STABILITY OF FAST ALGORITHMS FOR STRUCTURED LINEAR SYSTEMS*

RICHARD P. BRENT†

Abstract. We survey the numerical stability of some fast algorithms for solving systems of linear equations and linear least squares problems with a low displacement-rank structure. For example, the matrices involved may be Toeplitz or Hankel. We consider algorithms which incorporate pivoting without destroying the structure, and describe some recent results on the stability of these algorithms. We also compare these results with the corresponding stability results for the well known algorithms of Schur/Bareiss and Levinson, and for algorithms based on the semi-normal equations.

Key words. Bareiss algorithm, Levinson algorithm, Schur algorithm, Toeplitz matrices, displacement rank, generalized Schur algorithm, numerical stability.

AMS subject classifications. 65F05, 65G05, 47B35, 65F30

1. Motivation. The standard direct method for solving dense $n \times n$ systems of linear equations is Gaussian elimination with partial pivoting. The usual implementation requires of order n^3 arithmetic operations.

In practice, linear systems often arise from some physical system and have a structure which is a consequence of the physical system. For example, time-invariant physical systems often give rise to *Toeplitz* systems of linear equations (§3.2). An $n \times n$ Toeplitz matrix is a dense matrix because it generally has n^2 nonzero elements. However, it is determined by only $O(n)$ parameters (in fact, by the $2n-1$ entries in its first row and column). Similar examples are Hankel, Cauchy, Toeplitz-plus-Hankel, and Vandermonde matrices [38, 60].

When solving such a structured linear system it is possible to ignore the structure, and this may have advantages if standard software is available and n is not too large. However, if n is large or if many systems have to be solved, perhaps with real-time constraints (e.g. in radar and sonar applications), then it is desirable to take advantage of the structure. The primary advantage to be gained is that the time to solve a linear system is reduced by a factor of order n to $O(n^2)$. Storage requirements may also be reduced by a factor of order n, from $O(n^2)$ to $O(n)$.

Most papers concerned with algorithms for structured linear systems concentrate on the speed (usually measured in terms of the number of arithmetic operations required) and ignore questions of numerical accuracy. However, it is dangerous to use fast algorithms without considering their numerical properties. There is no point in obtaining an answer quickly if it is much less accurate than is justified by the data.

In this paper we reverse the usual emphasis on speed, and concentrate on the numerical properties of fast algorithms. Because there are many classes of structured matrices, and an ever increasing number of fast algorithms, we can not attempt to be comprehensive. Our aim is to introduce the reader to the subject, illustrate some of the main ideas, and provide pointers to the literature.

*Appeared as Report TR-CS-97-18, ANU, September 1997. Preliminary version of a chapter to appear in *Stability of Fast Methods for Linear Systems with Structure* (editors, Ali H. Sayed and Thomas Kailath), SIAM, Philadelphia, 1999, 103–116.

The subject of numerical stability/instability of fast algorithms is confused for several reasons:

1. Structured matrices are often very ill-conditioned. For example, the Hilbert matrix [50], defined by $a_{i,j} = 1/(i + j - 1)$, is often used as an example of a poorly conditioned matrix [29]. This is a typical example of a Hankel matrix. Reversing the order of the rows gives a Toeplitz matrix.

2. The solution might be less sensitive to structured perturbations (i.e. perturbations which are physically plausible because they preserve the structure) than to general (unstructured) perturbations. The effect of rounding errors in methods which ignore the structure is generally equivalent to the introduction of unstructured perturbations. Ideally we should use a method which only introduces structured perturbations, but this property often does not hold, or is difficult to prove, even for methods which take advantage of the structure to reduce the number of arithmetic operations.

3. The error bounds which can be proved are usually much weaker than what is observed on "real" or "random" examples. Thus, methods which are observed to work well in practice can not always be guaranteed, and it is hard to know if it is just the analysis which is weak or if the method fails in some rare cases.

4. An algorithm may perform well on special classes of structured matrices, e.g. positive definite matrices or matrices with positive reflection coefficients, but perform poorly or break down on broader classes of structured matrices.

1.1. Outline. Different authors have given different (and sometimes inconsistent) definitions of *stability* and *weak stability*. We shall follow Bunch [14, 15]. For completeness, our definitions are given in §2.

