Lecture 3

Fast and Numerically Stable Algorithms for Structured Matrices*

*Six lectures on Algorithms, Trinity term 1999.

 lec03

Abstract

We consider the numerical stability/instability of fast algorithms for solving systems of linear equations or linear least squares problems with a low displacement-rank structure. For example, the matrices involved may be Toeplitz or Hankel.

In particular, we consider algorithms which incorporate pivoting without destroying the structure, such as the GKO algorithm, and describe some recent results by Sweet and Brent, Ming Gu, Michael Stewart and others on the stability of these algorithms.

It is interesting to compare these results with the corresponding stability results for algorithms based on the seminormal equations and for the well known algorithms of Schur/Bareiss and Levinson.

Outline

- Structured matrices
  - Displacement structure
  - Cauchy-like matrices
  - Toeplitz-like matrices
  - Toeplitz $\leftrightarrow$ Cauchy
- Partial pivoting algorithms
  - Possible growth of generators
  - Improvements of Gu and Stewart
- Positive definite structured matrices
  - Schur/Bareiss algorithms
  - Comparison with Levinson
  - Generalised Schur algorithm
- Orthogonal factorisation
  - Weak stability
  - The problem of computing Q

Because of shortage of time, I will not consider look-ahead algorithms or iterative algorithms.

Acronyms

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

Notation

R is a structured matrix,
T is a Toeplitz or Toeplitz-type matrix,
P is a permutation matrix,
L is lower triangular,
U is upper triangular,
Q is orthogonal.

Error Bounds

In error bounds $O_n(\varepsilon)$ means $O(\varepsilon f(n))$, where $f(n)$ is a polynomial in n.

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 [11]. 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.)

Displacement Structure

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

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

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.

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).

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.

GE and Schur Complements

Let the input matrix, R_1, have the partitioning

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

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 factorisation

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

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

Structured Gaussian Elimination – The Idea

The key to *structured* Gaussian elimination is 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.

More precisely, we have the following theorem from GKO [21].

Theorem 1 *Let a matrix* $R_1 = \begin{bmatrix} d_1 & \mathbf{w}_1^T \\ \mathbf{y}_1 & \tilde{R}_1 \end{bmatrix}$ *satisfy the Sylvester equation*

$$\nabla_{\{A_{f,1},A_{b,1}\}}(R_1) = A_{f,1}R_1 - R_1A_{b,1} = \Phi^{(1)}\Psi^{(1)} ,$$

where $\Phi^{(1)} = [\varphi_1^{(1)T} \quad \varphi_2^{(1)T} \quad \cdots \quad \varphi_n^{(1)T}]^T$, $\Psi^{(1)} = [\psi_1^{(1)} \quad \psi_2^{(1)} \quad \cdots \quad \psi_n^{(1)}]$, $\varphi_i^{(1)} \in \mathbf{C}^{1\times\alpha}$ *and* $\psi_i^{(1)} \in \mathbf{C}^{1\times\alpha}$, $(i = 1, 2, \ldots, n)$. *Then* R_2, *the Schur complement of* d_1 *in* R_1, *satisfies the Sylvester equation*

$$\nabla_{\{A_{f,2},A_{b,2}\}}(R_2) = A_{f,2}R_2 - R_2A_{b,2} = \Phi^{(2)}\Psi^{(2)} ,$$

where $A_{f,2}$ *and* $A_{b,2}$ *are respectively* $A_{f,1}$ *and* $A_{b,1}$ *with their first rows and first columns deleted, and where* $\Phi^{(2)} = [0,\ \varphi_2^{(2)T},\ \varphi_3^{(2)T},\ \cdots, \varphi_n^{(2)T}]^T$ *and* $\Psi^{(2)} = [0,\ \psi_2^{(2)},\ \psi_3^{(2)},\ \cdots, \psi_n^{(2)}]$ *are given by*

$$\Phi^{(2)}_{2:n,:} = \Phi^{(1)}_{2:n,:} - \mathbf{y}_1\varphi_1^{(1)}/d_1 , \qquad (1)$$

$$\Psi^{(2)}_{:,2:n} = \Psi^{(1)}_{:,2:n} - \psi_1^{(1)}\mathbf{w}_1^T/d_1 . \qquad (2)$$

