A Weakly Stable Algorithm for General Toeplitz Systems*

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

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

Frank R. de Hoog
Division of Mathematics and Statistics
CSIRO, GPO Box 1965
Canberra, ACT 2601
frank@dmscanb.cbr.dms.csiro.au

Report TR-CS-93-15
6 August 1993
(revised 24 June 1994)

Abstract

We show that a fast algorithm for the QR factorization of a Toeplitz or Hankel matrix A is weakly stable in the sense that $R^T R$ is close to $A^T A$. Thus, when the algorithm is used to solve the semi-normal equations $R^T Rx = A^T b$, we obtain a weakly stable method for the solution of a nonsingular Toeplitz or Hankel linear system $Ax = b$. The algorithm also applies to the solution of the full-rank Toeplitz or Hankel least squares problem $\min \|Ax - b\|_2$.

1991 Mathematics Subject Classification. Primary 65F25; Secondary 47B35, 65F05, 65F30, 65Y05, 65Y10

Key words and phrases. Cholesky factorization, error analysis, Hankel matrix, least squares, normal equations, orthogonal factorization, QR factorization, semi-normal equations, stability, Toeplitz matrix, weak stability.

1 Introduction

Toeplitz linear systems arise in many applications, and there are many algorithms which solve nonsingular $n \times n$ Toeplitz systems

$$Ax = b$$

in $O(n^2)$ arithmetic operations [2, 13, 44, 45, 49, 54, 55, 65, 67, 77, 78, 80, 81, 85, 86]. Some algorithms are restricted to symmetric systems ($A = A^T$) and others apply to general Toeplitz systems. Because of their recursive nature, most $O(n^2)$ algorithms assume that all leading principal submatrices of A are nonsingular, and break down if this is not the case. These algorithms are generally unstable, because a leading principal submatrix may be poorly conditioned even if A is well conditioned. Thus, stability results often depend on the assumption that A is

The results were announced in *Parallel algorithms and numerical stability for Toeplitz systems*, invited paper presented by Brent at the SIAM Conference on Linear Algebra in Signals, Systems and Control, Seattle, August 16-19, 1993. Revision to appear in *Numerical Algorithms*.
rpb143tr typeset using LaTeX

symmetric positive definite, in which case the leading principal submatrices are at least as well conditioned as A.

Asymptotically faster algorithms exist [1, 4, 14, 20, 40, 47, 58, 59]. Sometimes these algorithms are called *superfast* [1]. We avoid this terminology because, even though the algorithms require only $O(n(\log n)^2)$ arithmetic operations, they may be slower than $O(n^2)$ algorithms for $n < 256$ (see [1, 20, 68]). We prefer the term *asymptotically fast*.

The numerical stability properties of asymptotically fast algorithms are generally either bad [16] or unknown, although some positive partial results have been obtained recently [42]. Attempts to stabilise asymptotically fast algorithms by look-ahead techniques have been made [40], but the look-ahead algorithms are complicated and their worst-case behaviour is unclear. Thus, we do not consider asymptotically fast algorithms further, but restrict our attention to $O(n^2)$ algorithms.

We are concerned with direct methods for the general case, where A is any nonsingular Toeplitz matrix. In this case no $O(n^2)$ algorithm has been proved to be stable. For example, the algorithm of Bareiss [2] has stability properties similar to those of Gaussian elimination without pivoting [9, 10, 73, 76], so is unstable and breaks down if a leading principal minor vanishes. Several authors have suggested the introduction of pivoting or look-ahead (with block steps) in the Bareiss and Levinson algorithms [18, 19, 28, 29, 30, 75, 76], and this is often successful in practice, but in the worst case the overhead is $O(n^3)$ operations. The recent algorithm GKO of Gohberg, Kailath and Olshevsky [34] may be as stable as Gaussian elimination with partial pivoting, but an error analysis has not been published.

In an attempt to achieve stability without pivoting or look-ahead, it is natural to consider algorithms for computing an orthogonal factorization

$$A = QR \tag{1}$$

of A. The first such $O(n^2)$ algorithm was introduced by Sweet [73, 74]. Unfortunately, Sweet's algorithm depends on the condition of a certain submatrices of A, so is unstable [8, 56]. Other $O(n^2)$ algorithms for computing the matrices Q and R or R^{-1} in (1) were given by Bojanczyk, Brent and de Hoog [8], Chun, Kailath and Lev-Ari [21], Cybenko [23], Luk and Qiao [56, 64], and Nagy [60]. To our knowledge none of them has been shown to be stable. In several cases examples show that they are not stable. Unlike the classical $O(n^3)$ Givens or Householder algorithms, the $O(n^2)$ algorithms do not form Q in a numerically stable manner as a product of matrices which are (close to) orthogonal.

Numerical experiments with the algorithm of Bojanczyk, Brent and de Hoog (BBH for short) suggest that the cause of instability is the method for computing the orthogonal matrix Q; the computed upper triangular matrix $\tilde{R}$ is about as good as can be obtained by performing a Cholesky factorization of A^TA, provided the downdates involved in the algorithm are implemented in a certain way (see §4). This result is proved in §5. As a consequence, in §6 we show how the method of semi-normal equations (i.e. the solution of $R^TRx = A^Tb$) can be used to give a weakly stable algorithm for the solution of general Toeplitz or Hankel systems. The result also applies to the solution of full-rank Toeplitz or Hankel least squares problems. For a discussion of the rank-deficient case and a "look-ahead" modification of Cybenko's algorithm, see [43].

In §2 we introduce some notation and conventions. The concepts of stability and weak stability are defined in §3. The Cholesky downdating problem, which arises in the BBH algorithm, is discussed in §4. Numerical results are discussed in §7, and some conclusions are given in §8.

If H is a Hankel matrix, and J is the permutation matrix which on premultiplication reverses the order of the rows of a matrix, then JH is Toeplitz. Also, $(JH)^T(JH) = H^TH$. Thus, our results apply equally to Hankel and Toeplitz matrices. Our results might also be extended to more general classes of matrices with a displacement structure [50, 51, 52, 53]. For simplicity we restrict our attention to the Toeplitz case.

2 Notation and Conventions

Let

$$A = \begin{pmatrix} a_0 & \cdots & a_{n-1} \\ \vdots & \ddots & \vdots \\ a_{1-m} & \cdots & a_{n-m} \end{pmatrix}$$

be a real $m \times n$ Toeplitz matrix, so $a_{i,j} = a_{j-i}$ for $1 \le i \le m$, $1 \le j \le n$. We assume that $m \ge n$ and that A has rank n. Thus $A^T A$ has a Cholesky factorization $A^T A = R^T R$, where R is an upper triangular $n \times n$ matrix. We assume that the diagonal elements of R are positive, so R is unique. Also, $A = QR$, where Q is an $m \times n$ matrix with orthonormal columns.

If the singular values of A are $\sigma_1, \ldots, \sigma_n$, where $\sigma_1 \ge \ldots \ge \sigma_n > 0$, then the spectral condition number of A is

$$\kappa = \kappa_2(A) = \sigma_1/\sigma_n.$$

For convenience in stating the error bounds, we often assume that σ_1 is of order unity, which can be achieved by scaling.