The concept of *displacement rank*, discussed in §3, may be used to unify the discussion of many algorithms for structured matrices [52, 53, 54]. It is well known that systems of n linear equations with a low displacement rank (e.g. Toeplitz or Hankel matrices) can be solved in $O(n^2)$ arithmetic operations. Asymptotically faster algorithms with time bound $O(n \log^2 n)$ exist [13], but are not considered here because their numerical properties are generally poor and the constant factors hidden in the "O" notation are large [1].

For positive definite Toeplitz matrices, the first $O(n^2)$ algorithms were introduced by Kolmogorov [56], Wiener [78] and Levinson [57]. These algorithms are related to recursions of Szegö [74] for polynomials orthogonal on the unit circle. Another class of $O(n^2)$ algorithms, e.g. the Bareiss algorithm [2], are related to Schur's algorithm for finding the continued fraction representation of a holomorphic function in the unit disk [64]. This class can be generalized to cover unsymmetric matrices and more general "low displacement rank" matrices [54]. In §§4–6 we consider the numerical stability of some of these algorithms. The GKO-Cauchy and GKO-Toeplitz algorithms are discussed in §4. The Schur/Bareiss algorithm for positive definite matrices is considered in §5.1, and generalized Schur algorithms are mentioned in §5.2. In §6 we consider fast orthogonal factorization algorithms and the fast solution of structured least squares problems.

Algorithms for Vandermonde and many other classes of structured matrices are not considered here – we refer to the extensive survey by Olshevsky [60]. Also, we have omitted any discussion of fast "lookahead" algorithms [18, 19, 30, 31, 42, 43, 44, 45, 72] because, although such algorithms often succeed in practice, in the worst case they require of order n^3 operations.

Much work has been done on iterative methods for Toeplitz and related systems. Numerical stability is not a major problem with iterative methods, but the speed of convergence depends on a good choice of preconditioner. We do not consider iterative methods here, except to mention iterative refinement as a way of improving the accuracy of weakly stable direct methods.

1.2. Notation. In the following, R denotes a structured matrix, T is a Toeplitz or Toeplitz-type matrix, P is a permutation matrix, L is lower triangular, U is upper triangular, and Q is orthogonal. In error bounds, $O_n(\varepsilon)$ means $O(\varepsilon f(n))$, where $f(n)$ is a polynomial in n. We do not usually try to specify the polynomial $f(n)$ precisely because it depends on the norm that is used to measure the error and on many unimportant details of the implementation.

2. 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 $Ax = b$. There are many definitions of numerical stability in the literature, for example [3, 4, 8, 9, 14, 24, 38, 51, 59, 61, 65]. Definitions 2.1 and 2.2 below are taken from Bunch [15].

DEFINITION 2.1. *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 2.1 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 [79, 80, 81]. We interpret "close" to mean close in the relative sense in some norm, i.e.

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

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 [15] calls *strong stability.* For a discussion of the difference between stability and strong stability for Toeplitz algorithms, see [34, 35, 48, 76].

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

$$\|\tilde{x} - x\|/\|x\| = O_n(\kappa\varepsilon). \tag{2.1}$$

For more precise results, see Bunch [15] and Wilkinson [79].

As an example, consider the method of Gaussian elimination. Wilkinson [79] shows that

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

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 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 [79] showed that $g(n) \le n^{(\log n)/4+O(1)}$, and Gould [39] 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 2.1, 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* [81].

Although stability is desirable, it is more than we can prove for many useful algorithms. Thus, following Bunch [15], we define the (weaker, but still useful) property of *weak stability.*

DEFINITION 2.2. *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 2.2, 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 (2.1), stability implies weak stability.

Define the *residual* $r = A\tilde{x} - b$. It is well-known [80] that

$$\frac{1}{\kappa}\frac{\|r\|}{\|b\|} \le \frac{\|\tilde{x} - x\|}{\|x\|} \le \kappa\frac{\|r\|}{\|b\|}. \tag{2.2}$$

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 2.3. *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.*

2.1. Example: orthogonal factorization. To illustrate the concepts of stability and weak stability, 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 [38]. It can be shown [80] that the computed matrices $\tilde{Q}$, $\tilde{R}$ satisfy

$$\widehat{A} = \widehat{Q}\tilde{R} \tag{2.3}$$

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 [36, 66, 81].

