Title
A Modified Direct Preconditioner for Indefinite Symmetric Toeplitz Systems

Authors
Concus, P.
Saylor, P.

Publication Date
1992-11-01

LBL-33167
UC-405
Preprint

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

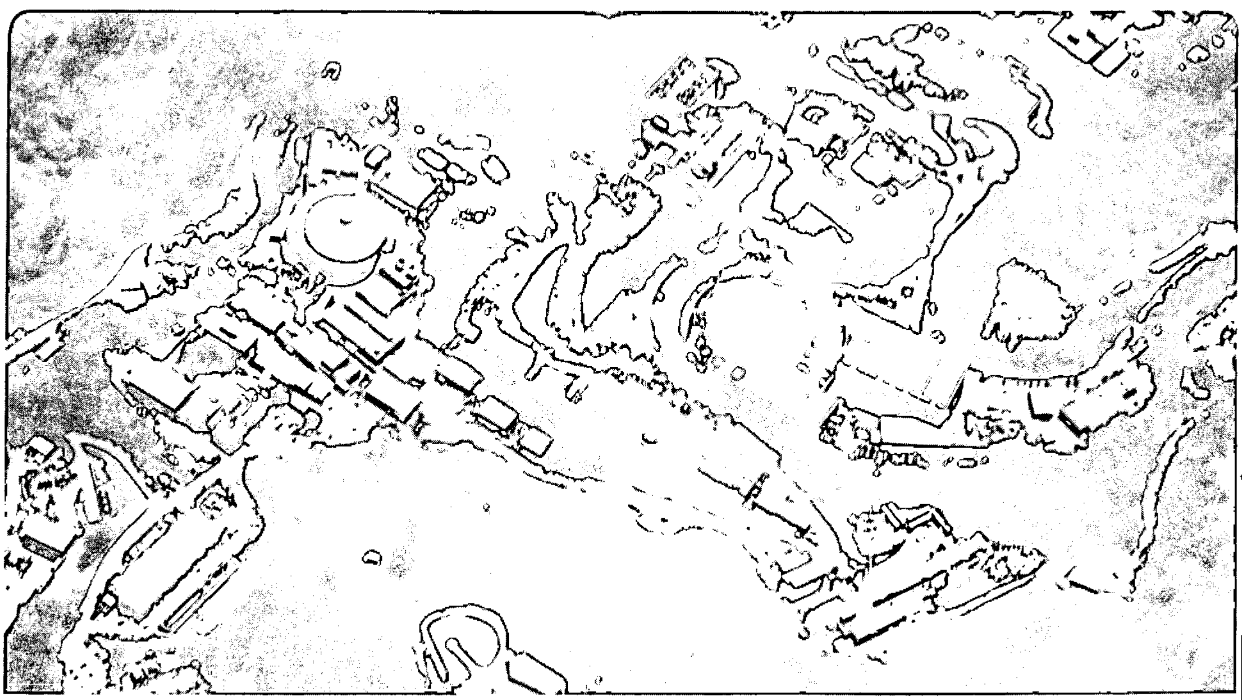
Physics Division

To be submitted for publication

A Modified Direct Preconditioner for Indefinite Symmetric Toeplitz Systems

P. Concus and P. Saylor

November 1992



Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

LBL-33167
UIUCDCS-R-92-1782

A MODIFIED DIRECT PRECONDITIONER FOR INDEFINITE SYMMETRIC TOEPLITZ SYSTEMS*

Paul Concus
Lawrence Berkeley Laboratory
and
Department of Mathematics
University of California
Berkeley, CA 94720

Paul Saylor
Department of Computer Science
University of Illinois
Urbana, IL 61801

November 1992

*This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research of the U. S. Department of Energy under Contract DE-AC03-76SF00098 and by NSF grant DMS87-03226. The authors are indebted to the Swiss Federal Institute of Technology, Zürich, for hospitality during a phase of the research. The work of the second author was supported in part by a grant from the Schlumberger Foundation.