A *displacement operator* $\mathcal{D} : \Re^{m\times n} \to \Re^{(m-1)\times(n-1)}$ is defined as follows: for any $m \times n$ matrix B, $\mathcal{D}(B) = C$, where C is the $(m-1)\times(n-1)$ matrix with entries $c_{i,j} = b_{i+1,j+1} - b_{i,j}$, $1 \le i < m, 1 \le j < n$. Note that $\mathcal{D}B = 0$ iff B is Toeplitz.

Let ε be the machine precision. In our analysis we neglect terms of order $O(\varepsilon^2)$. This can be justified by considering our bounds as asymptotic expansions in the (sufficiently small) parameter ε.

It is often convenient to subsume a polynomial in m and/or n into the "O" notation, and indicate this by a subscript. Thus, an error bound of the form

$$\|E\| = O_m(\varepsilon)$$

means that

$$\|E\| \le P(m)\varepsilon$$

for some polynomial P and all sufficiently small ε. This notation is useful because minor changes in the algorithm or changes in the choice of norm will be absorbed by a change in the polynomial $P(m)$. It is often stated (e.g. by Björck [6]) that *the primary purpose of rounding error analysis is insight*, and insight can be aided by the suppression of superfluous details.[1]

If the error bound depends on κ then this will be mentioned explicitly (e.g. $\|E\| = O_m(\kappa\varepsilon)$). The meaning of "sufficiently small" may depend on κ For example, we may need $\varepsilon < 1/\kappa^2$.

We distinguish several classes of numerical quantities –

1. Exact values, e.g. input data such as a_i.
2. Computed values, usually indicated by a tilde, e.g. $\tilde{u}_i$.
3. Perturbed values given by error analysis, usually indicated by a hat, e.g. $\widehat{a}_{i,j}$, or by the argument ε, e.g. $a_{i,j}(\varepsilon)$. These are not computed, but the error analysis shows that they exist and gives bounds on their difference from the corresponding exact values. Sometimes the perturbations are of computed quantities, e.g. $\tilde{u}_i(\varepsilon)$.

[1] Wilkinson [82] states "... there is a danger that the essential simplicity of the error analysis may be obscured by an excess of detail."

3 Stability and Weak Stability

In this section we give definitions of stability and weak stability of algorithms for solving linear systems.

Consider algorithms for solving a nonsingular, $n \times n$ linear system $Ax = b$, so $m = n$. There are many definitions of numerical stability in the literature, for example [5, 6, 9, 12, 16, 22, 36, 48, 57, 61, 70]. Definitions 1 and 2 below are taken from Bunch [17].

Definition 1 *An algorithm for solving linear equations is* stable *for a class of matrices $\mathcal{A}$ if for each A in $\mathcal{A}$ and for each b the computed solution $\tilde{x}$ to $Ax = b$ satisfies $\widehat{A}\tilde{x} = \widehat{b}$, where $\widehat{A}$ is close to A and $\widehat{b}$ is close to b.*

Definition 1 says that, for stability, the *computed* solution has to be the *exact* solution of a problem which is close to the original problem. This is the classical *backward stability* of Wilkinson [82, 83, 84]. We interpret "close" to mean close in the relative sense in some norm, i.e.

$$\|\widehat{A} - A\|/\|A\| = O_n(\varepsilon), \; \|\widehat{b} - b\|/\|b\| = O_n(\varepsilon). \tag{2}$$

Note that the matrix $\widehat{A}$ is not required to be in the class $\mathcal{A}$. For example, $\mathcal{A}$ might be the class of nonsingular Toeplitz matrices, but $\widehat{A}$ is not required to be a Toeplitz matrix. If we require $\widehat{A} \in \mathcal{A}$ we get what Bunch [17] calls *strong stability.* For a discussion of the difference between stability and strong stability for Toeplitz algorithms, see [46, 79].

Stability does not imply that the computed solution $\tilde{x}$ is close to the exact solution x, unless the problem is well-conditioned. Provided $\kappa\varepsilon$ is sufficiently small, stability implies that

$$\|\tilde{x} - x\|/\|x\| = O_n(\kappa\varepsilon). \tag{3}$$

For more precise results, see Bunch [17] and Wilkinson [82].

As an example, consider the method of Gaussian elimination. Wilkinson [82] shows that

$$\|\widehat{A} - A\|/\|A\| = O_n(g\varepsilon), \tag{4}$$

where $g = g(n)$ is the "growth factor". g depends on whether partial or complete pivoting is used. In practice g is usually moderate, even for partial pivoting. However, a well-known example shows that $g(n) = 2^{n-1}$ is possible for partial pivoting, and recently it has been shown that examples where $g(n)$ grows exponentially with n may arise in applications, e.g. for linear systems arising from boundary value problems. Even for complete pivoting, it has not been *proved* that $g(n)$ is bounded by a polynomial in n. Wilkinson [82] showed that $g(n) \le n^{(\log n)/4+O(1)}$, and Gould [38] recently showed that $g(n) > n$ is possible for $n > 12$; there is still a large gap between these results. Thus, to be sure that Gaussian elimination satisfies Definition 1, we must restrict $\mathcal{A}$ to the class of matrices for which g is $O_n(1)$. In practice this is not a problem, because g can easily be checked *a posteriori.*

Although stability is desirable, it is more than we can prove for many useful algorithms. Thus, following Bunch [17], we define the (weaker, but still useful) property of *weak stability.*

Definition 2 *An algorithm for solving linear equations is* weakly stable *for a class of matrices $\mathcal{A}$ if for each well-conditioned A in $\mathcal{A}$ and for each b the computed solution $\tilde{x}$ to $Ax = b$ is such that $\|\tilde{x} - x\|/\|x\|$ is small.*

In Definition 2, we take "small" to mean $O_n(\varepsilon)$, and "well-conditioned" to mean that $\kappa(A)$ is $O_n(1)$, i.e. is bounded by a polynomial in n. From (3), stability implies weak stability.

Define the *residual* $r = A\tilde{x} - b$. It is well-known [83] that

$$\frac{1}{\kappa}\frac{\|r\|}{\|b\|} \le \frac{\|\tilde{x} - x\|}{\|x\|} \le \kappa\frac{\|r\|}{\|b\|}. \tag{5}$$

Thus, for well-conditioned A, $\|\tilde{x}-x\|/\|x\|$ is small if and only if $\|r\|/\|b\|$ is small. This observation clearly leads to an alternative definition of weak stability:

Definition 3 *An algorithm for solving linear equations is* weakly stable *for a class of matrices $\mathcal{A}$ if for each well-conditioned A in $\mathcal{A}$ and for each b the computed solution $\tilde{x}$ to $Ax = b$ is such that $\|A\tilde{x} - b\|/\|b\|$ is small.*

Now consider computation of the Cholesky factor R of A^TA, where A is an $m \times n$ matrix of full rank n. A good $O(mn^2)$ algorithm is to compute the QR factorization

$$A = QR$$

of A using Householder or Givens transformations [36]. It can be shown [83] that the computed matrices $\tilde{Q}$, $\tilde{R}$ satisfy

$$\widehat{A} = \widehat{Q}\tilde{R} \tag{6}$$

where $\widehat{Q}^T\widehat{Q} = I$, $\tilde{Q}$ is close to $\widehat{Q}$, and $\widehat{A}$ is close to A. Thus, the algorithm is stable in the sense of backward error analysis. Note that $\|A^TA - \tilde{R}^T\tilde{R}\|/\|A^TA\|$ is small, but $\|\tilde{Q} - Q\|$ and $\|\tilde{R} - R\|/\|R\|$ are not necessarily small. Bounds on $\|\tilde{Q} - Q\|$ and $\|\tilde{R} - R\|/\|R\|$ depend on κ, and are discussed in [35, 71, 84].