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 (2.3) holds (with $\widehat{A}$ close to A and some $\widehat{Q}$ with $\widehat{Q}^T\widehat{Q} = I$) unless A is well-conditioned [67]. By analogy with Definition 2.3 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.

3. Classes of Structured Matrices. Structured matrices R satisfy a *Sylvester equation* which has the form

$$\nabla_{\{A_f, A_b\}}(R) = A_fR - RA_b = \Phi\Psi\ , \tag{3.1}$$

where A_f and A_b have some simple structure (usually banded, with 3 or fewer full diagonals), Φ and Ψ are $n \times \alpha$ and $\alpha \times n$ respectively, and α is some fixed integer. The pair of matrices (Φ, Ψ) is called the $\{A_f, A_b\}$-*generator* of R.

α is called the $\{A_f, A_b\}$-*displacement rank* of R. We are interested in cases where α is small (say at most 4).

For a discussion of the history and some variants of (3.1), see [33] and the references given there, in particular [47, 53].

3.1. Cauchy and Cauchy-type matrices. Particular choices of A_f and A_b lead to definitions of basic classes of matrices. Thus, for a Cauchy matrix

$$C(\mathbf{t}, \mathbf{s}) = \left[\frac{1}{t_i - s_j} \right]_{ij} ,$$

we have

$$A_f = D_t = \operatorname{diag}(t_1, t_2, \ldots, t_n) ,$$

$$A_b = D_s = \operatorname{diag}(s_1, s_2, \ldots, s_n)$$

and

$$\Phi^T = \Psi = [1, 1, \ldots, 1] .$$

As a natural generalization, we can take Φ and Ψ to be any rank-α matrices, with A_f, A_b as above. Then a matrix R satisfying the Sylvester equation (3.1) is said to be a *Cauchy-type* matrix.

3.2. Toeplitz matrices. For a Toeplitz matrix $T = [t_{ij}] = [a_{i-j}]$, we take

$$A_f = Z_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} , \quad A_b = Z_{-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & & & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \ddots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} ,$$

$$\Phi = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_0 & a_{1-n} + a_1 & \cdots & a_{-1} + a_{n-1} \end{bmatrix}^T ,$$

and

$$\Psi = \begin{bmatrix} a_{n-1} - a_{-1} & \cdots & a_1 - a_{1-n} & a_0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} .$$

We can generalize to *Toeplitz-type* matrices by taking Φ and Ψ to be general rank-α matrices.

4. Structured Gaussian Elimination. Let an input matrix, R_1, have the partitioning

$$R_1 = \begin{bmatrix} d_1 & \mathbf{w}_1^T \\ \mathbf{y}_1 & \tilde{R}_1 \end{bmatrix} .$$

The first step of normal Gaussian elimination is to premultiply R_1 by

$$\begin{bmatrix} 1 & \mathbf{0}^T \\ -\mathbf{y}_1/d_1 & I \end{bmatrix} ,$$

which reduces it to

$$\begin{bmatrix} d_1 & \mathbf{w}_1^T \\ \mathbf{0} & R_2 \end{bmatrix} ,$$

where

$$R_2 = \tilde{R}_1 - \mathbf{y}_1\mathbf{w}_1^T/d_1$$

is the *Schur complement* of d_1 in R_1. At this stage, R_1 has the factorization

$$R_1 = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{y}_1/d_1 & I \end{bmatrix} \begin{bmatrix} d_1 & \mathbf{w}_1^T \\ \mathbf{0} & R_2 \end{bmatrix} .$$

One can proceed recursively with the Schur complement R_2, eventually obtaining a factorization $R_1 = LU$.

The key to *structured* Gaussian elimination is the fact that the displacement structure is preserved under Schur complementation, and that the generators of the Schur complement R_{k+1} can be computed from the generators of R_k in $O(n)$ operations.

Row and/or column interchanges destroy the structure of matrices such as Toeplitz matrices. However, if A_f is diagonal (which is the case for Cauchy and Vandermonde type matrices), then *the structure is preserved under row permutations.*

This observation leads to the *GKO-Cauchy* algorithm of Gohberg, Kailath and Olshevsky [33] for fast factorization of Cauchy-type matrices with partial pivoting, and many recent variations on the theme by Boros, Gohberg, Ming Gu, Heinig, Kailath, Olshevsky, M. Stewart, *et al*: see [10, 33, 40, 46, 60, 68].