A MODIFIED DIRECT PRECONDITIONER FOR INDEFINITE SYMMETRIC TOEPLITZ SYSTEMS

Paul Concus
Lawrence Berkeley Laboratory
and Department of Mathematics
University of California
Berkeley, CA 94720
concus@lbl.gov

Paul E. Saylor
Department of Computer Science
University of Illinois
Urbana, IL 61801
saylor@cs.uiuc.edu

Abstract

A modification is presented of the classical $O(n^2)$ algorithm of Trench for the direct solution of Toeplitz systems of equations. The Trench algorithm can be guaranteed to be stable only for matrices that are (symmetric) positive definite; it is generally unstable otherwise. The modification permits extension of the algorithm to compute an approximate inverse in the indefinite symmetric case, for which the unmodified algorithm breaks down when principal submatrices are singular. As a preconditioner, this approximate inverse has an advantage that only matrix-vector multiplications are required for the solution of a linear system, without forward and backward solves. The approximate inverse so obtained can be sufficiently accurate, moreover, that, when it is used as a preconditioner for the applications investigated, subsequent iteration may not even be necessary. Numerical results are given for several test matrices. The perturbation to the original matrix that defines the modification is related to a perturbation in a quantity generated in the Trench algorithm; the associated stability of the Trench algorithm is discussed.

Keywords. Indefinite, inverse, Levinson algorithm, matrix, preconditioning, stability, symmetric, Toeplitz, Trench algorithm

AMS(MOS) subject classification. 65F05, 65F10, 65U05

1 Introduction

In an earlier paper [10] we report on initial findings in our study of preconditioned iterative methods for solving linear systems of equations with an indefinite symmetric Toeplitz coefficient matrix. The focus there is on circulant matrix preconditioners. It arose as an extension of work for the positive definite symmetric case by Olkin [18] and by Strang [19] (also see [6, 7, 8, 20]). Here we present a preconditioner that, in the applications we have investigated, can be sufficiently accurate that subsequent iteration is not necessary. It is based on a modification of the $O(n^2)$ Trench algorithm for direct inversion of a nonsingular $n \times n$ Toeplitz matrix. Although the Trench algorithm can break down or be unstable when the matrix is not (symmetric) positive definite, the modified algorithm is successful for indefinite symmetric matrices, for which breakdown would occur when principal submatrices are numerically singular. In the modification, the Trench algorithm is carried out on a suitably perturbed matrix nearby the original one, to obtain the desired approximate inverse. A different approach for the indefinite case is given in [9], in which a look-ahead technique is used to avoid singularities.

In the symmetric positive definite case, an advantage of circulant preconditioning is that the resulting clustering of the eigenvalues can yield a preconditioned conjugate gradient method so efficient that the total work is $O(n \ln n)$, and therefore is a method contending with various elegant $O(n \ln n)$ direct methods [1, 2, 3]. (The implementation of circulant preconditioning with the conjugate gradient method exploits the fact that the solution of a linear system with a circulant coefficient matrix is possible in $O(n \ln n)$ operations and that multiplication of a vector by a Toeplitz matrix also requires only $O(n \ln n)$ operations.)

In our earlier paper [10] our goal was to extend circulant preconditioning to indefinite symmetric Toeplitz systems. Although in several cases the results were promising, in important ones, such as for the discrete Helmholtz equation, the eigenvalues of the preconditioned system could appear in both the right and left half planes [10, p. 77], a situation particularly troublesome for iterative methods. Additionally, the class of matrices for which circulant preconditioning has been shown to yield good eigenvalue clustering is a restrictive one even for positive definite matrices. Here we consider a different approach for the indefinite case, which is based on the Trench algorithm. Since this is an $O(n^2)$ algorithm, the result may not be as efficient asymptotically as the $O(n \ln n)$ circulant preconditioning, but clustering of the eigenvalues about unity is, in our experiments, substantially superior to that calculated in [10] for circulant preconditioners.

An outline of the paper is as follows. In the next section we define some basic terms and state the Trench algorithm, together with a well-known decomposition that the algorithm yields of the given Toeplitz matrix. In the third section we describe the effect of indefiniteness on the Trench algorithm and propose a perturbation technique to allow the algorithm to proceed when encountering a potential divide by zero. In the subsequent sections we describe a set of test matrices and summarize the results of numerical experiments.

As a supplemental remark, we note that there is a need for the iterative solution of indefinite symmetric and nonsymmetric linear systems in applications such as signal processing [17]. Our modification technique can apply to nonsymmetric as well as symmetric cases. (Theorem 2 in §3.1 uses symmetry, however). For these applications, although the coefficient matrices are rarely Toeplitz, Toeplitz matrices often suggest themselves as the basis for preconditioners of the given matrices. For example, it might be reasonable to approximate a block tridiagonal matrix A by the block submatrices on the diagonal. This yields a block diagonal matrix Q_1 whose inverse serves as a preconditioner of A (block Jacobi preconditioning). Each block submatrix of Q_1 in turn might be well approximated by a Toeplitz submatrix, which yields a block Toeplitz matrix Q_2. The inverse of Q_2 may then be obtained by computing the inverse of each Toeplitz submatrix. (There are many ways to get an approximation involving Toeplitz matrices; this suggestion is only illustrative.) Finally, using our technique, the inverse of each Toeplitz submatrix may be approximated, which yields a matrix $C \approx Q_2^{-1}$ that is a preconditioner for A.