Structured Gaussian elimination

Algorithm 1 *(Structured Gaussian elimination)*

1. *Recover from the generator* $\Phi^{(1)}$, $\Psi^{(1)}$ *the first row and column of* $R_1 = \begin{bmatrix} d_1 & \mathbf{w}_1^T \\ \mathbf{y}_1 & \tilde{R}_1 \end{bmatrix}$.
2. $[1 \quad \mathbf{y}_1^T/d_1]^T$ *and* $[d_1 \quad \mathbf{w}_1^T]$ *are respectively the first column and row of* L_1 *and* U_1 *in the LU factorisation of* R_1.
3. *Compute by equations (1) and (2), the generator* $(\Phi^{(2)}, \Psi^{(2)})$ *for the Schur complement of* d_1 *in* R_1.
4. *Proceed recursively with* $\Phi^{(2)}$ *and* $\Psi^{(2)}$. *Each major step yields* $[1 \quad \mathbf{y}_k^T/d_k]^T$ *and* $[d_k \quad \mathbf{w}_k^T]$, *which are respectively the first column and row of* L_k *and* U_k *in the LU factorisation of* R_k. *Column* k *of* L *and row* k *of* U *are respectively* $[\mathbf{0}_{k-1}^T \quad 1 \quad \mathbf{y}_k^T/d_k]^T$ *and* $[\mathbf{0}_{k-1}^T \quad d_k \quad \mathbf{w}_k^T]$.

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 factorisation 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.*

Toeplitz to Cauchy

Heinig (1994) 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 related by unitary transformations (see [38, Thm. 2.2] for the details).

The transformation $T \leftrightarrow R$ is perfectly stable because F and D are unitary.

Note that F and R are (in general) complex even if T is real.

GKO-Toeplitz

As pointed out by Heinig (1994) and exploited by GKO (1995), it is possible to convert the generators of T to the generators of R in $O(n \log n)$ operations via FFTs. R can then be factorised as $R = P^T LU$ using GKO-Cauchy. Thus, from the factorisation

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

a linear system involving T can be solved in $O(n^2)$ (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 normalised solution error grows like κ^2 and the normalised 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 & 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.

Gu and Stewart's Improvements

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*

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

where the generators Φ and Ψ are $n \times \alpha$ and $\alpha \times n$ respectively. 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 (1995) proposes taking M to orthogonalize the columns of Φ (that is, at each stage we do an orthogonal factorisation of the generators). Michael Stewart (1997) proposes a (cheaper) LU factorisation of the generators. In both cases, clever pivoting schemes give error bounds analogous to those for Gaussian elimination with partial pivoting.

Gu and Stewart's Error Bounds

The error bounds obtained by Ming Gu and Michael Stewart involve an exponentially growing 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.

Michael Stewart (1997) 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.

In practice, we can use an $O(n^2)$ algorithm such as Michael Stewart's, check the residual, and resort to iterative refinement or a stable $O(n^3)$ algorithm in the (rare) cases that it is necessary.

Positive Definite Structured Matrices

An important class of algorithms, typified by the algorithm of Bareiss (1969), find an LU factorisation of a Toeplitz matrix T, 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.

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

[1] Discovered independently by Kolmogorov and Wiener in 1941.

Bareiss – Positive Definite Case

BBHS (1995) have shown 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 T.

The Levinson algorithm can be shown to be weakly stable for bounded n, and numerical results by Varah, 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 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. However, "random" positive-definite Toeplitz matrices do not usually satisfy Cybenko's condition.

The Generalised Schur Algorithm

The Schur algorithm can be generalised to factor a large variety of structured matrices – see Kailath and Chun (1994) or Kailath and Sayed (1995). For example, the generalised Schur algorithm applies 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 generalised Schur algorithm. This was considered by M. Stewart and Van Dooren (1997), and also (in more generality) by Chandrasekharan and Sayed (1998).