4.1. The GKO-Toeplitz algorithm. Heinig [46] showed that, if T is a Toeplitz-type matrix, then

$$R = FTD^{-1}F^*$$

is a Cauchy-type matrix, where

$$F = \frac{1}{\sqrt{n}}[e^{2\pi i(k-1)(j-1)/n}]_{1\le k,j\le n}$$

is the Discrete Fourier Transform matrix,

$$D = \mathrm{diag}(1, e^{\pi i/n}, \ldots, e^{\pi i(n-1)/n}),$$

and the generators of T and R are simply related.

The transformation $T \leftrightarrow R$ is perfectly stable because F and D are unitary. Note that R is (in general) complex even if T is real. This increases the constant factors in the time bounds, because complex arithmetic is required.

Heinig's observation was exploited by Gohberg, Kailath and Olshevsky [33]: R can be factorized as $R = P^T LU$ using GKO-Cauchy. Thus, from the factorization

$$T = F^* P^T LUFD \ ,$$

a linear system involving T can be solved in $O(n^2)$ operations. The full procedure of conversion to Cauchy form, factorization, and solution requires $O(n^2)$ (complex) operations.

Other structured matrices, such as Hankel, Toeplitz-plus-Hankel, Vandermonde, Chebyshev-Vandermonde, etc, can be converted to Cauchy-type matrices in a similar way.

4.2. Error Analysis. Because GKO-Cauchy and GKO-Toeplitz involve partial pivoting, we might guess that their stability would be similar to that of Gaussian elimination with partial pivoting. Unfortunately, there is a flaw in this reasoning. During GKO-Cauchy the *generators* have to be transformed, and the partial pivoting does not ensure that the transformed generators are small.

Sweet and Brent [73] show that significant generator growth can occur if all the elements of $\Phi\Psi$ are small compared to those of $|\Phi||\Psi|$. This can not happen for ordinary Cauchy matrices because $\Phi^{(k)}$ and $\Psi^{(k)}$ have only one column and one row respectively. However, it can happen for higher displacement-rank Cauchy-type matrices, even if the original matrix is well-conditioned.

4.3. The Toeplitz Case. In the Toeplitz case there is an extra constraint on the selection of Φ and Ψ, but it is still possible to give examples where the normalized solution error grows like κ^2 and the normalized residual grows like κ, where κ is the condition number of the Toeplitz matrix. Thus, the GKO-Toeplitz algorithm is (at best) weakly stable.

It is easy to think of modified algorithms which avoid the examples given by Sweet and Brent, but it is difficult to prove that they are stable in all cases. Stability depends on the worst case, which may be rare and hard to find by random sampling.

The problem with the original GKO algorithm is growth in the generators. Ming Gu suggested exploiting the fact that the generators are not unique. Recall the Sylvester equation (3.1). Clearly we can replace Φ by ΦM and Ψ by $M^{-1}\Psi$, where M is any invertible $\alpha \times \alpha$ matrix, because this does not change the product $\Phi\Psi$. Similarly at later stages of the GKO algorithm.

Ming Gu [40] proposes taking M to orthogonalize the columns of Φ (that is, at each stage perform an orthogonal factorization of the generators). M. Stewart [68] proposes a (cheaper) LU factorization of the generators. In both cases, clever pivoting schemes give error bounds analogous to those for Gaussian elimination with partial pivoting.

4.4. Gu and Stewart's error bounds. The error bounds obtained by Ming Gu and M. Stewart involve a factor K^n, where K depends on the ratio of the largest to smallest modulus elements in the Cauchy matrix

$$\left[\frac{1}{t_i - s_j}\right]_{ij} \ .$$

Although this is unsatisfactory, it is similar to the factor 2^{n-1} in the error bound for Gaussian elimination with partial pivoting. In practice, the latter factor is extremely pessimistic, which explains why Gaussian elimination with partial pivoting is popular [49]. Perhaps the bounds of Ming Gu and M. Stewart are similarly pessimistic, although practical experience is not yet extensive enough to be confident of this. M. Stewart [68] gives some interesting numerical results which indicate that his scheme works well, but more numerical experience is necessary before a definite conclusion can be reached.

