On the stability of the Bareiss and related Toeplitz factorization algorithms*

A. W. Bojanczyk
School of Electrical Engineering
Cornell University
Ithaca, NY 14853-5401, USA
adamb@toeplitz.ee.cornell.edu

R. P. Brent
Computer Sciences Laboratory
Australian National University
Canberra, ACT 0200, Australia
rpb@cslab.anu.edu.au

F. R. de Hoog
Division of Maths. and Stats.
CSIRO
Canberra, ACT 2601, Australia
frank@dmscanb.cbr.dms.csiro.au

D. R. Sweet
Electronics Research Laboratory
DSTO
Salisbury, SA 5108, Australia
dougs@ewd.dsto.gov.au

Report TR-CS-93-14
November 1993

Abstract

This paper contains a numerical stability analysis of factorization algorithms for computing the Cholesky decomposition of symmetric positive definite matrices of displacement rank 2. The algorithms in the class can be expressed as sequences of *elementary downdating* steps. The stability of the factorization algorithms follows directly from the numerical properties of algorithms for realizing elementary downdating operations. It is shown that the Bareiss algorithm for factorizing a symmetric positive definite Toeplitz matrix is in the class and hence the Bareiss algorithm is stable. Some numerical experiments that compare behavior of the Bareiss algorithm and the Levinson algorithm are presented. These experiments indicate that in general (when the reflection coefficients are not all positive) the Levinson algorithm can give much larger residuals than the Bareiss algorithm.

1 Introduction

We consider the numerical stability of algorithms for solving a linear system

$$Tx = b, \tag{1.1}$$

where T is an $n \times n$ positive definite Toeplitz matrix and b is an $n \times 1$ vector. We assume that the system is solved in floating point arithmetic with relative precision ϵ by first computing the Cholesky factor of T. Hence the emphasis of the paper is on factorization algorithms for the matrix T.

Roundoff error analyses of Toeplitz systems solvers have been given by Cybenko [10] and Sweet [22]. Cybenko showed that the Levinson-Durbin algorithm produces a residual which, under the condition that all reflection coefficients are positive, is of comparable size to that produced by the well behaved Cholesky method. He hypothesised that the same is true even if the reflection coefficients are not all positive. If correct, this would indicate that numerical quality of the Levinson-Durbin algorithm is comparable to that of the Cholesky method.

In his PhD thesis [22], Sweet presented a roundoff error analysis of a variant of the Bareiss algorithm [2], and concluded that the algorithm is numerically stable (in the sense specified in Section 7). In this paper we strengthen and generalize these early results on the stability of the Bareiss algorithm. In particular, our approach via elementary downdating greatly simplifies roundoff error analysis and makes it applicable to a larger-than-Toeplitz class of matrices.

After introducing the notation and the concept of *elementary downdating* in Sections 2 and 3, in Section 4 we derive matrix factorization algorithms as a sequence of elementary downdating operations (see also [4]). In Section 5 we present a first order analysis by bounding the first term in an asymptotic expansion for the error in powers of ϵ. By analyzing the propagation of first order error in the sequence of downdatings that define the algorithms, we obtain bounds on the perturbations of the factors in the decompositions. We show that the computed upper triangular factor $\tilde{U}$ of a positive definite Toeplitz matrix T satisfies

$$T = \tilde{U}^T\tilde{U} + \Delta T \ , \quad ||\Delta T|| \le c(n)\epsilon||T|| \ ,$$

where $c(n)$ is a low order polynomial in n and is independent of the condition number of T. Many of the results of Sections 2–5 were first reported in [5], which also contains some results on the stability of Levinson's algorithm.

In Section 6 we discuss the connection with the Bareiss algorithm and conclude that the Bareiss algorithm is stable for the class of symmetric positive definite matrices. Finally, in Section 7 we report some interesting numerical examples that contrast the behaviour of the Bareiss algorithm with that of the Levinson algorithm. We show numerically that, in cases where the reflection coefficients are not all positive, the Levinson algorithm can give much larger residuals than the Bareiss or Cholesky algorithms.

2 Notation

Unless it is clear from the context, all vectors are real and of dimension n. Likewise, all matrices are real and their default dimension is $n \times n$. If $\mathbf{a} \in \Re^n$, $||\mathbf{a}||$ denotes the usual Euclidean norm, and if $T \in \Re^{n\times n}$, $||T||$ denotes the induced matrix norm:

$$||T|| = \max_{||\mathbf{a}||=1} ||T\mathbf{a}|| \ .$$

Our primary interest is in a symmetric positive definite Toeplitz matrix T whose i,jth entry is

$$t_{ij} = t_{|i-j|}.$$

We denote by $\mathbf{e}_k$, $k = 1, \ldots, n$, the unit vector whose kth element is 1 and whose other elements are 0. We use the following special matrices:

$$Z \equiv \sum_{i=1}^{n-1} e_{i+1}e_i^T = \begin{pmatrix} 0 & & \cdots & \cdots & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix} ,$$

$$J \equiv \sum_{i=1}^{n} e_{n-i+1} e_i^T = \begin{pmatrix} 0 & \cdots & \cdots & 0 & 1 \\ \vdots & & \cdot & 1 & 0 \\ \vdots & \cdot & \cdot & \cdot & \vdots \\ 0 & 1 & \cdot & & \vdots \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix} .$$

The matrix Z is known as a *shift-down* matrix. We also make use of powers of the matrix Z, for which we introduce the following notation:

$$Z_k = \begin{cases} I & \text{if } k = 0, \\ Z^k & \text{if } k > 0. \end{cases}$$

The antidiagonal matrix J is called a *reversal* matrix, because the effect of applying J to a vector is to reverse the order of components of the vector:

$$J \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_1 \end{bmatrix} .$$

The *hyperbolic rotation* matrix $H(\theta) \in \Re^{2\times 2}$ is defined by

$$H(\theta) = \frac{1}{\cos\theta} \begin{bmatrix} 1 & -\sin\theta \\ -\sin\theta & 1 \end{bmatrix} . \tag{2.1}$$

The matrix $H(\theta)$ satisfies the relation

$$H(\theta) \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} H(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} ,$$

and it has eigenvalues $\lambda_1(\theta)$, $\lambda_2(\theta)$ given by

$$\lambda_1(\theta) = \lambda_2^{-1}(\theta) = \sec\theta - \tan\theta. \tag{2.2}$$

For a given pair of real numbers a and b with $|a| > |b|$, there exists a hyperbolic rotation matrix $H(\theta)$ such that

$$H(\theta) \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{a^2 - b^2} \\ 0 \end{bmatrix} . \tag{2.3}$$

The angle of rotation θ is determined by

$$\sin\theta = \frac{b}{a} \quad , \quad \cos\theta = \frac{\sqrt{a^2 - b^2}}{a} . \tag{2.4}$$

3 Elementary Downdating

In this section we introduce the concept of elementary downdating. The elementary downdating problem is a special case of a more general downdating problem that arises in Cholesky factorization of a positive definite difference of two outer product matrices [1, 6, 7, 12]. In Section 4, factorization algorithms are derived in terms of a sequence of downdating steps. The numerical properties of the algorithms are then related to the properties of the sequence of elementary downdating steps.