A different algorithm is to compute (the upper triangular part of) A^TA, and then compute the Cholesky factorization of A^TA by the usual (stable) algorithm. The computed result $\tilde{R}$ is such that $\tilde{R}^T\tilde{R}$ is close to A^TA. However, this does not imply the existence of $\widehat{A}$ and $\widehat{Q}$ such that (6) holds (with $\widehat{A}$ close to A and some $\widehat{Q}$ with $\widehat{Q}^T\widehat{Q} = I$) unless A is well-conditioned [72]. By analogy with Definition 3 above, we may say that Cholesky factorization of A^TA gives a *weakly stable* algorithm for computing R, because the "residual" $A^TA - \tilde{R}^T\tilde{R}$ is small.

4 Cholesky Updating and Downdating

4.1 Updating

The Cholesky updating problem is: given an upper triangular matrix $R \in \Re^{n\times n}$ and a vector $x \in \Re^n$, find an upper triangular matrix U such that

$$U^TU = R^TR + xx^T. \tag{7}$$

The updating problem can be solved in a numerically stable manner by transforming the matrix $\begin{pmatrix} x^T \\ R \end{pmatrix}$ to upper triangular form $\begin{pmatrix} U \\ 0^T \end{pmatrix}$ by applying a sequence of plane rotations on the left. For details, see [36].

4.2 Downdating

The Cholesky downdating problem is: given a upper triangular matrix $R \in \Re^{n\times n}$ and a vector $x \in \Re^n$ such that $R^TR - xx^T$ is positive definite, find an upper triangular matrix U such that

$$U^TU = R^TR - xx^T. \tag{8}$$

Proceeding formally, we can obtain (8) from (7) by replacing x by ix. However, the numerical properties of the updating and downdating problems are very different. The condition that $R^TR - xx^T$ be positive semi-definite is necessary for the existence of a real U satisfying (8). Thus, we would expect the downdating problem to be illconditioned if $R^TR - xx^T$ has small singular values, and Stewart [72] shows that this is true.

There are several algorithms for the Cholesky downdating problem [3, 7, 11, 12, 24, 25, 32, 36, 62, 66] and we shall not discuss them in detail here. What is relevant to us is the error analysis. To simplify the statement of the error bounds, suppose that $\|R\| = O_n(1)$, which implies that $\|x\| = O_n(1)$. Observe that, in exact arithmetic, there is an orthogonal matrix Q such that

$$\begin{pmatrix} x^T \\ U \end{pmatrix} = Q \begin{pmatrix} R \\ 0^T \end{pmatrix}. \tag{9}$$

Suppose the computed upper triangular matrix is $\tilde{U}$. Stewart [72] shows that, for the "Linpack" algorithm [24],

$$\begin{pmatrix} x^T(\varepsilon) \\ \tilde{U}(\varepsilon) \end{pmatrix} = Q(\varepsilon) \begin{pmatrix} R \\ 0^T \end{pmatrix}, \tag{10}$$

where $Q(\varepsilon)$ is an exactly orthogonal matrix,

$$\|x(\varepsilon) - x\| = O_n(\varepsilon), \tag{11}$$

and

$$\|\tilde{U}(\varepsilon) - \tilde{U}\| = O_n(\varepsilon). \tag{12}$$

We can regard $x(\varepsilon)$ as a (backward) perturbation of the input data x, and $\tilde{U}(\varepsilon)$ as a (forward) perturbation of the computed result $\tilde{U}$. Because of this mixture of forward and backward perturbations, a result of this form is sometimes called a "mixed" stability result. If the problem is illconditioned, the backward perturbation is more significant than the forward perturbation.

It is important to note that the error analysis does *not* show that the computed matrix $\tilde{U}$ is close to the exact result U, or that $Q(\varepsilon)$ is close to Q, unless U is well-conditioned.

A stability result of the same form as (10–12) has been established by Bojanczyk, Brent, Van Dooren and de Hoog [7] for their "Algorithm C". Because Algorithm C computes $\tilde{U}$ one row at a time, several (updates and/or) downdates can be pipelined, which is not the case for the Linpack algorithm[2].

The result (10–12) implies that

$$\tilde{U}^T \tilde{U} = R^T R - x x^T + G(\varepsilon), \tag{13}$$

where

$$\|G(\varepsilon)\| = O_n(\varepsilon). \tag{14}$$

Clearly a similar result holds if a sequence of (updates and) downdates is performed, provided that the final and intermediate results are positive definite.

5 A Weak Stability Result for the BBH Algorithm

Our main result (Theorem 1 below) is that the BBH algorithm computes R about as well as would be expected if the Cholesky factorization of $A^T A$ were computed by the usual (stable) algorithm. More precisely, the computed $\tilde{R}$ satisfies

$$A^T A - \tilde{R}^T \tilde{R} = O_m(\varepsilon \|A^T A\|). \tag{15}$$

To avoid having to include $\|A^T A\|$ in the error bounds, we assume for the time being that $\sigma_1(A) = O(1)$.

[2] The recent algorithm of Bischof et al [3] can also be pipelined, but we do not know if it gives results which satisfy equations (10–12) or (13–14).

First consider exact arithmetic. We partition A in two ways:

$$A = \left(\begin{array}{c|c} a_0 & y^T \\ \hline z & A_{-1} \end{array}\right) = \left(\begin{array}{c|c} A_{-1} & \overline{y} \\ \hline \overline{z}^T & a_{n-m} \end{array}\right), \tag{16}$$

where A_{-1} is an $(m-1)\times(n-1)$ Toeplitz matrix,

$$\begin{aligned} y &= (a_1, \ldots, a_{n-1})^T, \\ z &= (a_{-1}, \ldots, a_{1-m})^T, \\ \overline{y} &= (a_{n-1}, \ldots, a_{n-m+1})^T, \\ \overline{z} &= (a_{1-m}, \ldots, a_{n-m-1})^T. \end{aligned}$$

Similarly, we partition R in two ways:

$$R = \left(\begin{array}{c|c} r_{1,1} & u^T \\ \hline 0 & R_b \end{array}\right) = \left(\begin{array}{c|c} R_t & \overline{u} \\ \hline 0^T & r_{n,n} \end{array}\right), \tag{17}$$

where R_b and R_t are $(n-1)\times(n-1)$ upper triangular matrices,

$$\begin{aligned} u &= (r_{1,2}, \ldots, r_{1,n})^T, \\ \overline{u} &= (r_{1,n}, \ldots, r_{n-1,n})^T. \end{aligned}$$