There is an important observation to make about such a preconditioner. The fact that C is explicitly given is an aid to computing on high performance processors, because in an iterative method the preconditioning step would require only the multiplication of the matrix C times a vector; none of the forward and backward solves associated with preconditioners such as those obtained by incomplete factorization would be necessary.

2 Toeplitz Matrices

Let A denote a real symmetric $n \times n$ Toeplitz matrix. A Toeplitz matrix is constant along its diagonals and thus, in the symmetric case, is determined by the n elements of the first row, $a_0, a_1, \ldots, a_{n-1}$,

$$A = \begin{bmatrix} a_0 & a_1 & \ldots & a_{n-2} & a_{n-1} \\ a_1 & a_0 & a_1 & \ddots & a_{n-2} \\ a_2 & a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_1 \\ a_{n-1} & \ldots & a_2 & a_1 & a_0 \end{bmatrix}. \tag{1}$$

Such matrices arise in many areas, such as the theory of Padé approximation and the numerical solution of differential and integral equations. They are important also in time series analysis and signal processing.

2.1 The Trench Algorithm

Our presentation of the Trench algorithm follows the one given in [13, pp. 184ff.]. It requires the given Toeplitz matrix to have a unit main diagonal, i.e., $a_0 = 1$. We shall use T_n to denote the special case of a symmetric $n \times n$ Toeplitz matrix with unit main diagonal,

$$T_n = \begin{bmatrix} 1 & r_1 & \ldots & r_{n-2} & r_{n-1} \\ r_1 & 1 & r_1 & \ldots & r_{n-2} \\ \vdots & r_1 & 1 & \ddots & \vdots \\ r_{n-2} & \ddots & \ddots & \ddots & r_1 \\ r_{n-1} & r_{n-2} & \ldots & r_1 & 1 \end{bmatrix}. \tag{2}$$

Also, it will be convenient to define $r^{(k)} = (r_1, \ldots, r_k)^\top$.

A crucial property of Toeplitz matrices is that the inverse of a Toeplitz matrix does not necessarily have the constant-diagonal Toeplitz property, but is *persymmetric*, i.e., symmetric with respect to the main cross diagonal. A precise definition uses the *exchange matrix*. For $k \geq 2$, the $k \times k$ exchange matrix E_k is defined recursively to be

$$E_k = \begin{bmatrix} 0 & \ldots & 0 & 1 \\ & & & 0 \\ & E_{k-1} & & \vdots \\ & & & 0 \end{bmatrix},$$

where $E_1 = 1$. A $k \times k$ persymmetric matrix B satisfies $B = E_k B^\top E_k$.

We now have the necessary algebraic framework and proceed to the Trench algorithm. The ingenious discovery of the fundamental idea was first due to Levinson [16], embodied in the Levinson algorithm. The Levinson algorithm is often given in terms of the Durbin algorithm [12], which solves a Toeplitz linear system for which the right-hand side is a special case, $T_k y^{(k)} = -r^{(k)}$, collectively called the Yule-Walker equations. Accordingly, we shall refer to this as the Levinson-Durbin algorithm. The Trench algorithm builds upon the Levinson-Durbin algorithm to compute the inverse of the given (unit diagonal) Toeplitz matrix. The customarily used Zohar variant [21] of the Trench algorithm, which is the one considered here, improves the efficiency.

At step k, the algorithm builds the inverse of T_{k+1} from the inverse of T_k by using the solution of the Yule-Walker equations. In turn, the Levinson-Durbin algorithm uses the persymmetry property of the inverse of a Toeplitz matrix to obtain the solution of the Yule-Walker equations with only $O(k)$ operations. A complete derivation is given in [13]. A statement of the Trench algorithm is as follows:

Statement of the Algorithm.

Assume T_n is a given unit-main-diagonal Toeplitz matrix, as in (2).