Let $\mathbf{u}_k, \mathbf{v}_k \in \Re^n$ have the following form:

$$\begin{array}{cccccccccc}
 & & & & & k & & & & \\
 & & & & & \downarrow & & & & \\
\mathbf{u}_k^T & = & [0 & \ldots & 0 & \times & \times & \times & \ldots & \times\,], \\
\mathbf{v}_k^T & = & [0 & \ldots & 0 & 0 & \times & \times & \ldots & \times\,], \\
 & & & & & & \uparrow & & & \\
 & & & & & & k+1 & & &
\end{array}$$

that is:

$$\mathbf{e}_j^T \mathbf{u}_k = 0 \ , \ j < k \ , \quad \text{and} \quad \mathbf{e}_j^T \mathbf{v}_k = 0 \ , \ j \le k \ .$$

Applying the shift-down matrix Z to $\mathbf{u}_k$, we have

$$\begin{array}{ccccccccccc}
 & & & & & & k+1 & & & \\
 & & & & & & \downarrow & & & \\
\mathbf{u}_k^T Z^T & = & [0 & \ldots & 0 & 0 & \times & \times & \ldots & \times], \\
\mathbf{v}_k^T & = & [0 & \ldots & 0 & 0 & \times & \times & \ldots & \times]. \\
 & & & & & & \uparrow & & & \\
 & & & & & & k+1 & & &
\end{array}$$

Suppose that we wish to find $\mathbf{u}_{k+1}, \mathbf{v}_{k+1} \in \Re^n$ to satisfy

$$\mathbf{u}_{k+1}\mathbf{u}_{k+1}^T - \mathbf{v}_{k+1}\mathbf{v}_{k+1}^T = Z\mathbf{u}_k\mathbf{u}_k^T Z^T - \mathbf{v}_k\mathbf{v}_k^T , \tag{3.1}$$

where

$$\begin{array}{cccccccccc}
 & & & & & & k+1 & & & \\
 & & & & & & \downarrow & & & \\
\mathbf{u}_{k+1}^T & = & [0 & \ldots & 0 & 0 & \times & \times & \ldots & \times], \\
\mathbf{v}_{k+1}^T & = & [0 & \ldots & 0 & 0 & 0 & \times & \ldots & \times], \\
 & & & & & & & \uparrow & & \\
 & & & & & & & k+2 & &
\end{array}$$

that is

$$\mathbf{e}_j^T \mathbf{u}_{k+1} = 0 \ , \ j < k+1 \ , \quad \text{and} \quad \mathbf{e}_j^T \mathbf{v}_{k+1} = 0 \ , \ j \le k+1 \ . \tag{3.2}$$

We refer to the problem of finding $\mathbf{u}_{k+1}$ and $\mathbf{v}_{k+1}$ to satisfy (3.1), given $\mathbf{u}_k$ and $\mathbf{v}_k$, as the *elementary downdating* problem. It can be rewritten as follows:

$$[\mathbf{u}_{k+1}\ \mathbf{v}_{k+1}] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{k+1}^T \\ \mathbf{v}_{k+1}^T \end{bmatrix} = [Z\mathbf{u}_k\ \mathbf{v}_k] \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \mathbf{u}_k^T Z^T \\ \mathbf{v}_k^T \end{bmatrix} .$$

From (2.1), (2.3) and (2.4), it is clear that the vectors $\mathbf{u}_{k+1}$ and $\mathbf{v}_{k+1}$ can be found by using a hyperbolic rotation $H(\theta_k)$ defined by the following relations:

$$\sin\theta_k = \mathbf{e}_{k+1}^T \mathbf{v}_k / \mathbf{e}_k^T \mathbf{u}_k \ , \tag{3.3a}$$

$$\cos\theta_k = \sqrt{1 - \sin^2\theta_k} \ , \tag{3.3b}$$

and

$$\begin{bmatrix} \mathbf{u}_{k+1}^T \\ \mathbf{v}_{k+1}^T \end{bmatrix} = H(\theta_k) \begin{bmatrix} \mathbf{u}_k^T Z^T \\ \mathbf{v}_k^T \end{bmatrix} . \tag{3.4}$$

The elementary downdating problem has a unique solution (up to obvious sign changes) if

$$|\mathbf{e}_k^T \mathbf{u}_k| > |\mathbf{e}_{k+1}^T \mathbf{v}_k| \ .$$

The calculation of $\mathbf{u}_{k+1}$, $\mathbf{v}_{k+1}$ via (3.4) can be performed in the obvious manner. Following common usage, algorithms which perform downdating in this manner will be referred to as *hyperbolic* downdating algorithms.

Some computational advantages may be obtained by rewriting (3.1) as follows:

$$[\mathbf{u}_{k+1}\ \mathbf{v}_k]\begin{bmatrix} \mathbf{u}_{k+1}^T \\ \mathbf{v}_k^T \end{bmatrix} = [Z\mathbf{u}_k\ \mathbf{v}_{k+1}]\begin{bmatrix} \mathbf{u}_k^T Z^T \\ \mathbf{v}_{k+1}^T \end{bmatrix} .$$

Consider now an orthogonal rotation matrix $G(\theta_k)$,

$$G(\theta_k) = \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{bmatrix} ,$$

where $\cos\theta_k$ and $\sin\theta_k$ are defined by (3.3b) and (3.3a), respectively. Then it is easy to check that

$$G(\theta_k)\begin{bmatrix} \mathbf{u}_{k+1}^T \\ \mathbf{v}_k^T \end{bmatrix} = \begin{bmatrix} \mathbf{u}_k^T Z^T \\ \mathbf{v}_{k+1}^T \end{bmatrix} , \tag{3.5}$$

or, equivalently,

$$\begin{bmatrix} \mathbf{u}_{k+1}^T \\ \mathbf{v}_k^T \end{bmatrix} = G(\theta_k)^T \begin{bmatrix} \mathbf{u}_k^T Z^T \\ \mathbf{v}_{k+1}^T \end{bmatrix} . \tag{3.6}$$

Thus, we may rewrite (3.6) as

$$\mathbf{v}_{k+1} = (\mathbf{v}_k - \sin\theta_k Z\mathbf{u}_k)/\cos\theta_k \ , \tag{3.7a}$$
$$\mathbf{u}_{k+1} = -\sin\theta_k \mathbf{v}_{k+1} + \cos\theta_k Z\mathbf{u}_k \ . \tag{3.7b}$$

Note that equation (3.7a) is the same as the second component of (3.4). However, (3.7b) differs from the first component of (3.4) as it uses $\mathbf{v}_{k+1}$ in place of $\mathbf{v}_k$ to define $\mathbf{u}_{k+1}$. It is possible to construct an alternative algorithm by using the first component of (3.5) to define $\mathbf{u}_{k+1}$. This leads to the following formulas:

$$\mathbf{u}_{k+1} = (Z\mathbf{u}_k - \sin\theta_k \mathbf{v}_k)/\cos\theta_k \ , \tag{3.8a}$$
$$\mathbf{v}_{k+1} = -\sin\theta_k \mathbf{u}_{k+1} + \cos\theta_k \mathbf{v}_k \ . \tag{3.8b}$$

We call algorithms based on (3.7a)–(3.7b) or (3.8a)–(3.8b) *mixed* elementary downdating algorithms. The reason for considering mixed algorithms is that they have superior stability properties to hyperbolic algorithms in the following sense.