From (16),

$$A^T A = \left(\begin{array}{c|c} a_0^2 + z^T z & a_0 y^T + z^T A_{-1} \\ \hline a_0 y + A_{-1}^T z & A_{-1}^T A_{-1} + y y^T \end{array}\right) = \left(\begin{array}{c|c} A_{-1}^T A_{-1} + \overline{z}\,\overline{z}^T & \vdots \\ \hline \cdots & \cdot \end{array}\right), \tag{18}$$

where the dots indicate entries which are irrelevant for our purposes. Similarly, from (17),

$$R^T R = \left(\begin{array}{c|c} r_{1,1}^2 & r_{1,1} u^T \\ \hline r_{1,1} u & {R_b}^T R_b + u u^T \end{array}\right) = \left(\begin{array}{c|c} {R_t}^T R_t & \vdots \\ \hline \cdots & \cdot \end{array}\right). \tag{19}$$

Equating $A^T A$ and $R^T R$, we obtain

$$r_{1,1}^2 = a_0^2 + z^T z, \tag{20}$$

$$r_{1,1} u = a_0 y + A_{-1}^T z, \tag{21}$$

$$A_{-1}^T A_{-1} + y y^T = {R_b}^T R_b + u u^T, \tag{22}$$

and

$$A_{-1}^T A_{-1} + \overline{z}\,\overline{z}^T = {R_t}^T R_t. \tag{23}$$

Eliminating $A_{-1}^T A_{-1}$ from (22–23) gives the relation

$${R_b}^T R_b = {R_t}^T R_t + y y^T - u u^T - \overline{z}\,\overline{z}^T \tag{24}$$

which is the basis for the BBH algorithm. If R_t were known, then R_b could be computed from (24) using one Cholesky updating step and two Cholesky downdating steps, as discussed in §4. Also, since updating and downdating algorithms can proceed by rows, knowledge of the first k rows of R_t is sufficient to allow the computation of the first k rows of R_b. It is easy to compute the first row of R from relations (20–21). (For future reference, suppose that the computed first row of R is $(\tilde{r}_{1,1}, \tilde{u}^T)$.) It is clear from (17) that the k-th row of R_b defines the $(k+1)$-th row of R_t. Thus, we can compute R_t and R_b row by row. For details see [8].

A straightforward extension of the result (13–14) applies to our problem of computing R_t and R_b. Provided the "Algorithm C" variant of downdating [7] is used, the computed results $\tilde{R}_t$ and $\tilde{R}_b$ satisfy

$$\tilde{R}_b^T \tilde{R}_b = \tilde{R}_t^T \tilde{R}_t + yy^T - \tilde{u}\tilde{u}^T - \overline{z}\,\overline{z}^T + G(\varepsilon) \tag{25}$$

where

$$\|G(\varepsilon)\| = O_m(\varepsilon). \tag{26}$$

Here y, $\overline{z}$ and $\tilde{u}$ are inputs to the up/downdating procedures ($\tilde{u}$ may differ slightly from the exact u because $\tilde{u}$ has been computed from (20–21)). At this point we make no claims about the size of $\|\tilde{R}_b - R_b\|$ and $\|\tilde{R}_t - R_t\|$. All we need is that $\tilde{R}_b$ and $\tilde{R}_t$ exist and are bounded for sufficiently small ε.

Because of the algorithm for their computation, the computed matrices $\tilde{R}_t$ and $\tilde{R}_b$ are related so that we can define the "computed R", say $\tilde{R}$, in a consistent manner by

$$\tilde{R} = \left(\begin{array}{c|c} \tilde{r}_{1,1} & \tilde{u}^T \\ \hline 0 & \tilde{R}_b \end{array}\right) = \left(\begin{array}{c|c} \tilde{R}_t & \vdots \\ \hline 0^T & \cdot \end{array}\right). \tag{27}$$

From (27) we have

$$\tilde{R}^T \tilde{R} = \left(\begin{array}{c|c} \tilde{r}_{1,1}^2 & \tilde{r}_{1,1}\tilde{u}^T \\ \hline \vdots & \tilde{R}_b^T \tilde{R}_b + \tilde{u}\tilde{u}^T \end{array}\right) = \left(\begin{array}{c|c} \tilde{R}_t^T \tilde{R}_t & \vdots \\ \hline \cdots & \cdot \end{array}\right). \tag{28}$$

Recall our definition of the operator $\mathcal{D}$ in §2. From (28) we have

$$\mathcal{D}(\tilde{R}^T \tilde{R}) = \tilde{R}_b^T \tilde{R}_b + \tilde{u}\tilde{u}^T - \tilde{R}_t^T \tilde{R}_t. \tag{29}$$

Thus, from (25),

$$\mathcal{D}(\tilde{R}^T \tilde{R}) = yy^T - \overline{z}\overline{z}^T + G(\varepsilon). \tag{30}$$

Also, from (18),

$$\mathcal{D}(A^T A) = yy^T - \overline{z}\,\overline{z}^T. \tag{31}$$

If $E = \tilde{R}^T \tilde{R} - A^T A$ and $F = \mathcal{D}(E)$ then, from (30–31),

$$F = G(\varepsilon). \tag{32}$$

If $1 \le j \le i \le n$ then, by the definition of $\mathcal{D}(E)$,

$$e_{i,j} - e_{i-j+1,1} = \sum_{k=1}^{j-1} (e_{i-k+1,j-k+1} - e_{i-k,j-k}) = f_{i-1,j-1} + f_{i-2,j-2} + \cdots + f_{i-j+1,1}.$$

The first row of $\tilde{R}^T \tilde{R}$ is $\tilde{r}_{1,1}(\tilde{r}_{1,1}, \tilde{u}^T)$, which is close to $r_{1,1}(r_{1,1}, u^T)$, so the first row of E has norm $O_m(\varepsilon)$. Also, E is symmetric. It follows that

$$\|E\| \le (n-1)\|F\| + O_m(\varepsilon) = O_m(\varepsilon).$$

Thus, after scaling to remove our assumption that $\sigma_1 = O(1)$, we have proved:

Theorem 1 *If the BBH algorithm is used with the downdating steps performed as in "Algorithm C" of* [7], *then the computed Cholesky factor $\tilde{R}$ of $A^T A$ satisfies*

$$\|\tilde{R}^T \tilde{R} - A^T A\| = O_m(\varepsilon \|A^T A\|). \tag{33}$$

Since $A^TA = R^TR$, a comparison of the partitioned forms (19) and (28) shows that

$$\|\tilde{R}_b^T\tilde{R}_b - R_b{}^TR_b\| = O_m(\varepsilon\|R^TR\|) \tag{34}$$

and

$$\|\tilde{R}_t^T\tilde{R}_t - R_t{}^TR_t\| = O_m(\varepsilon\|R^TR\|). \tag{35}$$

From (33) and Stewart's perturbation analysis [72], it follows that

$$\|\tilde{R} - R\|/\|R\| = O_m(\kappa\varepsilon). \tag{36}$$

Note that the condition number κ appears in (36) but not in (33–35).

6 The Semi-Normal Equations

Suppose that our aim is to solve a nonsingular $n \times n$ Toeplitz linear system

$$Ax = b, \tag{37}$$

using $O(n^2)$ arithmetic operations. In exact arithmetic, the *normal equations*

$$A^TAx = A^Tb \tag{38}$$

and the *semi-normal equations*

$$R^TRx = A^Tb, \tag{39}$$

where R satisfies (1), are equivalent to (37).

