Comparison of the Numerical Stability of Some Fast Algorithms for Structured Matrices*

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

1 August 1996

*Presented at the *International Workshop on Numerical Methods for Structured Matrices in Filtering and Control*, Santa Barbara, 1–3 August 1996.
Copyright © 1996, R. P. Brent.
rpb157t typeset using LaTeX

Abstract

We consider the numerical stability of $O(n^2)$ algorithms for solving systems of n linear equations with a low displacement-rank structure. For example, the matrices involved may be Toeplitz or Hankel. We compare the results which have been obtained for algorithms of the Levinson and Schur type, fast QR algorithms combined with the seminormal equations, and recent algorithms which incorporate partial pivoting without destroying the structure.

Outline

- Stability and weak stability (briefly)
- Positive definite Toeplitz matrices
  - The Levinson algorithm
  - The Bareiss algorithm
- General Toeplitz and other structured matrices
  - QR algorithms (e.g. BBH)
  - Partial pivoting algorithms (e.g. GKO)

Acronyms

BBH	=	Bojanczyk, Brent & de Hoog.
BBHS	=	Bojanczyk, Brent, de Hoog & Sweet.
GKO	=	Gohberg, Kailath & Olshevsky.

Notation

Let

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

be a real, nonsingular, $n \times n$ Toeplitz matrix, so

$$a_{i,j} = a_{j-i}$$

for $1 \le i, j \le n$. (The results sometimes extend to full-rank $m \times n$ matrices, $m \ge n$, but for simplicity we assume $m = n$ today. We consider more general structured matrices later.)

$A^T A$ has a Cholesky factorization

$$A^T A = R^T R,$$

where R is a (unique) upper triangular $n \times n$ matrix with positive diagonal elements. Also,

$$A = QR,$$

where Q is orthogonal, i.e.

$$Q^T Q = I.$$

Condition Number of A

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

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

We say that A is *well-conditioned* if $\kappa(A)$ is "small" in some sense.

We do not considered "structured" condition numbers here (though the distinction may be important, e.g. for Cauchy matrices).

For convenience in stating the error bounds, we may assume that σ_1 is of order unity.

Let B be a principal $k \times k$ submatrix of A.
If A is symmetric positive definite then

$$\kappa_2(B) \leq \kappa_2(A).$$

However, in general this is not true –
B could be badly conditioned or even singular when A is well-conditioned.

Error Bounds and "O" Notation

Let ε be the machine precision. It is convenient to subsume a polynomial in n into the "O" notation. Thus, an error bound of the form $\|E\| = O_n(\varepsilon)$ or simply $\|E\| = O(\varepsilon)$ will mean that

$$\|E\| \leq P(n)\varepsilon$$

for some polynomial $P(n)$ and all sufficiently small ε. If the error bound depends on κ then this will be mentioned explicitly, e.g.

$$\|E\| = O(\kappa\varepsilon).$$

The meaning of "sufficiently small" may depend on κ (for example, we may need $\kappa^2\varepsilon < 1$).
We shall ignore $O(\varepsilon^2)$ terms in the error analyses.

Numerical Quantities

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. $\widehat{a}_{i,j}$. These are not computed, but the error analysis shows that they exist and gives bounds on their difference from the corresponding exact values.

Stability

Consider algorithms for solving a nonsingular, $n \times n$ linear system $Ax = b$.
There are many definitions of numerical stability in the literature. Our definitions follow those of Bunch(1987). Definition 1 says that 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.

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.*

Note that the matrix $\widehat{A}$ does not have to be in the class $\mathcal{A}$. For example, $\mathcal{A}$ might be the class of nonsingular Toeplitz matrices, but $\widehat{A}$ need not be a Toeplitz matrix. (If we do require $\widehat{A} \in \mathcal{A}$ we get what Bunch calls *strong stability*.)

Closeness

In Definition 1, "close" means close in a relative sense, using some norm, i.e.

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

Recall our convention that polynomials in n may be omitted from $O(\varepsilon)$ terms.
We are ruling out faster than polynomial growth in n, such as $O(2^n\varepsilon)$ or $O(n^{\frac{\log n}{4}}\varepsilon)$. Perhaps this too strict (consider Gaussian elimination).