Let $\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{v}}_k$ be the values of $\mathbf{u}_k$, $\mathbf{v}_k$ that are computed in floating point arithmetic with relative machine precision ϵ. The computed values $\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{v}}_k$ satisfy a perturbed version of (3.1), that is,

$$\tilde{\mathbf{u}}_{k+1}\tilde{\mathbf{u}}_{k+1}^T - \tilde{\mathbf{v}}_{k+1}\tilde{\mathbf{v}}_{k+1}^T = Z\tilde{\mathbf{u}}_k\tilde{\mathbf{u}}_k^T Z^T - \tilde{\mathbf{v}}_k\tilde{\mathbf{v}}_k^T + \epsilon G_k + O(\epsilon^2) \ , \tag{3.9}$$

where the second order term $O(\epsilon^2)$ should be understood as a matrix whose elements are bounded by a constant multiple of $\epsilon^2||G_k||$. The norm of the perturbation G_k depends on the precise specification of the algorithm used. It can be shown [6] that the term G_k satisfies

$$\|G_k\| \le c_m \left(\|Z\mathbf{u}_k\|^2 + \|\mathbf{v}_k\|^2 + \|\mathbf{u}_{k+1}\|^2 + \|\mathbf{v}_{k+1}\|^2\right) \tag{3.10}$$

when a mixed downdating strategy is used (here c_m is a positive constant). When hyperbolic downdating is used the term G_k satisfies

$$\|G_k\| \le c_h \|H(\theta_k)\| \left(\|Z\mathbf{u}_k\| + \|\mathbf{v}_k\|\right)\left(\|\mathbf{u}_{k+1}\| + \|\mathbf{v}_{k+1}\|\right) \ , \tag{3.11}$$

where c_h is a positive constant [6]. (The constants c_m and c_h are dependent on implementation details, but are of order unity and independent of n.) Note the presence of the multiplier $\|H(\theta_k)\|$ in the bound (3.11) but not in (3.10). In view of (2.2), $\|H(\theta_k)\|$ could be large. The significance of the multiplier $\|H(\theta_k)\|$ depends on the context in which the downdating arises. We consider the implications of the bounds (3.10) and (3.11) in Section 5 after we make a connection between downdating and the factorization of Toeplitz matrices.

It is easily seen that a single step of the hyperbolic or mixed downdating algorithm requires $4(n-k)+O(1)$ multiplications. A substantial increase in efficiency can be achieved by considering the following modified downdating problem. Given α_k, $\beta_k \in \Re$ and $\mathbf{w}_k$, $\mathbf{x}_k \in \Re^n$ that satisfy

$$\mathbf{e}_j^T\mathbf{w}_k = 0 \ , \ j < k \quad \text{and} \quad \mathbf{e}_j^T\mathbf{x}_k = 0 \ , \ j \le k \ ,$$

find α_{k+1}, β_{k+1} and $\mathbf{w}_{k+1}$, $\mathbf{x}_{k+1} \in \Re^n$ that satisfy

$$\alpha_{k+1}^2\mathbf{w}_{k+1}\mathbf{w}_{k+1}^T - \beta_{k+1}^2\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T = \alpha_k^2 Z\mathbf{w}_k\mathbf{w}_k^T Z^T - \beta_k^2\mathbf{x}_k\mathbf{x}_k^T \ ,$$

with

$$\mathbf{e}_j^T\mathbf{w}_k = 0 \ , \ j < k \quad \text{and} \quad \mathbf{e}_j^T\mathbf{x}_k = 0 \ , \ j \le k \ .$$

If we make the identification

$$\mathbf{u}_k = \alpha_k\mathbf{w}_k \quad \text{and} \quad \mathbf{v}_k = \beta_k\mathbf{x}_k \ ,$$

then we find that the modified elementary downdating problem is equivalent to the elementary downdating problem. However, the extra parameters can be chosen judiciously to eliminate some multiplications. For example, if we take $\alpha_k = \beta_k$, $\alpha_{k+1} = \beta_{k+1}$, then from (3.3a), (3.3b) and (3.4),

$$\sin\theta_k = \mathbf{e}_{k+1}^T\mathbf{x}_k/\mathbf{e}_k^T\mathbf{w}_k \ , \tag{3.12a}$$

$$\alpha_{k+1} = \alpha_k/\cos\theta_k \ , \tag{3.12b}$$

and

$$\mathbf{w}_{k+1} = Z\mathbf{w}_k - \sin\theta_k\mathbf{x}_k \ , \tag{3.13a}$$

$$\mathbf{x}_{k+1} = -\sin\theta_k Z\mathbf{w}_k + \mathbf{x}_k \ . \tag{3.13b}$$

Equations (3.12a)–(3.13b) form a basis for a *scaled hyperbolic* elementary downdating algorithm which requires $2(n-k)+O(1)$ multiplications. This is about half the number required by the unscaled algorithm based on (3.4). (The price is an increased likelihood of underflow or overflow, but this can be avoided if suitable precautions are taken in the code.)

Similarly, from (3.7a) and (3.7b) we can obtain a *scaled mixed* elementary downdating algorithm via

$$\begin{aligned} \sin\theta_k &= \beta_k\mathbf{e}_{k+1}^T\mathbf{x}_k/\alpha_k\mathbf{e}_k^T\mathbf{w}_k \ , \\ \alpha_{k+1} &= \alpha_k\cos\theta_k \ , \\ \beta_{k+1} &= \beta_k/\cos\theta_k \ , \end{aligned}$$

and

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k - \frac{\sin\theta_k\alpha_k}{\beta_k}Z\mathbf{w}_k \ , \\ \mathbf{w}_{k+1} &= -\frac{\sin\theta_k\beta_{k+1}}{\alpha_{k+1}}\mathbf{x}_{k+1} + Z\mathbf{w}_k \ . \end{aligned}$$

The stability properties of scaled mixed algorithms are similar to those of the corresponding unscaled algorithms [12].

4 Symmetric Factorization

We adopt the following definition from [18].

Definition 4.1: An $n \times n$ symmetric matrix T has displacement rank 2 iff there exist vectors $\mathbf{u}$, $\mathbf{v} \in \Re^n$ such that

$$T - ZTZ^T = \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T \ . \tag{4.1}$$

□

The vectors $\mathbf{u}$ and $\mathbf{v}$ are called the generators of T and determine the matrix T uniquely. Whenever we want to stress the dependence of T on $\mathbf{u}$ and $\mathbf{v}$ we write $T = T(\mathbf{u}\,,\,\mathbf{v})$.

In the sequel we will be concerned with a subset $\mathcal{T}$ of all matrices satisfying (4.1). The subset is defined as follows.

Definition 4.2: A matrix T is in $\mathcal{T}$ iff

(a) T is positive definite,

(b) T satisfies (4.1) with generators $\mathbf{u}$ and $\mathbf{v}$,

(c) $\mathbf{v}^T\mathbf{e}_1 = 0$, i.e., the first component of $\mathbf{v}$ is zero.

□

It is well known that positive definite $n \times n$ Toeplitz matrices form a subset of $\mathcal{T}$. Indeed, if $T = (t_{|i-j|})_{i,j=0}^{n-1}$, then

$$T - ZTZ^T = \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T \ ,$$

where

$$\begin{aligned} \mathbf{u}^T &= (t_0\,,\,t_1\,,\,\ldots\,,\,t_{n-1})\,/\sqrt{t_0}\ , \\ \mathbf{v}^T &= (0\,,\,t_1\,,\,\ldots\,,\,t_{n-1})\,/\sqrt{t_0}\ . \end{aligned}$$

The set $\mathcal{T}$ also contains matrices which are not Toeplitz, as the following example shows.

Example: Let

$$T = \begin{bmatrix} 25 & 20 & 15 \\ 20 & 32 & 29 \\ 15 & 29 & 40 \end{bmatrix} \ , \quad \mathbf{u} = \begin{bmatrix} 5 \\ 4 \\ 3 \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} 0 \\ 3 \\ 1 \end{bmatrix} \ .$$

It is easy to check that T is positive definite. Moreover,

$$T - ZTZ^T = \begin{bmatrix} 25 & 20 & 15 \\ 20 & 7 & 9 \\ 15 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 25 & 20 & 15 \\ 20 & 16 & 12 \\ 15 & 12 & 9 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 9 & 3 \\ 0 & 3 & 1 \end{bmatrix} = \mathbf{u}\mathbf{u}^T - \mathbf{v}\mathbf{v}^T \ .$$

Hence $T = T(\mathbf{u}\,,\,\mathbf{v}) \in \mathcal{T}$, but T is not Toeplitz.

□

We now establish a connection between the elementary downdating problem and symmetric factorizations of a matrix from the set $\mathcal{T}$.

