \tag{5.15}$$

then

$$\|F\|_{1,\infty} \le \eta n \prod_{l=1}^{n}(1 + |\tilde{r}_l|)\|L^{(0)}|_{1,\infty} + O(\eta^2) \tag{5.16}$$

Proof. From (5.12) and by the definition of F we find

$$F = F^{(1)}\tilde{P}_2 \dots \tilde{P}_n + F^{(2)}\tilde{P}_3 \dots \tilde{P}_n + \dots + F^{(n-1)}\tilde{P}_n + F^{(n)}. \tag{5.17}$$

Using that norms we deal with are submultiplicative we get

$$\|F^{(k)}\tilde{P}_{k+1} \dots \tilde{P}_n\|_{1,\infty} \le \|F^{(k)}\|_{1,\infty} \prod_{l=k+1}^{n} (1 + |\tilde{r}_l|), \tag{5.18}$$

and allowing for (5.14) arrive at (5.16).□

Next, set

$$\tilde{d}_k \equiv (-1)^{i_k - 1} \prod_{l=1}^{k} \frac{1}{1 - \tilde{r}_l^2}, \quad k = 1, \dots, n, \tag{5.19}$$

and define diagonal matrices $\tilde{D}^{(k)}$ by formulas similar to (5.7)-(5.10) but with (5.14) replacing (5.9).

Lemma 5.3 *If*

$$\tilde{L}^{(n)}\tilde{D}^{(n)} = L^{(0)}D^{(0)}\tilde{P}_1^{-T} \dots \tilde{P}_n^{-T} + G, \tag{5.20}$$

then

$$\|G\|_{1,\infty} \le \eta n \prod_{l=1}^{n} \frac{1}{1 - |\tilde{r}_l|} \|L^{(0)}\|_{1,\infty} + O(\eta^2) \tag{5.21}$$

Proof. First of all, note that equalities (5.12) entail

$$\tilde{L}^{(k)}\tilde{D}^{(k)} = \tilde{L}^{(k-1)}\tilde{D}^{(k-1)}\tilde{P}_k^{-T} + F^{(k)}\tilde{D}^{(k)}; \tag{5.22}$$

to this end, it is sufficient to ascertain what k-th iteration does with any two columns. Further, by (5.20) and (5.22)

$$G = F^{(1)}\tilde{D}^{(1)}\tilde{P}_2^{-T} \dots \tilde{P}_n^{-T} + F^{(2)}\tilde{D}^{(2)}\tilde{P}_3^{-T} \dots \tilde{P}_n^{-T} + \dots + F^{(n-1)}\tilde{D}^{(n-1)}\tilde{P}_{n-1}^{-T} + F^{(n)}\tilde{D}^{(n)}. \tag{5.23}$$

It now remains to resort to (5.14), (5.15), and use that the norms are submultiplicative. □

Theorem 5.1 *Let $\tilde{L}$ be a matrix actually computed through Algorithm 4.1, $\tilde{D} = diag(\tilde{d}_1, \dots, \tilde{d}_n)$. Then*

$$\|A - \tilde{L}\tilde{D}\tilde{L}^T\|_{1,\infty} \le 2\eta n \prod_{l=1}^{n} \frac{1 + |\tilde{r}_l|}{1 - |\tilde{r}_l|} \|L^{(0)}\|_1 \|L^{(0)}\|_\infty + O(\eta^2) \tag{5.24}$$

Proof. Setting $\tilde{P} \equiv \tilde{P}_1 \dots \tilde{P}_n$ we obtain

$$[\tilde{L}, 0] = L^0\tilde{P} + F,$$

$$[\tilde{L}\tilde{D}, 0] = L^0 D^0 \tilde{P}^{-T} + G,$$

and it thence follows that

$$\tilde{L}\tilde{D}\tilde{L}^T = (L^{(0)}\tilde{P}+F)(L^{(0)}D^{(0)}\tilde{P}^{-T}+G)^T = L^{(0)}D^{(0)}(L^{(0)})^T + L^{(0)}\tilde{P}G^T + F\tilde{P}^{-1}D^{(0)}(L^{(0)})^T + FG^T$$