The Residual

The condition of Definition 1 is equivalent to saying that the scaled residual
$\|A\tilde{x} - b\|/(\|A\| \cdot \|\tilde{x}\|)$ is small.

How Good is the Solution ?

Provided $\kappa\varepsilon$ is sufficiently small, stability implies that

$$\|\tilde{x} - x\|/\|x\| = O(\kappa\varepsilon).$$

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, "small" means $O(\varepsilon)$, and "well-conditioned" means that $\kappa(A)$ is bounded by a polynomial in n. It is easy to see that stability implies weak stability.
Define the *residual*

$$r = A\tilde{x} - b.$$

It is well-known that

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

Thus, for well-conditioned A,
$\|\tilde{x} - x\|/\|x\|$ is small *if and only if* $\|r\|/\|b\|$ is small. (This gives an equivalent definition of weak stability.)

The Levinson-Durbin Algorithm

In 1947, Levinson gave an algorithm for solving a symmetric $n \times n$ Toeplitz system in $O(n^2)$ operations and $O(n)$ storage. In linear prediction we want to solve a Toeplitz system with a special right-hand side, called the "Yule-Walker equations": $Ax = b$, where

$$b = -(a_1, a_2, \ldots, a_n)^T.$$

Durbin (1960) streamlined Levinson's algorithm for this special case.

The algorithm was actually discovered by Wiener (1941) and (independently) by Kolmogorov (1941). It is related to Szegö recursions for polynomials orthogonal on the unit circle. Thus, it could be called the Szegö-Kolmogorov-Wiener-Levinson-Durbin algorithm !

In the linear prediction problem, A is symmetric positive definite, so we assume this when discussing the Levinson-Durbin algorithm. It is also convenient to assume that $a_0 = 1$.

The Levinson-Durbin Recursion

The algorithm is defined by the recursion (for $j = 1, 2, \ldots, n$):

$$\begin{aligned} K_j &= -(a_j + x_{j-1,1}a_{j-1} + \cdots \\ &\quad + x_{j-1,j-1}a_1)/E_{j-1}, \\ x_j &= \begin{pmatrix} x_{j-1} + K_j x_{j-1}^R \\ K_j \end{pmatrix}, \\ E_j &= (1 - K_j^2)E_{j-1}, \end{aligned}$$

where $E_0 = 1$,

$$x_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,j})^T,$$

and

$$x_j^R = (x_{j,j}, x_{j,j-1}, \ldots, x_{j,1})^T$$

is the reflection of x_j. One can verify by induction that x_j solves the j-th order Yule-Walker equations and

$$x = x_n = (x_{n,1}, x_{n,2}, \ldots, x_{n,n})^T$$

solves $Ax = b$.

Interpretation of K_j and E_j

The K_j are called "reflection coefficients" or "partial correlation coefficients" and satisfy

$$-1 < K_j < 1.$$

Note that the formula for K_j involves an inner product.

The E_j are the mean square prediction errors.

Matrix Factorization Interpretation

If

$$C = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ x_{n-1,1} & 1 & 0 & \cdots & 0 \\ x_{n-1,2} & x_{n-2,1} & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ x_{n-1,n-1} & x_{n-2,n-2} & & \cdots & 1 \end{pmatrix}$$

and

$$D = \mathrm{diag}(E_{n-1}, E_{n-2}, \ldots, E_0)$$

then

$$A^{-1} = CD^{-1}C^T,$$

so the Levinson-Durbin algorithm computes a Cholesky factorization of A^{-1}. The factorization can be used to find upper and lower bounds on $\|A^{-1}\|$ (which is much the same thing as $\kappa(A)$).

Note that A is sure to be poorly conditioned if E_{n-1} is small, i.e. if the mean square prediction error is small. (But this is what we want – Catch 22 !)

Theorem 1 (Cybenko, 1980)

$$\max\left(\frac{1}{E_{n-1}}, \frac{1}{\prod_{j=1}^{n-1}(1-K_j)}\right) \leq \|A^{-1}\|_1 \leq \prod_{j=1}^{n-1}\frac{1+|K_j|}{1-|K_j|},$$

where

$$E_{n-1} = \prod_{j=1}^{n-1}(1-K_j^2),$$

and the K_j are the reflection coefficients.