Let $T = T(\mathbf{u}\,,\,\mathbf{v}) \in \mathcal{T}$. Set

$$\mathbf{u}_1 = \mathbf{u}, \qquad \mathbf{v}_1 = \mathbf{v}$$

and, for $k = 1, \ldots, n-1$, solve the elementary downdating problem defined by (3.1),

$$\mathbf{u}_{k+1}\mathbf{u}_{k+1}^T - \mathbf{v}_{k+1}\mathbf{v}_{k+1}^T = Z\mathbf{u}_{\mathbf{k}}\mathbf{u}_k^T Z^T - \mathbf{v}_k\mathbf{v}_k^T \ ,$$

which we assume for the moment has a solution for each k. On summing over $k = 1, \ldots, n-1$ we obtain

$$\sum_{k=1}^{n-1} \mathbf{u}_{k+1}\mathbf{u}_{k+1}^T - \sum_{k=1}^{n-1} \mathbf{v}_{k+1}\mathbf{v}_{k+1}^T = \sum_{k=1}^{n-1} Z\mathbf{u}_k\mathbf{u}_k^T Z^T - \sum_{k=1}^{n-1} \mathbf{v}_k\mathbf{v}_k^T .$$

If we now observe that, from (3.2),

$$Z\mathbf{u}_n = \mathbf{v}_n = 0 ,$$

we arrive at the following relation:

$$\sum_{k=1}^{n} \mathbf{u}_k\mathbf{u}_k^T - Z\left(\sum_{k=1}^{n} \mathbf{u}_k\mathbf{u}_k^T\right) Z^T = \mathbf{u}_1\mathbf{u}_1^T - \mathbf{v}_1\mathbf{v}_1^T , \tag{4.2}$$

which implies that $\sum_{k=1}^{n} \mathbf{u}_k\mathbf{u}_k^T \in \mathcal{T}$. Moreover, as matrices having the same generators are identical, we obtain

$$T = \sum_{k=1}^{n} \mathbf{u}_k\mathbf{u}_k^T = U^T U ,$$

where

$$U = \sum_{k=1}^{n} \mathbf{e}_k\mathbf{u}_k^T$$

is upper triangular, and hence is the Cholesky factor of T. We have derived, albeit in a rather indirect manner, the basis of an algorithm for calculating the Cholesky decomposition of a matrix from the set $\mathcal{T}$.

We now return to the question of existence of a solution to the elementary downdating problem for each $k = 1, \ldots, n-1$. It is easy to verify that, if $T \in \mathcal{T}$, then $|\mathbf{e}_1^T\mathbf{u}_1| > |\mathbf{e}_2^T\mathbf{v}_1|$. Using (4.2) and (3.1), it can be shown by induction on k that

$$|\mathbf{e}_k^T\mathbf{u}_k| > |\mathbf{e}_{k+1}^T\mathbf{v}_k|, \qquad k = 2, \ldots, n-1.$$

Consequently, $|\sin\theta_k| < 1$ in (3.3a), and the elementary downdating problem has a solution for each $k = 1, \ldots, n-1$.

To summarize, we have the following algorithm for factorizing a matrix $T = T(\mathbf{u}\,,\,\mathbf{v}) \in \mathcal{T}$.

Algorithm *FACTOR(T)*:

Set $\mathbf{u}_1 = \mathbf{u}$, $\mathbf{v}_1 = \mathbf{v}$.

For $k = 1, \ldots, n-1$ calculate $\mathbf{u}_{k+1}$, $\mathbf{v}_{k+1}$ such that

$$\begin{aligned} \mathbf{u}_{k+1}\mathbf{u}_{k+1}^T - \mathbf{v}_{k+1}\mathbf{v}_{k+1}^T &= Z\mathbf{u}_k\mathbf{u}_k^T Z^T - \mathbf{v}_k\mathbf{v}_k^T , \\ \mathbf{e}_{k+1}^T\mathbf{v}_{k+1} &= 0 . \end{aligned}$$

Then $T = U^T U$, where $U = \sum_{k=1}^{n} \mathbf{e}_k\mathbf{u}_k^T$.

$\square$

In fact we have not one algorithm but a class of factorization algorithms, where each algorithm corresponds to a particular way of realizing the elementary downdating steps. For example, the connection with the scaled elementary downdating problem is straightforward. On making the identification

$$\mathbf{u}_k = \alpha_k\mathbf{w}_k \quad \text{and} \quad \mathbf{v}_k = \beta_k\mathbf{x}_k , \tag{4.3}$$

we obtain

$$T = W^T D^2 W ,$$

where

$$W = \sum_{k=1}^{n} \mathbf{e}_k \mathbf{w}_k^T ,$$

$$D = \sum_{k=1}^{n} \alpha_k \mathbf{e}_k \mathbf{e}_k^T .$$

It is clear from Section 3 that Algorithm *FACTOR(T)* requires $2n^2 + O(n)$ multiplications when the unscaled version of elementary downdating is used, and $n^2 + O(n)$ multiplications when the scaled version of elementary downdating is used. However, in the sequel we do not dwell on the precise details of algorithms. Using (4.3), we can relate algorithms based on the scaled elementary downdating problem to those based on the unscaled elementary downdating problem. Thus, for simplicity, we consider only the unscaled elementary downdating algorithms.

5 Analysis of Factorization Algorithms

In this section we present a numerical stability analysis of the factorization of $T \in \mathcal{T}$ via Algorithm *FACTOR(T)*. The result of the analysis is applied to the case when the matrix T is Toeplitz.

Let $\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{v}}_k$ be the values of $\mathbf{u}_k$, $\mathbf{v}_k$ that are computed in floating point arithmetic with relative machine relative precision ϵ. The computed quantities $\tilde{\mathbf{u}}_k$ and $\tilde{\mathbf{v}}_k$ satisfy the relations

$$\tilde{\mathbf{u}}_k = \mathbf{u}_k + O(\epsilon), \quad \tilde{\mathbf{v}}_k = \mathbf{v}_k + O(\epsilon), \tag{5.1}$$

and the aim of this section is to provide a first order analysis of the error. By a first order analysis we mean that the error can be bounded by a function which has an asymptotic expansion in powers of ϵ, but we only consider the first term of this asymptotic expansion. One should think of $\epsilon \to 0+$ while the problem remains fixed [19]. Thus, in this section (except for Corollary 5.1) we omit functions of n from the "O" terms in relations such as (5.1) and (5.2).

The computed vectors $\tilde{\mathbf{u}}_k$, $\tilde{\mathbf{v}}_k$ satisfy a perturbed version (3.9) of (3.1). On summing (3.9) over $k = 1, \ldots, n-1$ we obtain

$$\tilde{T} - Z\tilde{T}Z^T = \tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T - (Z\tilde{\mathbf{u}}_n\tilde{\mathbf{u}}_n^T Z^T - \tilde{\mathbf{v}}_n\tilde{\mathbf{v}}_n^T) + \epsilon \sum_{k=1}^{n-1} G_k + O(\epsilon^2) ,$$

where

$$\tilde{T} = \tilde{U}^T\tilde{U} ,$$

$$\tilde{U} = \sum_{k=1}^{n} \mathbf{e}_k \tilde{\mathbf{u}}_k^T .$$

Since

$$Z\tilde{\mathbf{u}}_n = O(\epsilon), \quad \tilde{\mathbf{v}}_n = O(\epsilon) ,$$

we find that

$$\tilde{T} - Z\tilde{T}Z^T = \tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T + \epsilon \sum_{k=1}^{n-1} G_k + O(\epsilon^2) . \tag{5.2}$$

Now define

$$\tilde{E} = \tilde{T} - T. \tag{5.3}$$

Then, using (4.1), (5.2) and (5.3),

$$\tilde{E} - Z\tilde{E}Z^T = \tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \mathbf{u}\mathbf{u}^T + \tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T - \mathbf{v}\mathbf{v}^T + \epsilon\sum_{k=1}^{n-1} G_k + O(\epsilon^2)\,.$$