In most circumstances the use of the normal or semi-normal equations is not recommended, because the condition number $\kappa(A^TA)$ may be as large as $\kappa(A)^2$ (see §5.3 of Golub and Van Loan [36]). When A is Toeplitz (but not symmetric positive definite) we can justify use of the semi-normal equations. This is because the usual stable algorithms for solving (37) directly require $O(n^3)$ arithmetic operations, but we can use the algorithm of §5 to compute (a numerical approximation $\tilde{R}$ to) R in $O(n^2)$ operations, and then solve the seminormal equations (39) in an additional $O(n^2)$ operations.

From Theorem 1, we can compute an upper triangular matrix $\tilde{R}$ such that (33) holds. We can also compute an accurate approximation $\tilde{d}$ to $d = A^Tb$ in $O(n^2)$ operations (using the obvious algorithm) or in $O(n \log n)$ operations (using the Fast Fourier Transform). Now solve the two triangular systems $\tilde{R}^Tw = d$ and $\tilde{R}x = w$. We expect to obtain a result $\tilde{x}$ for which

$$\|\tilde{x} - x\|/\|x\| = O_n(\kappa^2\varepsilon), \tag{40}$$

where $\kappa = \kappa(A)$, provided $\kappa^2\varepsilon \ll 1$. The residual $r = A\tilde{x} - b$ should satisfy

$$\frac{\|r\|}{\|A\|\|x\|} = O_n(\kappa\varepsilon), \tag{41}$$

because $\|A^Tr\| = \|A^TA\tilde{x} - A^Tb\| = O_n(\varepsilon\|A^TA\|\|x\|)$. From (40), the method is *weakly stable* (according to Definition 2), although we can not expect the stronger bound (3) to be satisfied.

The bounds (40–41) are similar to those usually given for the method of normal equations [36], not those usually given for the method of semi-normal equations [5, 61, 66]. This is because, in applications of the semi-normal equations, it is usually assumed that $\tilde{R}$ is computed via an orthogonal factorization of A, so there is a matrix $\widehat{A}$ such that $\tilde{R}^T\tilde{R} = \widehat{A}^T\widehat{A}$ and

$$\|\widehat{A} - A\|/\|A\| = O_n(\varepsilon). \tag{42}$$

However, in our case we only have $\|\tilde{R}^T\tilde{R} - R^TR\|/\|R^TR\| = O_n(\varepsilon)$, which implies the weaker bound

$$\|\widehat{A} - A\|/\|A\| = O_n(\kappa\varepsilon) \tag{43}$$

by Stewart's perturbation analysis [71, 72].

An alternative to the use of (39) was suggested by Paige [61]: compute R such that $R^TR = AA^T$, solve $R^TRw = b$ by solving two triangular systems, then set $x = A^Tw$. We prefer to use (39) because it is also applicable in the rectangular (least squares) case – see §6.2.

6.1 Storage Requirements

The algorithm described in §6 for the solution of the semi-normal equations (39) requires working storage $O(n^2)$ words, because the upper triangular matrix R is not Toeplitz. However, it is possible to reduce the storage requirement to $O(n)$ words. Recall that $\tilde{R}$ is generated row by row. Thus, we can solve $\tilde{R}^Tw = d$ as $\tilde{R}$ is generated, accumulating the necessary inner products with $O(n)$ storage. We now have to solve $\tilde{R}x = w$ without having saved $\tilde{R}$. Provided the $O(n)$ rotations defining the updates and downdates have been saved, we can regenerate the rows of $\tilde{R}$ in reverse order $(n,\ n-1,\ \ldots,\ 1)$ and solve $\tilde{R}x = w$ as this is done. A similar idea was used in [15] to save storage in a systolic implementation of the Bareiss algorithm.

The regenerated matrix $\tilde{R}'$ differs slightly from $\tilde{R}$ because of rounding errors. Numerical experiments suggest (though we have not proved) that $\|\tilde{R}' - \tilde{R}\| = O_n(\kappa\varepsilon)$. If true, this would imply that the method is weakly stable, and (40) would hold, but the right hand side of (41) would need to be multiplied by κ.

An alternative is to use an idea which was suggested by Griewank [39] in a different context. Suppose we have a procedure $\mathcal{F}(k)$ which generates k consecutive rows of $\tilde{R}$ in a forward direction (say rows $d+1, \ldots, d+k$), and $\mathcal{B}(k)$ which generates k consecutive rows in a backward direction (say rows $d+k, \ldots, d+1$). In each case $O(n)$ words of storage suffice for the initial conditions. Then $\mathcal{B}(2k)$ can be defined recursively –

1. Generate rows $d+1, \ldots, d+k$ using $\mathcal{F}(k)$, with row d for initial conditions if $d > 0$, saving row $d+k$.

2. Generate rows $d+2k, \ldots, d+k+1$ using $\mathcal{B}(k)$ with row $d+k$ for initial conditions.

3. Generate rows $d+k, \ldots, d+1$ using $\mathcal{B}(k)$ with row d for initial conditions if $d > 0$.

From the discussion in §5, $\mathcal{F}(k)$ requires $O(kn)$ operations and $O(n)$ storage. Thus, we can prove by induction that $\mathcal{B}(k)$ requires $O(kn\log k)$ operations and $O(n\log k)$ storage. Overall, to generate the n rows of $\tilde{R}$ in reverse order takes $O(n^2\log n)$ operations and $O(n\log n)$ storage. Thus, at the cost of a factor $O(\log n)$ in the operation count, we can reduce the storage requirements from $O(n^2)$ to $O(n\log n)$. The numerical properties of this method are exactly the same as those of the method which uses $O(n^2)$ storage, since exactly the same rows of $\tilde{R}$ are computed. The scheme described here is not optimal in its use of storage, but is within a factor of two of the optimal scheme described in [39].

6.2 Toeplitz Least Squares Problems

If $A \in \Re^{m\times n}$ is Toeplitz with full rank n, then the semi-normal equations (39) may be used to solve the least squares problem

$$\min \|Ax - b\|_2. \tag{44}$$

The use of semi-normal equations for the general full-rank linear least squares problem is discussed in detail by Björck [5], and the only significant difference in our case is that the bound (43) holds instead of (42), so an additional factor κ appears in some of the terms in the error bounds.

6.3 Operation Counts

We briefly estimate the number of arithmetic operations required by the BBH algorithm and some of its competitors. We assume that the Toeplitz matrix A is real. Some of the operation counts can be reduced by using fast Givens transformations [31, 36], but for simplicity we ignore this possibility. We only count multiplications; the number of additions is comparable.

For simplicity, first consider the case $m = n$. In the BBH algorithm, the computation of R takes $7n^2 + O(n)$ multiplications. The computation of $A^T b$ takes n^2 multiplications if done in the obvious way, or $O(n \log n)$ multiplications if the FFT is used. Solving two triangular systems takes $n^2 + O(n)$ multiplications. Thus, to solve a Toeplitz linear system by the method of §6 takes $9n^2 + O(n)$ multiplications, or $8n^2 + O(n \log n)$ if the FFT is used. This is much cheaper than the $19n^2 + O(n)$ multiplications of the BBH QR factorisation algorithm given in §3 of [8], which computes Q explicitly. Thus, considering both speed and stability, it is best to avoid the computation of Q.

For the method of Nagy [60], the multiplication count is $16n^2 + O(n)$, and for Cybenko's method [23] it is $23n^2 + O(n)$. The method TpH of [34] requires $21n^2/2 + O(n \log n)$ real multiplications, and the method GKO of [34] requires $13n^2/2 + O(n \log n)$ complex multiplications. Thus, the method of §6 should be faster than any of these methods, although GKO may be competitive if the Toeplitz matrix A complex.