Lemma 1 (Cybenko)

$$\|x\|_1 \geq \left|\prod_{j=1}^{n}(1+K_j) - 1\right|$$

Theorem 2 (Cybenko)
If floating-point arithmetic is used with machine precision ε, then the residual $r = A\tilde{x} - b$ satisfies

$$\|r\| = O\left(\varepsilon\left(\prod_{j=1}^{n}(1+|K_j|) - 1\right)\right).$$

Corollary 1
If all $K_j \geq 0$, then $\|r\|/\|x\| = O(\varepsilon)$.

Corollary 2
If all the reflection coefficients are non-negative, then the Levinson-Durbin algorithm is stable.

Comparison with Cholesky

Cybenko notes that when $K_j \geq 0$ the bounds on the residual for the Levinson-Durbin method and Cholesky's method are of comparable size. However, this does not prove that the Levinson-Durbin method is stable (in the sense of Definition 1) for *all* positive-definite matrices A.

What if some $K_j < 0$?

Cybenko's analysis is not sharp if some of the reflection coefficients are negative, because of the absolute values $|K_j|$ in his inequalities[1]. However, from Cybenko's results we can deduce a bound which is similar to that for Gaussian elimination with partial pivoting. The proof is easy, but I have not seen it in print. From Theorem 1,

$$\frac{1}{\kappa_1(A)} \le \frac{1}{\|A^{-1}\|_1} \le E_{n-1} = \prod_1^{n-1}(1-K_j^2),$$

so, from Theorem 2,

$$\frac{\|r\|}{\kappa\varepsilon} = O(M^{n-1}),$$

where

$$M = \max_{-1<K<1}(1+|K|)(1-K^2) = \frac{32}{27}$$

(the maximum occurs at $|K| = \frac{1}{3}$). Thus

[1] Koltracht and Lancaster (1986) have improved Cybenko's upper bound on $\kappa(A)$ in this case.

Theorem 3
For the Levinson-Durbin algorithm with positive definite symmetric A, but no restriction on the signs of the reflection coefficients,

$$\|r\| = O\left(\kappa\varepsilon\left(\frac{32}{27}\right)^n\right).$$

This is not too bad when compared with the bound $O(2^n\varepsilon)$ for Gaussian elimination with partial pivoting. In both cases, $\|r\|$ is usually much smaller than the corresponding bound.

Corollary 3
The Levinson-Durbin algorithm for solving the positive definite symmetric Yule-Walker equations of *bounded size* n is weakly stable.

Numerical experiments (BBHS, Varah) suggest that weak stability is all that we can expect to prove.

Szegö/Levinson vs Schur/Bareiss

The algorithms of Wiener, Levinson, Durbin, Trench and Zohar find an LU factorization of A^{-1} and (in the symmetric case) are related to the classical Szegö recursions for polynomials orthogonal on the unit circle. These algorithms typically involve inner products.

Another class of algorithms, typified by the algorithm of Bareiss (1969), find an LU factorization of A, and (in the symmetric case) are related to the classical algorithm of Schur for the continued fraction representation of a holomorphic function in the unit disk. These algorithms typically involve outer products.

It is interesting to compare the numerical properties of the two classes of algorithms.

Bareiss – Positive Definite Case

Sweet (1982–1993) and BBHS (1993) have shown that the numerical properties of the Bareiss algorithm when implemented in floating-point arithmetic are similar to those of Gaussian elimination (*without* pivoting).

Thus, the algorithm is stable for positive definite symmetric A. This is a stronger result than has been proved for the Levinson algorithm – we only showed that it was weakly stable for bounded n, and numerical results by Varah, BBHS suggest that this is all that we can expect.

Bareiss – General Full Rank Case

For general Toeplitz A the Bareiss algorithm is unstable, just like Gaussian elimination without pivoting. In fact, both break down immediately if $a_{1,1} = 0$, and exhibit instability if $a_{1,1}$ is small.

Sweet (1993) has shown that it is possible to introduce pivoting into the Bareiss algorithm to avoid instability[2]. However, in the worst case the overhead of pivoting is $O(n^3)$ so we no longer have a "fast" $O(n^2)$ algorithm.