In a similar manner we obtain expressions for $Z_j\tilde{E}Z_j^T - Z_{j+1}\tilde{E}Z_{j+1}^T$, $j = 0, \ldots, n-1$. Summing over j gives

$$\tilde{E} = \sum_{j=0}^{n-1} Z_j\Big((\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \mathbf{u}_1\mathbf{u}_1^T) + (\tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T - \mathbf{v}_1\mathbf{v}_1^T)\Big)Z_j^T + \epsilon\sum_{j=0}^{n-1}\sum_{k=1}^{n-1} Z_jG_kZ_j^T + O(\epsilon^2)\,. \tag{5.4}$$

We see from (5.4) that the error consists of two parts – the first associated with initial errors and the second associated with the fact that (5.2) contains an inhomogeneous term. Now

$$\|\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \mathbf{u}\mathbf{u}^T\| \le 2\|\mathbf{u}\|\,\|\tilde{\mathbf{u}}_1 - \mathbf{u}\| + O(\epsilon^2)\,,$$
$$\|\tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T - \mathbf{v}\mathbf{v}^T\| \le 2\|\mathbf{v}\|\,\|\tilde{\mathbf{v}}_1 - \mathbf{v}\| + O(\epsilon^2)\,.$$

Furthermore, from (4.1),

$$Tr(T) - Tr(ZTZ^T) = \|\mathbf{u}\|^2 - \|\mathbf{v}\|^2 > 0\,,$$

and hence

$$\Big\|\sum_{j=0}^{n-1} Z_j(\tilde{\mathbf{u}}_1\tilde{\mathbf{u}}_1^T - \mathbf{u}\mathbf{u}^T + \tilde{\mathbf{v}}_1\tilde{\mathbf{v}}_1^T - \mathbf{v}\mathbf{v}^T)Z_j^T\Big\| \le 2n\|\mathbf{u}\|\Big(\|\tilde{\mathbf{u}}_1 - \mathbf{u}\| + \|\tilde{\mathbf{v}}_1 - \mathbf{v}\|\Big) + O(\epsilon^2)\,. \tag{5.5}$$

This demonstrates that initial errors do not propagate unduly. To investigate the double sum in (5.4) we require a preliminary result.

Lemma 5.1 For $k = 1, 2, \ldots, n-1$ and $j = 0, 1, 2, \ldots$,

$$\|Z_j\mathbf{v}_k\| \le \|Z_{j+1}\mathbf{u}_k\|\,.$$

□

Proof Let

$$T_k = T - \sum_{l=1}^{k}\mathbf{u}_l\mathbf{u}_l^T = \sum_{l=k+1}^{n}\mathbf{u}_l\mathbf{u}_l^T\,.$$

It is easy to verify that

$$T_k - ZT_kZ^T = Z\mathbf{u}_k\mathbf{u}_k^TZ^T - \mathbf{v}_k\mathbf{v}_k^T$$

and, since T_k is positive semi-definite,

$$Tr\Big(Z_jT_kZ_j^T - Z_{j+1}T_kZ_{j+1}^T\Big) = \|Z_{j+1}\mathbf{u}_k\|^2 - \|Z_j\mathbf{v}_k\|^2 \ge 0\,.$$

□

We now demonstrate stability when the mixed version of elementary downdating is used in Algorithm *FACTOR(T)*. In this case the inhomogeneous term G_k satisfies a shifted version of (3.10), that is

$$\|Z_jG_kZ_j^T\| \le c_m\Big(\|Z_{j+1}\mathbf{u}_k\|^2 + \|Z_j\mathbf{v}_k\|^2 + \|Z_j\mathbf{u}_{k+1}\|^2 + \|Z_j\mathbf{v}_{k+1}\|^2\Big)\,, \tag{5.6}$$

where c_m is a positive constant.

Theorem 5.1 Assume that (3.9) and (5.6) hold. Then

$$\|T-\tilde{U}^T\tilde{U}\| \le 2n\|\mathbf{u}\|\Big(\|\tilde{\mathbf{u}}_1-\mathbf{u}\|+\|\tilde{\mathbf{v}}_1-\mathbf{v}\|\Big)+4\epsilon c_m\sum_{j=0}^{n-1}Tr(Z_jTZ_j^T)+O(\epsilon^2)\,.$$

□

Proof Using Lemma 5.1,

$$\|Z_jG_kZ_j^T\| \le 2c_m\Big(\|Z_{j+1}\mathbf{u}_k\|^2+\|Z_j\mathbf{u}_{k+1}\|^2\Big).$$

Furthermore, since

$$Tr(Z_jTZ_j^T)=\sum_{k=1}^{n}\|Z_j\mathbf{u}_k\|^2,$$

it follows that

$$\Big\|\sum_{j=0}^{n-1}\sum_{k=1}^{n}Z_jG_kZ_j^T\Big\| \le 4c_m\sum_{j=0}^{n-1}Tr(Z_jTZ_j^T)\,. \tag{5.7}$$

The result now follows from (5.4), (5.5) and (5.7).

□

For the hyperbolic version of the elementary downdating algorithms a shifted version of the weaker bound (3.11) on G_k holds (see [6]), namely

$$\|Z_jG_kZ_j^T\| \le c_h\|H(\theta_k)\|(\|Z_{j+1}\mathbf{u}_k\|+\|Z_j\mathbf{v}_k\|)(\|Z_j\mathbf{u}_{k+1}\|+\|Z_j\mathbf{v}_{k+1}\|)\,. \tag{5.8}$$

By Lemma 5.1, this simplifies to

$$\|Z_jG_kZ_j^T\| \le 4c_h\|H(\theta_k)\|\;\|Z_{j+1}\mathbf{u}_k\|\;\|Z_j\mathbf{u}_{k+1}\|\,. \tag{5.9}$$

The essential difference between (3.10) and (3.11) is the occurence of the multiplier $\|H(\theta_k)\|$ which can be quite large. This term explains numerical difficulties in applications such as the downdating of a Cholesky decomposition [6]. However, because of the special structure of the matrix T, it is of lesser importance here, in view of the following result.

Lemma 5.2 For $k=1,2,\ldots,n-1$ and $j=0,1,\ldots,n-k$,

$$\|H(\theta_k)\|\;\|Z_j\mathbf{u}_{k+1}\| \le 2(n-k-j)\|Z_{j+1}\mathbf{u}_k\|.$$

□

Proof It is easy to verify from (3.4) that

$$\frac{1\mp\sin\theta_k}{\cos\theta_k}(\mathbf{u}_{k+1}\mp\mathbf{v}_{k+1})=Z\mathbf{u}_k\mp\mathbf{v}_k\,,$$

and from (2.1) that

$$\|H(\theta_k)\|=\frac{1+|\sin\theta|}{\cos\theta}\,.$$

Thus,

$$\begin{aligned}\|H(\theta_k)\|\;\|Z_j\mathbf{u}_{k+1}\| &\le \|H(\theta_k)\|\;\|Z_j\mathbf{v}_{k+1}\|+\|Z_{j+1}\mathbf{u}_k\|+\|Z_j\mathbf{v}_k\|\\ &\le \|H(\theta_k)\|\;\|Z_{j+1}\mathbf{u}_{k+1}\|+2\|Z_{j+1}\mathbf{u}_k\|\,,\end{aligned}$$

where the last inequality was obtained using Lemma 5.1. Thus

$$\|H(\theta_k)\|\;\|Z_j\mathbf{u}_{k+1}\| \le 2\sum_{l=j+1}^{n-k}\|Z_l\mathbf{u}_k\|\,,$$

and the result follows.