For the rectangular case ($m \geq n$), the corresponding multiplication counts (omitting low-order terms and assuming that the FFT is not used) are:

$2mn + 7n^2$ for the method of §6;
$2mn + 9n^2$ for the method of Nagy [60] using the semi-normal equations;
$2mn + 14n^2$ for the method of Nagy [60] using inverse factorisation (algorithm IF);
$9mn + 14n^2$ for the method of Cybenko [23];
$13mn + 7n^2$ for the method of [8], computing Q explicitly.

For details of the components making up these operation counts, see Table 4.1 of [60].

6.4 Iterative Refinement

Iterative refinement (sometimes called iterative improvement) can be used to improve an approximate solution $\tilde{x}$ to the linear system (37) or the linear least squares problem (44). In practice this gives an accurate solution in a small number of iterations so long as the residual is computed accurately and the working precision is sufficient to ensure convergence. For details see [5, 36, 37, 48, 69, 84].

A related idea is to improve the accuracy of $\tilde{R}_b$ and $\tilde{R}_t$ by using Björck's "Corrected Semi-Normal Equations" [5, 63] or Foster's scheme of iterative improvement [26]. However, if the aim is simply to solve a linear system, then it is more economical to apply iterative refinement directly to the system.

6.5 Ill-conditioned Problems

For very ill-conditioned Toeplitz linear systems and least squares problems, it may be desirable to use regularisation, as discussed in §5 of [60]. We do not consider this here, except to note that our algorithm can easily be modified to compute the Cholesky factorisation of $A^T A + \alpha I$, where α is a positive regularisation parameter. Only small changes in equations (20–23) are required.

7 Numerical Results

The algorithm described in §§5–6 has been implemented in Pascal on an IBM PC and DEC VAX. In Table 1 we give some results for randomly chosen $n \times n$ Toeplitz systems on a DEC

VAX with $\varepsilon = 2^{-56} \simeq 1.4 \times 10^{-17}$. The elements a_k defining the Toeplitz matrix

$$A = \begin{pmatrix} a_0 & \cdots & a_{n-1} \\ \vdots & \ddots & \vdots \\ a_{1-n} & \cdots & a_0 \end{pmatrix}$$

were chosen from a normal distribution with specified mean μ and standard deviation σ. (The condition number $\kappa(A)$ tends to increase with $|\mu/\sigma|$.) The solution vector x was chosen with normally distributed components (mean 0) and the vector b computed from $b \leftarrow Ax$. A consequence is that $\|x\|/\|b\|$ is unlikely to be large, even if A is poorly conditioned, but this is typical of most applications. Table 1 gives the condition number

$$\kappa_1(R) = \|R\|_1 \cdot \|R^{-1}\|_1,$$

which is a rough approximation to $\kappa_2(R) = \kappa_2(A)$ (the 1-norm was used for computational convenience).

Table 1 gives

$$\begin{aligned} e_1 &= \frac{\|\tilde{R}^T\tilde{R} - A^TA\|_1}{\varepsilon\|A^TA\|_1}, \\ e_2 &= \frac{\|\tilde{x} - x\|_2}{\varepsilon\kappa_1(R)^2\|x\|_2}, \\ e_3 &= \frac{\|r\|_2}{\varepsilon\kappa_1(R)\|A\|_1\|x\|_2}, \end{aligned}$$

where $r = A\tilde{x} - b$. From Theorem 1 and the bounds (40–41), we expect these quantities to be bounded by (low degree) polynomials in n. The results confirm this.

For comparison, the last column of Table 1 gives e_3^c, the value of the normalised residual e_3 obtained via Cholesky factorization of A^TA. It can be seen that e_3 is not much larger than e_3^c.

We also tried Toeplitz matrices A with some singular principal submatrices (e.g. A with $a_{-1} = a_0 = a_1$). The results were similar to those given in Table 1.

8 Conclusion

The method described in §6 for the solution of general nonsingular Toeplitz or Hankel linear systems[3] requires $O(n^2)$ operations, is weakly stable, and makes no assumption about the conditioning of submatrices of A. We do not know any other methods which have been proved to be stable or weakly stable and have worst-case time bound $O(n^2)$. Algorithms which involve pivoting and/or look-ahead [18, 27, 41, 43, 76] may work well in practice, but seem to require worst-case overhead $O(n^3)$ to ensure stability.

Our method should be faster than $O(n^3)$ methods which ignore the Toeplitz structure, even if the working precision has to be increased (i.e. ε reduced) in our method to ensure that $\kappa^2\varepsilon \ll 1$. Storage requirements are $O(n^2)$, but may be reduced to $O(n \log n)$ or $O(n)$ as discussed in §6.1.

We have not looked in detail at all the fast Toeplitz QR factorization algorithms mentioned in §1. It is quite likely that some of these give weakly stable algorithms for the computation of the Cholesky factor R of A^TA, but no proofs have been published.

It remains an open question whether there is a fast Toeplitz QR algorithm which is backward stable for the computation of Q and R (or R^{-1}). Such an algorithm would give a stable algorithm for the solution of $Ax = b$ without recourse to the semi-normal equations.

[3] Similar remarks apply for full-rank Toeplitz or Hankel least squares problems.

n	μ/σ	$\kappa_1(R)$	e_1	e_2	e_3	e_3^c
50	0.0	3.3′3	6.7	9.2′-4	5.2′-3	1.4′-3
50	1.0	3.6′3	1.7′1	6.2′-3	2.7′-2	2.1′-3
50	1.0′1	9.8′2	4.0′1	3.1′-1	3.7′-1	9.4′-2
50	1.0′2	1.0′4	1.0′2	2.0′-1	5.3′-1	7.0′-2
50	1.0′3	1.5′5	3.4′1	4.6′-1	3.0′-1	4.6′-2
50	1.0′4	9.1′5	1.0′2	1.0	1.2	9.2′-2
50	1.0′5	6.3′6	9.2	1.4′-1	1.6′-1	2.9′-1
100	0.0	1.2′3	1.4′1	6.2′-4	4.7′-3	1.6′-3
100	1.0	1.8′3	1.5′1	5.2′-4	4.4′-3	3.1′-3
100	1.0′1	5.2′3	8.7′1	8.5′-2	1.6′-1	3.3′-2
100	1.0′2	3.3′4	1.2′2	4.2′-1	3.7′-1	3.5′-2
100	1.0′3	4.0′5	1.4′2	2.2′-1	6.6′-1	4.8′-2
100	1.0′4	8.8′6	1.5′2	1.0	8.9′-1	2.6′-2
100	1.0′5	7.8′6	3.8′1	5.5′-1	8.4′-1	1.4′-1
200	0.0	5.7′3	1.9′1	1.3′-4	1.7′-3	8.7′-4
200	1.0	1.2′6	8.4′1	1.0′-5	1.9′-4	1.6′-3
200	1.0′1	5.6′5	1.6′1	3.6′-3	7.0′-3	4.7′-3
200	1.0′2	2.4′4	2.1′2	3.0	2.7	1.4′-1
200	1.0′3	7.9′5	1.4′1	8.2′-2	6.6′-2	5.2′-2
200	1.0′4	5.5′6	2.8′2	6.1′-1	5.0′-1	4.3′-2
200	1.0′5	1.3′8	3.6′2	1.8′-1	3.8′-1	4.3′-2

Table 1: Weakly stable solution of Toeplitz systems

Acknowledgements

Thanks to Greg Ammar, George Cybenko, Lars Eldén, Andreas Griewank, Georg Heinig, Franklin Luk, Haesun Park, and Douglas Sweet, for helpful comments on earlier versions of this paper. James Bunch, Roland Freund, Martin Gutknecht, Vadim Olshevsky and James Varah kindly sent us relevant reprints and preprints.