There are analogous "lookahead" modifications of Levinson's algorithm (Chan & Hansen, Freund & Zha, Gutknecht, ...) but similar comments apply.

[2] Using a connection between the Bareiss multipliers and the Trench-Zohar algorithm, Sweet also shows how to introduce pivoting into the Trench-Zohar algorithm.

Fast Orthogonal Factorization

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

$$A = QR$$

of A. The first such $O(n^2)$ algorithm was introduced by Sweet (1982–84). Unfortunately, Sweet's algorithm is unstable[3].

Other $O(n^2)$ algorithms for computing the matrices Q and R or R^{-1} were given by BBH (1986), Chun *et al* (1987), Cybenko (1987), and Qiao (1988), but none of them has been shown to be stable. In several cases examples show that they are not stable.

[3] It depends on the condition of a submatrix of A – see Luk and Qiao (1987).

The Problem – Q

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.

For example, the algorithms of Bojanczyk, Brent and de Hoog (1986) and Chun *et al* (1987) depend on Cholesky downdating, and numerical experiments show that they do not give a Q which is close to orthogonal.

The Saving Grace – R and Semi-Normal Equations

It can be shown (BBH, 1995) that, provided the Cholesky downdates are implemented in a certain way, the BBH algorithm computes R in a weakly stable manner. In fact, the computed upper triangular matrix $\tilde{R}$ is about as good as can be obtained by performing a Cholesky factorization of $A^T A$, so

$$\|A^T A - \tilde{R}^T \tilde{R}\| / \|A^T A\| = O(\varepsilon) \ .$$

Thus, by solving

$$\tilde{R}^T \tilde{R} x = A^T b$$

(the so-called *semi-normal* equations) we have a *weakly stable* algorithm for the solution of general Toeplitz systems $Ax = b$ in $O(n^2)$ operations. The solution can be improved by iterative refinement if desired.

Note that the computation of Q is avoided. I do not know a satisfactory $O(n^2)$ algorithm for the computation of Q.

Error Bounds

We obtain a result $\tilde{x}$ for which

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

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

$$\|r\|/\|x\| = O(\kappa\varepsilon) \ .$$

The method is *weakly stable* (according to Definition 2), although probably not *stable*. For a *stable* method the error bounds would be reduced by a factor of κ.

Use of Partial Pivoting

The idea of computing a QR factorization in $O(n^2)$ operations is only partially successful because, as we saw above, the computed matrix Q is useless. This motivates returning to the LU factorization, but trying to incorporate partial pivoting.

At first sight, pivoting seems to destroy the structure. However, there is a clever way around this.

First, a small digression to more general structured matrices $\cdots$

Displacement Structure

Structured matrices satisfy a *Sylvester equation* which has the form

$$\nabla_{\{A_f, A_b\}}(R) = A_f R - R A_b = \Phi \Psi \ , \qquad (1)$$

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 small integer (usually 4 or less).

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.

Example – Cauchy

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

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

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

and

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

More general matrices, where Φ and Ψ are any rank-α matrices, are called *Cauchy-type*.

Example – Toeplitz

For a Toeplitz matrix $T = [t_{ij}] = [a_{i-j}]$

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

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

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

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 in the obvious way.

Gaussian Elimination and Schur Complements

Let the input matrix, R_1, have the partitioning

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

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

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

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 $R_1 = LU$.

Structured Gaussian Elimination

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

Partial Pivoting

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 for fast factorization of Cauchy-type matrices with partial pivoting (and many recent variations on the theme by Boros, Gohberg, Heinig, Kailath, Olshevsky, ...).

Toeplitz to Cauchy

Heinig (1994) and (later) GKO (1995) show 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 (we omit the details).

GKO-Toeplitz

We can convert the generators of T to the generators of R in $O(2\alpha n \log n)$ operations via FFTs. R can then 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 $n^2 + 2n \log n$ operations. The procedure of conversion of Cauchy form, factorization, and solution requires $O(n^2)$ (complex) operations.

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

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.

The Catch

Unfortunately, there is a flaw in the above reasoning. During GKO-Cauchy the *generators* have to be transformed, and the partial pivoting does not ensure that the transformed generators are small.