□

Remark Lemma 5.2 does not hold for the computed quantities unless we introduce an $O(\epsilon)$ term. However in a first order analysis we only need it to hold for the exact quantities.

Theorem 5.2 Assume that (3.9) and (5.8) hold. Then

$$\|T - \tilde{U}^T\tilde{U}\| \leq 2n\|\mathbf{u}\|(\|\tilde{\mathbf{u}}_1 - \mathbf{u}\| + \|\tilde{\mathbf{v}}_1 - \mathbf{v}\|) + 8\epsilon c_h \sum_{j=1}^{n-1}(n-j)Tr(Z_jTZ_j^T) + O(\epsilon^2) .$$

□

Proof Applying Lemma 5.2 to (5.9) gives

$$\|Z_jG_kZ_j^T\| \leq 8c_h(n-j-1)\|Z_{j+1}\mathbf{u}_k\|^2 ,$$

and hence

$$\begin{aligned} \left\|\sum_{j=0}^{n-1}\sum_{k=1}^{n-1} Z_jG_kZ_j^T\right\| &\leq 8c_h \sum_{j=1}^{n-1}\sum_{k=1}^{n-1}(n-j)\|Z_j\mathbf{u}_k\|^2 \\ &\leq 8c_h \sum_{j=1}^{n-1}(n-j)Tr(Z_jTZ_j^T) . \end{aligned} \tag{5.10}$$

The result now follows from (5.4), (5.5) and (5.10).

□

Note that, when T is Toeplitz,

$$Tr(Z_jTZ_j^T) = (n-j)t_0 .$$

Hence, from Theorems 5.1 and 5.2, we obtain our main result on the stability of the factorization algorithms based on Algorithm *FACTOR(T)* for a symmetric positive definite Toeplitz matrix:

Corollary 5.1 The factorization algorithm *FACTOR(T)* applied to a symmetric positive definite Toeplitz matrix T produces an upper triangular matrix $\tilde{U}$ such that

$$T = \tilde{U}^T\tilde{U} + \Delta T ,$$

where $\|\Delta T\| = O(\epsilon t_0 n^2)$ when mixed downdating is used, and $\|\Delta T\| = O(\epsilon t_0 n^3)$ when hyperbolic downdating is used.

□

6 The Connection with the Bareiss algorithm

In his 1969 paper [2], Bareiss proposed an $O(n^2)$ algorithm for solving Toeplitz linear systems. For a symmetric Toeplitz matrix T, the algorithm, called a *symmetric Bareiss algorithm* in [22], can be expressed as follows. Start with a matrix $A^{(0)} := T$ and partition it in two ways:

$$A^{(0)} = \begin{pmatrix} U^{(0)} \\ T^{(1)} \end{pmatrix} \quad , \quad A^{(0)} = \begin{pmatrix} T^{(-1)} \\ L^{(0)} \end{pmatrix} ,$$

where $U^{(0)}$ is the first row of T and $L^{(0)}$ is the last row of T. Now, starting from $A^{(0)}$, compute successively two matrix sequences $\{A^{(i)}\}$ and $\{A^{(-i)}\}$, $i = 1, \ldots, n-1$, according to the relations

$$A^{(i)} = A^{(i-1)} - \alpha_{i-1}Z_iA^{(-i+1)} \quad , \quad A^{(-i)} = A^{(-i+1)} - \alpha_{-i+1}Z_i^TA^{(i-1)} . \tag{6.1}$$

For $1 \leq i \leq n-1$, partition $A^{(i)}$ and $A^{(-i)}$ as follows:

$$A^{(i)} = \begin{pmatrix} U^{(i)} \\ T^{(i+1)} \end{pmatrix} \quad , \quad A^{(-i)} = \begin{pmatrix} T^{(-i-1)} \\ L^{(i)} \end{pmatrix} ,$$

where $U^{(i)}$ denotes the first $i+1$ rows of $A^{(i)}$, and $L^{(i)}$ denotes the last $i+1$ rows of $A^{(-i)}$. It is shown in [2] that

(a) $T^{(i+1)}$ and $T^{(-i-1)}$ are Toeplitz,

(b) for a proper choice of α_{i-1} and α_{-i+1}, the matrices $L^{(i)}$ and $U^{(i)}$ are lower and upper trapezoidal, respectively, and

(c) with the choice of α_{i-1} and α_{-i+1} as in (b), the Toeplitz matrix $T^{(-i-1)}$ has zero elements in positions $2, \ldots, i+1$ of its first row, while the Toeplitz matrix $T^{(i+1)}$ has zero elements in positions $n-1, \ldots, n-i$ of its last row.

Pictorially,

$$A^{(i)} = \left(\frac{U^{(i)}}{T^{(i+1)}} \right) = \left(\begin{array}{cccccccc} \times & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \times \\ 0 & \times & & & & & & \times \\ \vdots & \ddots & \times & \times & & & & \vdots \\ 0 & \cdots & 0 & \times & \cdots & \cdots & \cdots & \times \\ \hline \times & 0 & \cdots & 0 & \times & \cdots & \cdots & \times \\ \vdots & \ddots & & & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & & & \ddots & \ddots & \vdots \\ \times & \cdots & \cdots & \times & 0 & \cdots & 0 & \times \end{array} \right)$$

$$A^{(-i)} = \left(\frac{T^{(-i-1)}}{L^{(i)}} \right) = \left(\begin{array}{cccccccc} \times & 0 & \cdots & 0 & \times & \cdots & \cdots & \times \\ \vdots & \ddots & & & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & & & \ddots & \ddots & \vdots \\ \times & \cdots & \cdots & \times & 0 & \cdots & 0 & \times \\ \hline \times & \cdots & \cdots & \times & \times & 0 & \cdots & 0 \\ \vdots & & & & & \ddots & \ddots & \vdots \\ \vdots & & & & & & \ddots & 0 \\ \times & \times & \cdots & \cdots & \cdots & \cdots & \cdots & \times \end{array} \right)$$

After $n-1$ steps, the matrices $A^{(n-1)}$ and $A^{(-n+1)}$ are lower and upper triangular, respectively. At step i only rows $i+1, \ldots, n$ of $A^{(i)}$ and rows $1, 2, \ldots, n-i$ of $A^{(-i)}$ are modified; the remaining rows stay unchanged. Moreover, Bareiss [2] noticed that, because of the symmetry of T,

$$T^{(i+1)} = J_{i+1} T^{(-i-1)} J_n \quad \text{and} \quad \alpha_{i-1} = \alpha_{-i+1} \ , \tag{6.2}$$

Here J_{i+1} and J_n are the reversal matrices of dimension $(i+1) \times (i+1)$ and $n \times n$ respectively.

Now, taking into account (6.2), it can be seen that the essential part of a step of the Bareiss algorithm (6.1) can be written as follows:

$$\left(\begin{array}{cccc} t_{i+2}^{(i+1)} & t_{i+3}^{(i+1)} & \cdots & t_n^{(i+1)} \\ 0 & t_{i+3}^{(-i-1)} & \cdots & t_n^{(-i-1)} \end{array} \right) = \left(\begin{array}{cc} 1 & -\alpha_{i-1} \\ -\alpha_{i-1} & 1 \end{array} \right) \left(\begin{array}{cccc} t_{i+2}^{(i)} & t_{i+3}^{(i)} & \cdots & t_n^{(i)} \\ t_{i+2}^{(-i)} & t_{i+3}^{(-i)} & \cdots & t_n^{(-i)} \end{array} \right) , \tag{6.3}$$

where $(t_{i+2}^{(-i)}, t_{i+2}^{(-i)}, \ldots, t_n^{(-i)})$ are the last $(n-i-1)$ components of the first row of $T^{(-i)}$, and $(t_{i+2}^{(i)}, t_{i+3}^{(i)}, \ldots, t_n^{(i)})$ are the last $(n-i-1)$ components of the first row of $T^{(i)}$.