The literature on Toeplitz and related linear systems is voluminous, so the following bibliography is by no means complete. We have attempted to include a representative sample of the recent literature and some important historical references.

References

[1] G. S. Ammar and W. B. Gragg, "Superfast solution of real positive definite Toeplitz systems", *SIAM J. Matrix Anal. Appl.* 9 (1988), 61–76.

[2] E. H. Bareiss, "Numerical solution of linear equations with Toeplitz and vector Toeplitz matrices", *Numerische Mathematik* 13 (1969), 404–424.

[3] C. H. Bischof, C.-T. Pan and P. T. P. Tang, "A Cholesky up- and downdating algorithm for systolic and SIMD architectures", *SIAM J. Sci. Comput.* 14 (1993), 670-676.

[4] R. R. Bitmead and B. D. O. Anderson, "Asymptotically fast solution of Toeplitz and related systems of equations", *Linear Algebra Appl.* 34 (1980), 103–116.

[5] Å. Björck, "Stability analysis of the method of semi-normal equations for linear least squares problems", *Linear Algebra Appl.* 88/89 (1987), 31–48.

[6] Å. Björck, "Error analysis of least squares algorithms", *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms* (edited by G. H. Golub and P. Van Dooren), Springer-Verlag, 1991, 41–73.

[7] 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. Computing*, 8 (1987), 210–220.

[8] A. W. Bojanczyk, R. P. Brent and F. R. de Hoog, "QR factorization of Toeplitz matrices", *Numerische Mathematik* 49 (1986), 81–94.

[9] 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, ANU, August 1991, 23 pp.

[10] A. W. Bojanczyk, R. P. Brent, F. R. de Hoog and D. R. Sweet, *On the Stability of the Bareiss and Related Toeplitz Factorization Algorithms*, Report TR-CS-93-14, CS Lab, ANU, November 1993, 18 pp. To appear in *SIAM J. Matrix Anal. Appl.*

[11] A. W. Bojanczyk and A. O. Steinhardt, "Matrix downdating techniques for signal processing", *Proceedings of the SPIE Conference on Advanced Algorithms and Architectures for Signal Processing III*, volume 975 (1988), 68–75.

[12] A. W. Bojanczyk and A. O. Steinhardt, "Stability analysis of a Householder-based algorithm for downdating the Cholesky factorization", *SIAM J. Sci. Statist. Comput.* 12 (1991), 1255–1265.

[13] R. P. Brent, "Parallel algorithms for Toeplitz systems", *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms* (edited by G. H. Golub and P. Van Dooren), Springer-Verlag, 1991, 75–92.

[14] R. P. Brent, F. G. Gustavson and D. Y. Y. Yun, "Fast solution of Toeplitz systems of equations and computation of Padé approximants", *J. Algorithms* 1 (1980), 259–295.

[15] R. P. Brent, H. T. Kung and F. T. Luk "Some linear-time algorithms for systolic arrays", *Information Processing 83* (edited by R. E. A. Mason), North-Holland, Amsterdam, 1983, 865–876.

[16] J. R. Bunch, "Stability of methods for solving Toeplitz systems of equations", *SIAM J. Sci. Statist. Comput.* 6 (1985), 349–364.

[17] J. R. Bunch, "The weak and strong stability of algorithms in numerical linear algebra", *Linear Algebra Appl.* 88/89 (1987), 49–66.

[18] T. F. Chan and P. C. Hansen, "A look-ahead Levinson algorithm for indefinite Toeplitz systems", *SIAM J. Matrix Anal. Appl.* 13 (1992), 490–506.

[19] T. F. Chan and P. C. Hansen, "A look-ahead Levinson algorithm for general Toeplitz systems", *IEEE Trans. Signal Process.* 40 (1992), 1079–1090.

[20] J. Chun and T. Kailath, "Divide-and-conquer solutions of least-squares problems for matrices with displacement structure", *SIAM J. Matrix Anal. Appl.* 12 (1991), 128-145.

[21] J. Chun, T. Kailath and H. Lev-Ari, "Fast parallel algorithms for QR and triangular factorization", *SIAM J. Sci. Statist. Computing* 8 (1987), 899–913.

[22] G. Cybenko, "The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations", *SIAM J. Sci. Statist. Computing* 1 (1980), 303–319.

[23] G. Cybenko, "Fast Toeplitz orthogonalization using inner products", *SIAM J. Sci. Statist. Computing* 8 (1987), 734–740.

[24] J. Dongarra, J. R. Bunch, C. B. Moler and G. W. Stewart, *Linpack Users Guide*, SIAM Publications, Philadelphia, Pennsylvania, 1978.

[25] R. Fletcher and M. J. D. Powell, "On the modification of LDL^T factorizations", *Math. Comp.* 28 (1974), 1067–1087.

[26] L. Foster, "Modifications of the normal equations method that are numerically stable", *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms* (edited by G. H. Golub and P. Van Dooren), Springer-Verlag, 1991, 501–512.

[27] R. W. Freund, *A Look-ahead Bareiss Algorithm for General Toeplitz Matrices*, Technical Memorandum 11274–930106–02, AT&T Bell Laboratories, Murray Hill, New Jersey, 1993.

[28] R. W. Freund, *A Look-ahead Schur-type Algorithm for Solving General Toeplitz Systems*, AT&T Bell Laboratories Numerical Analysis Manuscript 93–09, Murray Hill, New Jersey, June 1993. Also *Zeitschrift für Angewandte Mathematik und Mechanik*, to appear.

[29] R. W. Freund and H. Zha, "Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems", *Linear Algebra Appl.* 188/189 (1993), 255–303.

[30] R. W. Freund and H. Zha, "A look-ahead algorithm for the solution of general Hankel systems", *Numer. Math.* 64 (1993), 295–321.

[31] M. Gentleman, "Least squares computations by Givens transformations without square roots", *J. Inst. Math. Appl.* 12 (1973), 329–336.

[32] P. E. Gill, G. H. Golub, W. Murray and M. A. Saunders, "Methods for modifying matrix factorizations", *Math. Comp.* 28 (1974), 505–535.

[33] I. Gohberg (editor), *I. Schur Methods in Operator Theory and Signal Processing* (Operator Theory: Advances and Applications, Volume 18), Birkhäuser Verlag, Basel, 1986.

[34] I. Gohberg, T. Kailath and V. Olshevsky, *Gaussian Elimination with Partial Pivoting for Structured Matrices*, preprint, 14 May 1994.

[35] G. H. Golub, "Numerical methods for solving linear least squares problems", *Numer. Math.* 7 (1965), 206–216.

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

[37] G. H. Golub and J. H. Wilkinson, "Note on iterative refinement of least squares solution", *Numer. Math.* 9 (1966), 139–148.