Sweet & Brent (1995) 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.

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 the normalized solution error grows like κ^2 and the normalized residual grows like κ. Thus, the GKO-Toeplitz algorithm is (at best) weakly stable.

It is not hard to think of modified algorithms which avoid the examples given by Sweet & Brent, but it is not clear if they are stable in all cases. (Stability depends on the worst case, which may be rare and hard to find by random sampling.)

In practice, we can use an $O(n^2)$ algorithm; check the residual (or, cheaper, check the growth factors *a posteriori*), and resort to iterative refinement or a stable $O(n^3)$ algorithm in the (rare) cases that it is necessary.

Some Open Problems

Let A be a structured matrix (e.g. Toeplitz or Cauchy type).

- Are Schur-type algorithms (e.g. Bareiss) significantly more accurate than Levinson-type algorithms in the positive definite case ? (Yes ?)
- Is there a fast, stable algorithm for computing $Q = AR^{-1}$?
- Is there a fast algorithm using some form of pivoting which is as stable as Gaussian elimination with partial pivoting ? (The answer may depend on the displacement rank.) Is this good enough in practice ?
- Does consideration of structured perturbations and structured condition numbers give sharper error bounds than "standard" error analysis ?

References

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

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

[3] A. W. Bojanczyk, R. P. Brent, P. Van Dooren and F. R. de Hoog, "A note on downdating the Cholesky factorization", *SISSC* 8 (1987), 210–220.

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

[5] A. W. Bojanczyk, R. P. Brent and F. R. de Hoog, "Stability analysis of a general Toeplitz systems solver", *Numerical Algorithms* 10 (1995), 225–244.

[6] 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.

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

[8] J. R. Bunch, "Stability of methods for solving Toeplitz systems of equations", *SISSC* 6 (1985), 349–364.

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

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

[11] T. F. Chan and P. C. Hansen, "A look-ahead Levinson algorithm for indefinite Toeplitz systems", *SIMAX* 13 (1992), 490–506.

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

[13] G. Cybenko, "The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations", *SISSC* 1 (1980), 303–319.

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

[15] J. Durbin, "The fitting of time-series models", *Rev. Int. Stat. Inst.* 28 (1959), 229–249.

[16] R. W. Freund and H. Zha, *Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems*, Technical Report 91.27, RIACS, NASA Ames Research Center, December 1991 (revised September 1992).

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

[18] I. Gohberg, T. Kailath and V. Olshevsky, "Gaussian elimination with partial pivoting for matrices with displacement structure", *Math. Comp.* 64 (1995), 1557–1576.

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

[20] N. Gould, "On growth in Gaussian elimination with complete pivoting", *SIMAX* 12 (1991), 354–361.

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

[22] 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,* Springer, 1994, 95–114.

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

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

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

[26] I. Koltracht and P. Lancaster, "Condition numbers of Toeplitz and block Toeplitz matrices", in [17], 271–300.

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

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

[29] H. Park and L. Eldén, *Fast and accurate triangularization of Toeplitz matrices*, preprint, 14 April 1993.

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

[31] 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 [17], 31–59.

[32] D. R. Sweet, "Fast Toeplitz orthogonalization", *Numer. Math.* 43 (1984), 1–21.

[33] D. R. Sweet, "The use of pivoting to improve the numerical performance of algorithms for Toeplitz matrices", *SIMAX* 14 (1993), 468–493.

[34] 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.

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

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

[37] 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).

Postscript – additional references

[38] S. Chandrasekaran and A. H. Sayed, *Stabilizing the generalized Schur algorithm*, to appear in *SIMAX* 17 (1996).

[39] S. Chandrasekaran and A. H. Sayed, *A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations*, preprint, Jan. 1996.

[40] Ming Gu, *Stable and efficient algorithms for structured systems of linear equations*, Tech. Report LBL-37690, Lawrence Berkeley Laboratory, Aug. 1995.

[41] Ming Gu, *New fast algorithms for structured least squares problems*, Tech. Report LBL-37878, Lawrence Berkeley Laboratory, Nov. 1995.

[42] M. Stewart and P. Van Dooren, *Stability issues in the factorization of structured matrices*, preprint, Aug. 1996. To appear in *SIMAX*.