Note that (6.3) has the same form as (3.13a)–(3.13b), and hence a connection between the Bareiss algorithm and algorithm $FACTOR(T)$ is evident. That such a connection exists was observed by Sweet [22], and later by Delosme and Ipsen [11]. Sweet [22] related a step of the

Bareiss algorithm (6.3) to a step of Bennett's downdating procedure [3]. Next, he derived the LU factorization of a Toeplitz matrix as a sequence of Bennett's downdating steps. Finally, he estimated the forward error in the decomposition using Fletcher and Powell's methodology [12]. This paper generalizes and presents new derivations of the results obtained in [22].

7 Numerical examples

We adopt from [17] the following definitions of forward and backward stability.

Definition 7.1: An algorithm for solving the equation (1.1) is *forward stable* if the computed solution $\tilde{x}$ satisfies

$$||x - \tilde{x}|| \leq c_1(n)\epsilon \mathrm{cond}(T)||\tilde{x}|| ,$$

where $\mathrm{cond}(T) = ||T||\,||T^{-1}||$ is the condition number of T, and $c_1(n)$ may grow at most as fast as a polynomial in n, the dimension of the system.

Definition 7.2: An algorithm for solving the equation (1.1) is *backward stable* if the computed solution $\tilde{x}$ satisfies

$$||T\tilde{x} - b|| \leq c_2(n)\epsilon||T||\,||\tilde{x}|| ,$$

where $c_2(n)$ may grow at most as fast as a polynomial in n, the dimension of the system.

It is known that an algorithm (for solving a system of linear equations) is backward stable iff there exists a matrix ΔT such that

$$(T + \Delta T)\tilde{x} = b \;\;,\;\; ||\Delta T|| \leq c_3(n)\epsilon||T|| ,$$

where $c_3(n)$ may grow at most as fast as a polynomial in n.

Note that our definitions do not require the perturbation ΔT to be Toeplitz, even if the matrix T is Toeplitz. The case that ΔT is Toeplitz is discussed in [13, 24]. The reader is referred to [9, 14, 19] for a detailed treatment of roundoff analysis for general matrix algorithms.

It is easy to see that backward stability implies forward stability, but not vice versa. This is manifested by the size of the residual vector.

Cybenko [10] showed that the L_1 norm of the inverse of a $n \times n$ symmetric positive definite Toeplitz matrix T_n is bounded by

$$\max\left\{\frac{1}{\prod_{i=1}^{n-1}\cos^2\theta_i} \, , \, \frac{1}{\prod_{i=1}^{n-1}(1+\sin\theta_i)}\right\} \leq ||T_n^{-1}||_1 \leq \prod_{i=1}^{n-1}\frac{1+|\sin\theta_i|}{1-|\sin\theta_i|} ,$$

where $\{-\sin\theta_i\}_{i=1}^{n-1}$ are quantities called *reflection coefficients*. It is not difficult to pick the reflection coefficients in such a way that the corresponding Toeplitz matrix T_n satisfies

$$\mathrm{cond}(T_n) \approx 1/\epsilon .$$

One possible way of constructing a Toeplitz matrix with given reflection coefficients $\{-\sin\theta_i\}_{i=1}^{n-1}$ is by tracing the elementary downdating steps backwards.

An example of a symmetric positive definite Toeplitz matrix that can be made poorly conditioned by suitable choice of parameters is the *Prolate* matrix [21, 23], defined by

$$t_k = \begin{cases} 2\omega & \text{if } k = 0, \\ \frac{\sin(2\pi\omega k)}{\pi k} & \text{otherwise,} \end{cases}$$

where $0 \leq \omega \leq \frac{1}{2}$. For small ω the eigenvalues of the Prolate matrix cluster around 0 and 1.

We performed numerical tests in which we solved systems of Toeplitz linear equations using variants of the Bareiss and Levinson algorithms, and (for comparison) the standard Cholesky method. The relative machine precision was $\epsilon = 2^{-53} \approx 10^{-16}$. We varied the dimension of the system from 10 to 100, the condition number of the matrix from 1 to ϵ^{-1}, the signs of reflection coefficients, and the right hand side so the magnitude of the norm of the solution vector varied from 1 to ϵ^{-1}. In each test we monitored the errors in the decomposition, in the solution vector, and the size of the residual vector.

Let x_B and x_L denote the solutions computed by the Bareiss and Levinson algorithms. Also, let $r_B = Tx_B - b$ and $r_L = Tx_L - b$. Then for the Bareiss algorithms we always observed that the scaled residual

$$s_B \equiv \frac{||r_B||}{\epsilon ||x_B|| ||T||}$$

was of order unity, as small as would be expected for a backward stable method. However, we were not able to find an example which would demonstrate the superiority of the Bareiss algorithm based on mixed downdating over the Bareiss algorithm based on hyperbolic downdating. In fact, the Bareiss algorithm based on hyperbolic downdating often gave slightly smaller errors than the Bareiss algorithm based on mixed downdating. In our experiments with Bareiss algorithms, neither the norm of the error matrix in the decomposition of T nor the residual error in the solution seemed to depend in any clear way on n, although a quadratic or cubic dependence would be expected from the worst-case error bounds of Theorems 5.1–5.2 and Corollary 5.1.

For well conditioned systems the Bareiss and Levinson algorithms behaved similarly, and gave results comparable to results produced by a general stable method (the Cholesky method). Differences between the Bareiss and Levinson algorithms were noticeable only for very ill-conditioned systems and special right-hand side vectors.

For the Levinson algorithm, when the matrix was very ill-conditioned and the norm of the solution vector was of order unity (that is, when the norm of the solution vector did not reflect the ill-conditioning of the matrix), we often observed that the scaled residual

$$s_L \equiv \frac{||r_L||}{\epsilon ||x_L|| ||T||},$$

was as large as 10^5. Varah [23] was the first to observe this behavior of the Levinson algorithm on the Prolate matrix. Higham and Pickering [16] used a search method proposed in [15] to generate Toeplitz matrices for which the Levinson algorithm yields large residual errors. However, the search never produced s_L larger than $5 \cdot 10^5$. It plausible that s_L is a slowly increasing function of n and $1/\epsilon$.

Tables 7.1–7.3 show typical behavior of the Cholesky, Bareiss and Levinson algorithms for ill-conditioned Toeplitz systems of linear equations when the norm of the solution vectors is of order unity. The decomposition error was measured for the Cholesky and Bareiss algorithms by the quotient $||T - L \cdot L^T||/(\epsilon \cdot ||T||)$, where L was the computed factor of T. The solution error was measured by the quotient $||x_{comp} - x||/||x||$, where x_{comp} was the computed solution vector. Finally, the residual error was measured by the quotient $||T \cdot x_{comp} - b||/(||T|| \cdot ||x_{comp}|| \cdot \epsilon)$.

	decomp. error	soln. error	resid. error
Cholesky	$5.09 \cdot 10^{-1}$	$7.67 \cdot 10^{-3}$	$1.25 \cdot 10^{0}$
Bareiss(hyp)	$3.45 \cdot 10^{0}$	$1.40 \cdot 10^{-2}$	$8.72 \cdot 10^{-1}$
Bareiss(mixed)	$2.73 \cdot 10^{0}$	$1.41 \cdot 10^{0}$	$1.09 \cdot 10^{0}$
Levinson		$5.30 \cdot 10^{0}$	$4.57 \cdot 10^{3}$