[38] N. Gould, "On growth in Gaussian elimination with complete pivoting", *SIAM J. Matrix Anal. Appl.* 12 (1991), 354–361.

[39] A. Griewank, "Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation", *Optimization Methods and Software* 1 (1992), 35–54.

[40] M. H. Gutknecht, "Stable row recurrences for the Padé table and generically superfast look-ahead solvers for non-Hermitian Toeplitz systems", *Linear Alg. Appl.* 188/189 (1993), 351–421.

[41] M. H. Gutknecht and M. Hochbruck, *Look-ahead Levinson and Schur algorithms for non-Hermitian Toeplitz Systems*, IPS Research Report 93–11, ETH, Zürich, August 1993.

[42] M. H. Gutknecht and M. Hochbruck, *The Stability of Inversion Formulas for Toeplitz Matrices*, IPS Research Report 93–13, ETH, Zürich, October 1993.

[43] P. C. Hansen and H. Gesmar, "Fast orthogonal decomposition of rank deficient Toeplitz matrices", *Numerical Algorithms* 4 (1993), 151–166.

[44] G. Heinig, P. Jankowski and K. Rost, "Fast inversion of Toeplitz-plus-Hankel matrices", *Numer. Math.* 52 (1988), 665–682.

[45] G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators* (Operator Theory: Advances and Applications, Volume 13), Birkäuser Verlag, Basel, 1984.

[46] D. J. Higham and N. J. Higham, "Backward error and condition of structured linear systems", *SIAM J. Matrix Anal. Appl.* 13 (1992), 162-175.

[47] F. R. de Hoog, "A new algorithm for solving Toeplitz systems of equations", *Linear Algebra Appl.* 88/89 (1987), 123–138.

[48] M. Jankowski and H. Wozniakowski, "Iterative refinement implies numerical stability", *BIT* 17 (1977) 303–311.

[49] T. Kailath, "A theorem of I. Schur and its impact on modern signal processing", in [33], 9–30.

[50] T. Kailath and J. Chun, "Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices", *SIAM J. Matrix Anal. Appl.* 15 (1994), 114–128.

[51] T. Kailath, S. Y. Kung and M. Morf, "Displacement ranks of matrices and linear equations", *J. Math. Anal. Appl.* 68 (1979), 395–407.

[52] T. Kailath and A. H. Sayed, "Displacement structure: theory and applications", *SIAM Review*, to appear.

[53] T. Kailath, A. Vieira and M. Morf, "Inverses of Toeplitz operators, innovations and orthogonal polynomials", *SIAM Review* 20 (1978), 106–119.

[54] A. N. Kolmogorov, "Interpolation and extrapolation of stationary random sequences", *Izvestia Akad. Nauk SSSR* 5 (1941), 3–11 (in Russian); German summary, *ibid* 11–14.

[55] N. Levinson, "The Wiener RMS (Root-Mean-Square) error criterion in filter design and prediction", *J. Math. Phys.* 25 (1947), 261–278.

[56] F. T. Luk and S. Qiao, "A fast but unstable orthogonal triangularization technique for Toeplitz matrices", *Linear Algebra Appl.* 88/89 (1987), 495–506.

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

[58] M. Morf, "Doubling algorithms for Toeplitz and related equations", *Proc. ICASSP-80*, IEEE, 1980, 954–959.

[59] B. R. Musicus, *Levinson and fast Choleski algorithms for Toeplitz and almost Toeplitz matrices*, Tech. Report, Electronics Research Lab., MIT, Cambridge, Mass., 1984.

[60] J. G. Nagy, "Fast inverse QR factorization for Toeplitz matrices", *SIAM J. Sci. Comput.* 14 (1993), 1174–1193.

[61] C. C. Paige, "An error analysis of a method for solving matrix equations", *Math. Comp.* 27 (1973), 355–359.

[62] C.-T. Pan, "A modification to the Linpack downdating algorithm", *BIT* 30 (1990), 707–722.

[63] H. Park and L. Eldén, *Fast and Accurate Triangularization of Toeplitz Matrices*, preprint, 14 April 1993.

[64] S. Qiao, "Hybrid algorithm for fast Toeplitz orthogonalization", *Numer. Math.* 53 (1988), 351–366.

[65] J. Rissanen, "Solution of linear equations with Hankel and Toeplitz matrices", *Numer. Math.* 22 (1974), 361–366.

[66] M. A. Saunders, *Large-Scale Linear Programming Using the Cholesky Factorization*, Report CS 252, Computer Science Dept., Stanford University, Stanford, Calif., Jan. 1972.

[67] I. Schur, "Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind", *J. fur die Reine und Angewandte Mathematik* 147 (1917), 205–232. English translation in [33], 31–59.

[68] H. Sexton, M. Shensa and J. Spenser, "Remarks on a displacement-rank inversion method for Toeplitz systems", *Linear Algebra Appl.* 45 (1982), 127–130.

[69] R. D. Skeel, "Iterative refinement implies numerical stability for Gaussian elimination", *Math. Comp.* 35 (1980), 817–832.

[70] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[71] G. W. Stewart, "Perturbation bounds for the QR factorization of a matrix", *SIAM J. Numer. Anal.* 14 (1977), 509–518.

[72] G. W. Stewart, "The effect of rounding error on an algorithm for downdating a Cholesky factorization", *J. Inst. Math. Appl.* 23 (1979), 203-213.

[73] D. R. Sweet, *Numerical Methods for Toeplitz Matrices*, PhD thesis, University of Adelaide, 1982.

[74] D. R. Sweet, "Fast Toeplitz orthogonalization", *Numerische Mathematik* 43 (1984), 1–21.

[75] D. R. Sweet, "The propagation of rounding errors in pivoting techniques for Toeplitz matrix solvers", *Proc. Int. Conf. on Comput. Techniques and Appls. (CTAC-89)*, Hemisphere, New York, 1990, 623–630.

[76] D. R. Sweet, "The use of pivoting to improve the numerical performance of algorithms for Toeplitz matrices", *SIAM J. Matrix Anal. Appl.* 14 (1993), 468–493.

[77] G. Szegö, *Orthogonal Polynomials*, AMS Colloquium publ. XXIII, American Mathematical Society, Providence, Rhode Island, 1939.

[78] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices", *J. SIAM (SIAM J. Appl. Math.)* 12 (1964), 515–522.

[79] J. M. Varah, *Backward Error Estimates for Toeplitz Systems*, preprint, Computer Science Department, University of British Columbia, Vancouver, Sept. 1992.

[80] G. A. Watson, "An algorithm for the inversion of block matrices of Toeplitz form", *J. Assoc. Comput. Mach.* 20 (1973), 409–415.

[81] N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications*, Technology Press and Wiley, New York, 1949 (originally published in 1941 as a Technical Report).

[82] J. H. Wilkinson, "Error analysis of direct methods of matrix inversion", *J. Assoc. Comput. Mach.* 8 (1961), 281–330.

[83] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

[84] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.

[85] S. Zohar, "Toeplitz matrix inversion: the algorithm of W. F. Trench", *J. Assoc. Comput. Mach.* 16 (1969), 592–601.

[86] S. Zohar, "The solution of a Toeplitz set of linear equations", *J. Assoc. Comput. Mach.* 21 (1974), 272–276.