4.5. A general strategy for guaranteed results. It often happens that there is a choice of

1. A fast algorithm which usually gives an accurate result, but occasionally fails (or at least can not be proved to succeed every time), or
2. An algorithm which is guaranteed to be stable but is slow.

In such cases, a good strategy is to use the fast algorithm, but then check the normalized residual $||A\tilde{x} - b||/(||A|| \cdot ||\tilde{x}||$, where $\tilde{x}$ is the computed solution of the system $Ax = b$. If the residual is sufficiently small we can accept $\tilde{x}$ as a reasonably accurate solution. In the rare cases that the residual is not sufficiently small we can can use the slow but stable algorithm (alternatively, if the residual is not too large, one or two iterations of iterative refinement may be sufficient and faster [51, 81]).

An example of this general strategy is the solution of a Toeplitz system by Ming Gu or M. Stewart's modification of the GKO algorithm. We can use the $O(n^2)$ algorithm, check the residual, and resort to iterative refinement or a stable $O(n^3)$ algorithm in the (rare) cases that it is necessary. Computing the residual takes only $O(n \log n)$ arithmetic operations.

5. Positive Definite Structured Matrices. An important class of algorithms, typified by the algorithm of Bareiss [2], find an LU factorization of a Toeplitz matrix T, and (in the symmetric case) are related to the classical algorithm of Schur [17, 32, 64].

It is interesting to consider the numerical properties of these algorithms and compare with the numerical properties of the Levinson algorithm (which essentially finds an LU factorization of T^{-1}).

5.1. The Bareiss algorithm for positive definite matrices. Bojanczyk, Brent, de Hoog and Sweet (abbreviated BBHS) have shown in [8, 70] that the numerical properties of the Bareiss algorithm are similar to those of Gaussian elimination (*without* pivoting). Thus, the algorithm is stable for positive definite symmetric Toeplitz matrices.

The Levinson algorithm can be shown to be weakly stable for bounded n, and numerical results by Varah [77], BBHS and others suggest that this is all that we can expect. Thus, the Bareiss algorithm is (generally) better numerically than the Levinson algorithm.

Cybenko [24] showed that if certain quantities called "reflection coefficients" are positive then the Levinson-Durbin algorithm for solving the Yule-Walker equations (a positive-definite system with special right-hand side) is stable. Unfortunately, Cybenko's result is not usually applicable, because most positive-definite Toeplitz matrices do not usually satisfy the restrictive condition on the reflection coefficients. Some relevant numerical examples are given in [8].

5.2. The generalized Schur algorithm. The Schur algorithm can be generalized to factor a large variety of structured matrices [52, 54]. For example, suitably generalized Schur algorithms apply to block Toeplitz matrices, Toeplitz block matrices, and to matrices of the form $T^T T$, where T is rectangular Toeplitz.

It is natural to ask if the stability results of BBHS (which are for the classical Schur/Bareiss algorithm) extend to the generalized Schur algorithm. This was considered by Chandrasekharan and Sayed [20] and M. Stewart and Van Dooren [69]. The displacement structure considered in [20] is more general than that considered in [69], which in turn is more general than that considered in [8, 70]. Each increase in generality complicates the analysis and seems to make the stability result more dependent on details of the method of implementing the hyperbolic rotations which occur in the algorithms. The interested reader is referred to [20, 69] for details.

The overall conclusion is that the generalized Schur algorithm is stable for positive definite symmetric (or Hermitian) matrices, *provided* the hyperbolic transformations in the algorithm are implemented correctly.

6. Fast Orthogonal Factorization. In an attempt to achieve stability without pivoting, and to solve $m \times n$ least squares problems ($m \geq n$), it is natural to consider algorithms for computing an orthogonal factorization

$$T = QU \tag{6.1}$$

of an $m \times n$ Toeplitz matrix T. We assume that T has full rank n. For simplicity, in the time bounds we assume $m = O(n)$ to avoid functions of both m and n.