Table 7.1: Prolate matrix, $n = 21$, $\omega = 0.25$, $cond = 3.19 \cdot 10^{14}$

	decomp. error	soln. error	resid. error
Cholesky	$1.72 \cdot 10^{-1}$	$6.84 \cdot 10^{-2}$	$3.11 \cdot 10^{-1}$
Bareiss(hyp)	$2.91 \cdot 10^{0}$	$2.19 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$
Bareiss(mixed)	$3.63 \cdot 10^{0}$	$2.48 \cdot 10^{-1}$	$2.47 \cdot 10^{-1}$
Levinson		$5.27 \cdot 10^{-1}$	$1.47 \cdot 10^{5}$

Table 7.2: Reflection coefficients $|\sin\theta_i|$ of the same magnitude $|K|$ but alternating signs, $|K| = 0.8956680108101296$, $n = 41$, $cond = 8.5 \cdot 10^{15}$

	decomp. error	soln. error	resid. error
Cholesky	$8.51 \cdot 10^{-1}$	$3.21 \cdot 10^{-2}$	$4.28 \cdot 10^{-1}$
Bareiss(hyp)	$8.06 \cdot 10^{0}$	$1.13 \cdot 10^{-1}$	$2.28 \cdot 10^{-1}$
Bareiss(mixed)	$6.71 \cdot 10^{0}$	$1.16 \cdot 10^{-1}$	$3.20 \cdot 10^{-1}$
Levinson		$2.60 \cdot 10^{-1}$	$1.06 \cdot 10^{5}$

Table 7.3: Reflection coefficients $|\sin\theta_i|$ of the same magnitude but alternating signs, $|K| = 0.9795872473975045$, $n = 92$, $cond = 2.77 \cdot 10^{15}$

8 Conclusions

This paper generalizes and presents new derivations of results obtained earlier by Sweet [22]. The bound in Corollary 5.1 for the case of mixed downdating is stronger than that given in [22]. The applicability of the Bareiss algorithms based on elementary downdating steps is extended to a class of matrices, satisfying Definition 4.2, which includes symmetric positive definite Toeplitz matrices. The approach via elementary downdating greatly simplifies roundoff error analysis. Lemmas 5.1 and 5.2 appear to be new. The stability of the Bareiss algorithms follows directly from these Lemmas and the results on the roundoff error analysis for elementary downdating steps given in [6].

The approach via downdating can be extended to the symmetric factorization of positive definite matrices of displacement rank $k \geq 2$ (satisfying additional conditions similar to those listed in Definition 4.2); see [18]. For matrices of displacement rank k the factorization algorithm uses elementary rank-k downdating via hyperbolic Householder or mixed Householder reflections [8, 20].

We conclude by noting that the Bariess algorithms guarantee small residual errors in the sense of Definition 7.2, but the Levinson algorithm can yield residuals at least five orders of magnitude larger than those expected for a backward stable method. This result suggests that, if the Levinson algorithm is used in applications where the reflection coefficients are not known in advance to be positive, the residuals should be computed to see if they are acceptably small. This can be done in $O(n \log n)$ arithmetic operations (using the FFT).

It is an interesting open question whether the Levinson algorithm can give scaled residual errors which are arbitrarily large (for matrices which are numerically nonsingular). A related question is whether the Levinson algorithm, for positive definite Toeplitz matrices T without a restriction on the reflection coefficients, is stable in the sense of Definitions 7.1 or 7.2.

References

[1] S.T. Alexander, C-T. Pan and R.J. Plemmons, "Analysis of a Recursive least Squares Hyperbolic Rotation Algorithm for Signal Processing", *Linear Algebra and Its Applications*, vol 98, pp 3-40, 1988.

[2] E.H. Bareiss, "Numerical Solution of Linear Equations with Toeplitz and Vector Toeplitz Matrices", *Numerische Mathematik*, vol 13, pp 404-424, 1969.

[3] J.M. Bennett, "Triangular factorization of modified matrices", *Numerische Mathematik,* vol 7, pp 217-221, 1965.

[4] A.W. Bojanczyk, R.P. Brent and F.R. de Hoog, "QR Factorization of Toeplitz Matrices", *Numerische Mathematik*, vol 49, pp 81-94, 1986.

[5] A.W. Bojanczyk, R.P. Brent and F.R. de Hoog, "Stability Analysis of Fast Toeplitz Linear System Solvers", Report CMA-MR17-91, Centre for Mathematical Analysis, The Australian National University, August 1991.

[6] A.W. Bojanczyk, R.P. Brent, P. Van Dooren and F.R. de Hoog, "A Note on Downdating the Cholesky Factorization", *SIAM J. Sci. Statist. Comput.*, vol 8, pp 210-220, 1987.

[7] A.W. Bojanczyk and A. Steinhardt, "Matrix Downdating Techniques for Signal Processing", *Proceedings of the SPIE Conference on Advanced Algorithms and Architectures for Signal Processing III*, vol 975, pp 68-75, 1988.

[8] A.W. Bojanczyk and A.O. Steinhardt, "Stabilized Hyperbolic Householder Transformations", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol ASSP-37, 1989, pp 1286-1288.

[9] J.R. Bunch, "The Weak and Strong Stability of Algorithms in Numerical Linear Algebra", *Linear Algebra and Its Applications*, vol 88/89, pp 49-66, 1987.

[10] G. Cybenko, "The Numerical Stability of the Levinson-Durbin Algorithm for Toeplitz Systems of Equations", *SIAM J. Sci. Statist. Comput.*, vol 1, pp 303-319, 1980.

[11] J-M. Delosme and I.C.F. Ipsen, "From Bareiss's algorithm to the stable computation of partial correlations", *Journal of Computational and Applied Mathematics,* vol 27, pp 53-91, 1989.

[12] R. Fletcher and M.J.D. Powell, "On the Modification of LDL^T Factorizations", *Mathematics of Computation*, vol 28, pp 1067-87, 1974.

[13] I. Gohberg, I. Koltracht and D. Xiao, "On the solution of the Yule-Walker equation", *Proceedings of the SPIE Conference on Advanced Algorithms and Architectures for Signal Processing IV*, vol 1566, July 1991.

[14] G.H. Golub and C. Van Loan, *Matrix Computations*, second edition, Johns Hopkins Press, Baltimore, Maryland, 1989.

[15] N.J. Higham, "Optimization by Direct Search in Matrix Computations", Numerical Analysis Report No. 197, University of Manchester, England, 1991; to appear in *SIAM J. Matrix Anal. Appl.*

[16] N.J. Higham and R.L. Pickering, private communication.

[17] M. Jankowski and H. Wozniakowski, “Iterative Refinement Implies Numerical Stability”, *BIT*, vol 17, pp 303-311, 1977.

[18] T. Kailath, S.Y. Kung and M. Morf, “Displacement Ranks of Matrices and Linear Equations”, *J. Math. Anal. Appl.*, vol 68, pp 395-407, 1979.

[19] W. Miller and C. Wrathall, *Software for Roundoff Analysis of Matrix Algorithms*, Academic Press, 1980.

[20] C.M. Rader and A.O. Steinhardt, “Hyperbolic Householder Transformations”, *IEEE Transaction on Acoustics, Speech and Signal Processing,* vol ASSP-34, 1986, pp 1584-1602.

[21] D. Slepian, “Prolate spheroidal wave functions, Fourier analysis, and uncertainty V: the discrete case”, *Bell Systems Tech. J.*, vol 57, 1978, pp 1371-1430.

[22] D. Sweet, “Numerical Methods for Toeplitz Matrices”, PhD thesis, University of Adelaide, 1982.

[23] J.M. Varah, “The Prolate Matrix: A Good Toeplitz Test Example”, *SIAM Conference on Control, Signals and Systems*, San Francisco, 1990. Also *Linear Algebra Appl.*, vol 187, 1993, pp 269-278.

[24] J.M. Varah, “Backward Error Estimates for Toeplitz and Vandermonde Systems”, preprint, 1992. Also Tech. Report 91-20, Univ. of British Columbia, Sept. 1991.