The conclusion is that the generalised Schur algorithm is stable for positive definite matrices, *provided* that the hyperbolic transformations in the algorithm are implemented correctly. In contrast, BBHS showed that stability of the classical Schur/Bareiss algorithm is not so dependent on details of the implementation.

Fast Orthogonal Factorisation

In an attempt to achieve stability without pivoting, and to solve $m \times n$ least squares problems, it is natural to consider algorithms for computing an orthogonal factorisation

$$T = QU$$

of T. The first such $O(n^2)$ algorithm[2] was introduced by Sweet (1982–84). Unfortunately, Sweet's algorithm is unstable.

Other $O(n^2)$ algorithms for computing the matrices Q and U or U^{-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, and in several cases examples show that they are unstable.

[2]For simplicity the time bounds assume $m = O(n)$.

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 generalised Schur algorithm, applied to $T^T T$, computes the upper triangular matrix U but not the orthogonal matrix Q.

The Saving Grace – U and Semi-Normal Equations

It can be shown (BBH, 1995) that, provided the Cholesky downdates are implemented in a certain way (analogous to the condition for the stability of the generalised Schur algorithm) the BBH algorithm computes U in a weakly stable manner. In fact, the computed upper triangular matrix $\tilde{U}$ is about as good as can be obtained by performing a Cholesky factorisation 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.

Note that the computation of Q is avoided.

Computing Q Stably

It is difficult to give a satisfactory $O(n^2)$ algorithm for the computation of Q in the factorisation

$$T = QU$$

Chandrasekharan and Sayed get close – they give a stable algorithm to compute the factorisation

$$T = LQU$$

where L is lower triangular, provided that T is square. Their algorithm can be used to solve linear equations but not for the least squares problem. Also, because their algorithm involves embedding the $n \times n$ matrix T in a $2n \times 2n$ matrix

$$\left[\begin{array}{cc} T^T T & T^T \\ T & 0 \end{array} \right] ,$$

the constant factors in the operation count are large: $59n^2 + O(n \log n)$, compared to $8n^2 + O(n \log n)$ for BBH and seminormal equations.

Some Open Questions

- How do the GKO and similar algorithms using partial pivoting compare with the "lookahead" algorithms of Chan and Hansen [13], Freund and Zha [19], Gutknecht [25], and others ?
- Is there a stable (not just weakly stable) fast algorithm for the (rectangular) structured least squares problem ?
- What can be said about the stability (or instability) of the "superfast" algorithms whose running time is

  $$O\left(n(\log n)^2\right) \text{ ?}$$

  For these algorithms see Ammar and Gragg [1], Brent, Gustavson and Yun [9].
- What are the best generalisations to block-structured problems, e.g. block Toeplitz with $\sqrt{n} \times \sqrt{n}$ blocks ?

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] 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] R. P. Brent, Stability of fast algorithms for structured linear systems, in *Fast Reliable Algorithms for Matrices with Structure* (A. H. Sayed and T. Kailath, eds.), SIAM, Philadelphia, 1999, to appear.

[9] 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.

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

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

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

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

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

[15] S. Chandrasekaran and A. H. Sayed, A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations, *SIAM J. Matrix Anal. Appl.* 19 (1998), 107–139.

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

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

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

[19] 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.

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

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

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

[23] Ming Gu, Stable and efficient algorithms for structured systems of linear equations, *SIMAX* 19 (1998), 279–306.

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

[25] M. H. Gutknecht, Stable row recurrences for the Padé table and generically superfast lookahead solvers for non-Hermitian Toeplitz systems, *Linear Algebra Appl.* 188/189 (1993), 351–422.

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

[27] 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.

[28] 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.

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

[30] 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.

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

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

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

[34] 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 [20], 31–59.

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

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

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

[38] 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.

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

[40] 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).