The first $O(n^2)$ (more precisely, $O(mn)$) algorithm for computing the factorization (6.1) was introduced by Sweet [70]. Unfortunately, Sweet's algorithm is unstable: see Luk and Qiao [58].

Other $O(n^2)$ algorithms for computing the matrices Q and U or U^{-1} were given by Bojanczyk, Brent and de Hoog (abbreviated BBH) [6], Chun *et al* [23], Cybenko [25], and Qiao [62], but none of them has been shown to be stable, and in several cases examples show that they are unstable.

It may be surprising that fast algorithms for computing an orthogonal factorization (6.1) are unstable. The classical $O(n^3)$ algorithms are stable because they form Q as a product of elementary orthogonal matrices (usually Givens or Householder matrices [36, 38, 81]). Unlike the classical 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. This observation explains both their speed and their instability!

For example, the algorithms of BBH [6] and Chun *et al* [23] depend on Cholesky downdating, and numerical experiments show that they do not give a Q which is close to orthogonal. This is not too surprising, because Cholesky downdating is known to be a sensitive numerical problem [5, 67].

6.1. Use of the semi-normal equations. It can be shown that, provided the Cholesky downdates are implemented in a certain way (analogous to the condition for the stability of the generalized Schur algorithm), the BBH algorithm computes U in a weakly stable manner [7]. In fact, the computed upper triangular matrix $\tilde{U}$ is about as good as can be obtained by performing a Cholesky factorization of $T^T T$, so

$$\|T^T T - \tilde{U}^T \tilde{U}\| / \|T^T T\| = O_m(\varepsilon) \ .$$

Thus, by solving

$$\tilde{U}^T \tilde{U} x = T^T b$$

(the so-called *semi-normal* equations) we have a *weakly stable* algorithm for the solution of general Toeplitz systems $Tx = b$ in $O(n^2)$ operations. The solution can be improved by iterative refinement if desired [37]. The computation of Q is avoided, and the algorithm is applicable to full-rank Toeplitz least squares problems. The disadvantage of this method is that, by implicitly forming T^TT, the condition of the problem is effectively squared. If the condition number $\kappa = \kappa(T)$ is in the range

$$\frac{1}{\sqrt{\varepsilon}} \le \kappa \le \frac{1}{\varepsilon}$$

then it will usually be impossible to get any significant figures in the result (iterative refinement may fail to converge) without reverting to a slow but stable orthogonal factorization algorithm. One remedy is to use double-precision arithmetic, i.e. replace ε by ε^2, but this may be difficult if ε already corresponds to the maximum precision implemented by hardware.

Another way of computing the upper triangular matrix U, but not the orthogonal matrix Q, in (6.1), is to apply the generalized Schur algorithm to T^TT. This method also squares the condition number.

6.2. Computing Q stably. It seems difficult to give a satisfactory $O(n^2)$ algorithm for the computation of Q in the factorization (6.1). Using a modification of the embedding approach pioneered by Chun and Kailath [22, 52], Chandrasekharan and Sayed [21] give a stable algorithm to compute a factorization

$$T = LQU \tag{6.2}$$

where L is lower triangular. (Of course, the factorization (6.2) is not unique.) The algorithm of [21] can be used to solve linear equations, but not least squares problems (because T has to be square, and in any case the matrix Q in (6.2) is different from the matrix Q in (6.1)). Because the algorithm involves embedding the $n \times n$ matrix T in a $2n \times 2n$ matrix

$$\begin{bmatrix} T^TT & T^T \\ T & 0 \end{bmatrix},$$

the constant factors in the operation count are large: $59n^2 + O(n \log n)$, which should be compared to $8n^2 + O(n \log n)$ for BBH and the semi-normal equations. (These operation counts apply for $m = n$: see [7] for operation counts of various algorithms when $m \ge n$.) Thus, although the embedding approach is elegant and leads to interesting (and in some cases stable) $O(n^2)$ algorithms, a penalty is the significant increase in the constant factors. This is analogous to the penalty paid by the GKO algorithm because of the introduction of complex arithmetic.

7. Conclusions. Although this survey has barely scratched the surface, we hope that the reader who has come this far is convinced that questions of numerical stability are amongst the most interesting, difficult, and useful questions that we can ask about fast algorithms for structured linear systems. It is not too hard to invent a new fast algorithm, but to find a new stable algorithm is more difficult, and to *prove* its stability or weak stability is a real challenge!