Obviously, $A = L^{(0)}D^{(0)}(L^{(0)})^T$, and applying Lemmas 5.2 and 5.3 will do the proof.□

The appearance of estimate (5.24) resembles that of the estimate derived in [4] for the Levinson-Durbin algorithm; also cf.[2]. Note that the value

$$r(A) \equiv \prod_{l=1}^{n} \frac{1+|r_l|}{1-|r_l|}$$

can be interpreted as an estimate of $\|A^{-1}\|$. At the other hand, $\|L^{(0)}\|_1 \|L^{(0)}\|_\infty$ have something to do with $\|A\|$. Thus, there is a ground to consider the bound of Theorem 5.1, to some extent, as an evidence that Algorithm 4.1 computes the true LDL^T decomposition of some matrix which differs from A so that the deviation by norm is proportional to η and a condition number of A. Of course, we should mention that the last statement in our exposition goes without *a* rigorous proof.

According to Lemma 4.1 $|\tilde{r}_k| < 1, \quad k = 1, \ldots, n$. These inequalities follow from our principle of pivoting. We would like to stress that without pivoting the halting may come about rather quickly before end. For instance, if

$$A = \begin{bmatrix} 0.1 & 0 \\ 1 & 0.1 \end{bmatrix} \begin{bmatrix} 0.1 & 1 \\ 0 & 0.1 \end{bmatrix} - \begin{bmatrix} 10 & 0 \\ 1 & 10 \end{bmatrix} \begin{bmatrix} 10 & 1 \\ 0 & 10 \end{bmatrix}$$

then array L looks like

$$L = \begin{bmatrix} 0.1 & 10 \\ 1 & 1 \end{bmatrix}$$

Assume that we have a chopped 10-base arithmetic with 3 digit mantissa. If there is no pivoting, the first step affords

$$\tilde{r} = -100; \ \tilde{l}_{21} = fl(0.1 - 10 \cdot 100) = -1000; \tilde{l}_{22} = fl(1 - 10 \cdot 100) = -1000;$$

and so the procedure stalls because of $\tilde{l}_{21} = \tilde{l}_{22}$. At the same time, Algorithm 4.1 works successfully due to pivoting.

References

[1] R. P. Brent and F. T. Luk, *A systolic array for the linear-time solution of Toeplitz systems of equations*, J.VLSI and Comput. Syst. 1,no.1: 1-22 (1983).

[2] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J.Sci.Stat.Comput. 6, no.2: 349-364 (1985).

[3] J. Chun, T. Kailath, H. Lev-Ari, *Fast parallel algorithm for QR and triangular factorization*, SIAM J.Sci.Stat.Comput., 8, no.6: 899-913 (1987).

[4] G. Cybenko, *The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Stat. Comput. 1, no. 3: 303-319 (1980).

[5] B. Friedlander, M. Morf, T. Kailath, L. Ljung, *New inversion formulas for matrices classified in term of their distance from Toeplitz matrices*, Linear Algebra Appl. 27: 31-60 (1979).

[6] I. Gohberg, T. Kailath, I. Koltracht, *Efficient solution of linear systems of equations with recursive structure*, Linear Algebra Appl. 80: 80-113 (1986).

[7] I. Gohberg and A. Sementsul, *On the inversion of finite Toeplitz matrices and their continuous analogs*, Mat. Issled. 2: 201-233 (1972) (in Russian).

[8] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2-nd ed. The John Hopkins Univ. Press, Berkeley (1989).

[9] T. Kailath, S. Y. Kung, M. Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl. 68, no. 2: 395-407 (1979).

[10] S. A. Krasnov and E. E. Tyrtyshnikov, *Vectorized algorithms and systolic arrays for Toeplitz systems of equations*, Sov. J. Numer. Anal. Math. Modelling 2, no. 2: 83-158 (1987).

[11] E. E. Tyrtyshnikov, *Toeplitz Matrices, Some of their analogs and Applications*, Dept. Numer. Math., USSR Acad. of Sci., Moscow (1989) (in Russian).