Acknowledgements. A preliminary version of this review appeared in [12]. Thanks to Greg Ammar, Adam Bojanczyk, James Bunch, Shiv Chandrasekharan, George Cybenko, Paul Van Dooren, Lars Eldén, Roland Freund, Andreas Griewank, Ming Gu, Martin Gutknecht, Georg Heinig, Frank de Hoog, Franklin Luk, Vadim Olshevsky, Haesun Park, Ali Sayed, Michael Stewart, Douglas Sweet and James Varah for their assistance.

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*, Numer. Math. 13 (1969), 404–424.

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

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

[5] A. W. Bojanczyk, R. P. Brent, P. Van Dooren and F. R. de Hoog, *A note on downdating the Cholesky factorization*, SIAM J. Sci. Stat. Computing 8 (1987), 210–220.

[6] A. W. Bojanczyk, R. P. Brent and F. R. de Hoog, *QR factorization of Toeplitz matrices*, Numer. Math. 49 (1986), 81–94.

[7] ———, *Stability analysis of a general Toeplitz systems solver*, Numerical Algorithms 10 (1995), 225–244.

[8] 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*, SIAM J. Matrix Anal. Appl. 16 (1995), 40–57.

[9] 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.

[10] T. Boros, T. Kailath and V. Olshevsky, *Fast algorithms for solving Cauchy linear systems*, preprint, 1995.

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

[12] ———, *Numerical stability of some fast algorithms for structured matrices*, Proc. Workshop on Scientific Computing 1997 (Hong Kong, March 1997), Springer-Verlag, 1998, 41–48.

[13] 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.

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

[15] ———, *The weak and strong stability of algorithms in numerical linear algebra*, Linear Alg. Appl. 88/89 (1987), 49–66.

[16] ———, *Matrix properties of the Levinson and Schur algorithms*, J. Numerical Linear Algebra with Applications 1 (1992), 183–198.

[17] J. P. Burg, *Maximum Entropy Spectral Analysis*, PhD thesis, Stanford University, Stanford, California, 1975.

[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] ———, *A look-ahead Levinson algorithm for general Toeplitz systems*, IEEE Trans. Signal Process. 40 (1992), 1079–1090.

[20] S. Chandrasekaran and A. H. Sayed, *Stabilizing the generalized Schur algorithm*, SIAM J. Matrix Anal. Appl. 17 (1996), 950–983.

[21] ———, *A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations*, preprint, Jan. 1996. [Appeared in *SIAM J. Matrix Anal. Appl.* 19 (1998), 107–139.]

[22] J. Chun, *Fast array algorithms for structured matrices*, PhD thesis, Stanford University, Stanford, California, 1989.

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

[24] G. Cybenko, *The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations*, SIAM J. Sci. Stat. Computing 1 (1980), 303–319.
[25] ———, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Sci. Stat. Computing 8 (1987), 734–740.
[26] J.-M. Delosme and I. C. F. Ipsen, *Parallel solution of symmetric positive definite systems with hyperbolic rotations*, Linear Algebra Appl. 77 (1986), 75–111.
[27] P. Delsarte, Y. V. Genin and Y. G. Kamp, *A generalisation of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile*, IEEE Trans. Acoustics, Speech and Signal Processing ASSP–33 (1985), 964–971.
[28] J. Durbin, *The fitting of time-series models*, Rev. Int. Stat. Inst. 28 (1959), 229–249.
[29] G. E. Forsythe and C. B. Moler, *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, New Jersey, 1967.
[30] 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.
[31] ———, *A look-ahead algorithm for the solution of general Hankel systems*, Numer. Math. 64 (1993), 295–321.
[32] I. Gohberg (editor), *I. Schur Methods in Operator Theory and Signal Processing* (Operator Theory: Advances and Applications, Volume 18), Birkhäuser Verlag, Basel, 1986.
[33] I. Gohberg, T. Kailath and V. Olshevsky, *Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp. 64 (1995), 1557–1576.
[34] I. Gohberg and I. Koltracht, *Mixed, componentwise and structured condition numbers*, SIAM J. Matrix Anal. Appl. 14 (1993), 688–704.
[35] I. Gohberg, I. Koltracht and D. Xiao, *Condition and accuracy of algorithms for computing Schur coefficients of Toeplitz matrices*, SIAM J. Matrix Anal. Appl. 15 (1994), 1290–1309.
[36] G. H. Golub, *Numerical methods for solving linear least squares problems*, Numer. Math. 7 (1965), 206–216.
[37] G. H. Golub and J. H. Wilkinson, *Note on iterative refinement of least squares solution*, Numer. Math. 9 (1966), 139–148.
[38] G. H. Golub and C. Van Loan, *Matrix Computations*, third edition, Johns Hopkins Press, Baltimore, Maryland, 1997.
[39] N. Gould, *On growth in Gaussian elimination with complete pivoting*, SIAM J. Matrix Anal. Appl. 12 (1991), 354–361.
[40] Ming Gu, *Stable and efficient algorithms for structured systems of linear equations*, Tech. Report LBL-37690, Lawrence Berkeley Laboratory, Aug. 1995. [Appeared in *SIAM J. Matrix Anal. Appl.* 19 (1998), 279–306.]
[41] ———, *New fast algorithms for structured least squares problems*, Tech. Report LBL-37878, Lawrence Berkeley Laboratory, Nov. 1995.
[42] 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.
[43] 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.
[44] M. H. Gutknecht and M. Hochbruck, *The Stability of Inversion Formulas for Toeplitz Matrices*, IPS Research Report 93–13, ETH, Zürich, October 1993.
[45] P. C. Hansen and H. Gesmar, *Fast orthogonal decomposition of rank deficient Toeplitz matrices*, Numerical Algorithms 4 (1993), 151–166.
[46] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, Linear Algebra for Signal Processing, IMA Volumes in Mathematics and its Applications, Vol. 69, 1994, 95–114.
[47] G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Operator Theory, vol. 13, Birkhäuser, Basel, 1984.
[48] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, SIAM J. Matrix Anal. Appl. 13 (1992), 162-175.
[49] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[50] D. Hilbert, *Ein Beitrag zur Theorie des Legendre'schen Polynoms*, Acta Math. 18 (1894), 155–160.
[51] M. Jankowski and M. Wozniakowski, *Iterative refinement implies numerical stability*, BIT 17 (1977), 303–311.
[52] 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.
[53] T. Kailath, S. Y. Kung and M. Morf, *Displacement ranks of matrices and linear equations*, *J. Math. Anal. Appl.* 68 (1979), 395–407.

[54] T. Kailath and A. H. Sayed, *Displacement structure: theory and applications*, SIAM Review 37 (1995), 297–386.

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

[56] 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.

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

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

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

[60] V. Olshevsky, *Pivoting on structured matrices with applications*, preprint, July 1997.

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

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

[63] A. H. Sayed, *Displacement Structure in Signal Processing and Mathematics*, PhD thesis, Stanford University, Stanford, California, 1992.

[64] 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 [32], 31–59.

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

[66] ———, *Perturbation bounds for the QR factorization of a matrix*, SIAM J. Numer. Anal. 14 (1977), 509–518.

[67] ———, *The effect of rounding error on an algorithm for downdating a Cholesky factorization*, J. Inst. Math. Appl. 23 (1979), 203–213.

[68] M. Stewart, *Stable pivoting for the fast factorization of Cauchy-like matrices*, preprint, Jan. 13, 1997.

[69] M. Stewart and P. Van Dooren, *Stability issues in the factorization of structured matrices*, SIAM J. Matrix Anal. Appl. 18 (1997), 104–118.

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

[71] ———, *Fast Toeplitz orthogonalization*, Numer. Math. 43 (1984), 1–21.

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

[73] D. R. Sweet and R. P. Brent, *Error analysis of a fast partial pivoting method for structured matrices*, Proceedings SPIE, Volume 2563, Advanced Signal Processing Algorithms SPIE, Bellingham, Washington, 1995, 266–280.

[74] G. Szegö, *Orthogonal Polynomials*, AMS Colloquium publ. XXIII, AMS, Providence, Rhode Island, 1939.

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

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

[77] J. M. Varah, *The prolate matrix*, Linear Algebra Appl. 187 (1993), 269–278.

[78] 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).

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

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

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