1. Set $y^{(1)} := -r_1$.
2. Set $\beta_1 := 1 - r_1^2$.
3. $$\text{Form } T_2^{-1} := \frac{1}{\beta_1}\begin{bmatrix} 1 & y^{(1)} \\ y^{(1)} & 1 \end{bmatrix}.$$
4. **For** $k = 2 : n-1$
   (a) Set
   $$\alpha_{k-1} := \frac{-r_k - r^{(k-1)\top} E_{k-1} y^{(k-1)}}{\beta_{k-1}}. \tag{3}$$
   (b) Set $z^{(k-1)} := y^{(k-1)} + \alpha_{k-1} E_{k-1} y^{(k-1)}$.
   (c) Set $y^{(k)\top} := (z^{(k-1)\top}, \alpha_{k-1})^\top$.
   (d) Set
   $$\beta_k := \beta_{k-1}(1 - \alpha_{k-1}^2). \tag{4}$$
   {The previous steps (a), ..., (d) constitute the Levinson-Durbin algorithm for the solution of the Yule-Walker equations $T_k y^{(k)} = -r^{(k)}$. }
   (e)
   $$\text{Form } T_{k+1}^{-1} := \frac{1}{\beta_k}\begin{bmatrix} \beta_k T_k^{-1} + E_k y^{(k)} (E_k y^{(k)})^\top & E_k y^{(k)} \\ y^{(k)\top} E_k & 1 \end{bmatrix}. \tag{5}$$
5. **End**

The matrix T_n^{-1} thus obtained is the computed inverse of T_n.

REMARK 1. Persymmetry of the inverse can be exploited to increase the efficiency of step (e). See [13, p. 190].

2.2 Decomposition of the Matrix Inverse

The Levinson-Durbin algorithm for the Yule-Walker equations generates an $\mathrm{LDL}^{\mathsf{T}}$ decomposition of T_{k+1}^{-1} [11, p. 308],

$$T_{k+1}^{-1} = Y_{k+1}\Lambda_{k+1}^{-1}Y_{k+1}^{\mathsf{T}}, \tag{6}$$

where

$$Y_{k+1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ & 1 & \dots & 0 \\ & & \ddots & \vdots \\ & & 1 & 0 \\ y^{(k)} & y^{(k-1)} & \dots \; y^{(1)} & 1 \end{bmatrix}, \qquad y^{(j)} = \begin{bmatrix} y_1^{(j)} \\ y_2^{(j)} \\ \vdots \\ y_j^{(j)} \end{bmatrix}, \; 1 \le j \le k, \tag{7}$$

and Λ_{k+1} is the diagonal matrix $\Lambda_{k+1} = \mathrm{diag}\,[\beta_k, \dots, \beta_1, 1]$.

Some facts about positive definite Toeplitz matrices follow from this decomposition.

THEOREM 1. *Assume T_n is symmetric. The following statements are equivalent:*

1. *T_n is positive definite.*
2. *$\beta_k > 0$, for $1 \le k \le n-1$.*
3. *$1 - \alpha_k^2 > 0$, for $1 \le k \le n-2$.*

Proof: (Also see [13].) From (6), it follows that Λ_{k+1} is positive definite if and only if T_{k+1} is positive definite. Since the diagonal of Λ_{k+1} is $(\beta_k, \dots, \beta_1)$, we have (*2.*). The equivalence of (*3.*) follows from step (4d) of the Trench algorithm. □

If the inverse were known, its $\mathrm{LDL}^{\mathsf{T}}$ decomposition could be computed from a standard $\mathrm{LDL}^{\mathsf{T}}$ decomposition algorithm. The Trench algorithm does this, but with a different ordering of the elimination operations in which multiples of columns $n, n-1, \dots, 2$ are added to appropriate columns so as to eliminate columns under the diagonal. The Trench algorithm does more than compute a decomposition, of course, since it also computes the explicit inverse. Some insight into the organization of the algorithm comes from the observation that the columns of the $\mathrm{LDL}^{\mathsf{T}}$ decomposition result from the computation of polynomials orthogonal with respect to the bilinear form defined by T_n [14, 15].

3 Indefiniteness

There are two notable studies of stability of the Levinson-Durbin algorithm and the Trench algorithm [4, 11]. In [11] G. Cybenko shows that if T_n is symmetric positive definite, then the errors resulting from roundoff in the Levinson-Durbin algorithm are no worse than for Gaussian elimination, which is a known stable algorithm. To be precise, in [5] J. Bunch defines *weakly stable* algorithms and states that Cybenko proves the Trench algorithm is weakly stable in the case of symmetric positive definite T_n. In [4] Bunch discusses the causes and consequences of failure when T_n is not symmetric positive definite.

We see that the Trench algorithm fails at step j if $\beta_j = 0$. From (6), this occurs if and only if T_{j+1} is singular. If $\beta_j \neq 0$ is of small magnitude, Bunch makes the wry observation [4, p. 357] that this situation is worse, since the algorithm may not fail, depending on the magnitude, but could give incorrect results. We observe that if the results are incorrect, then it is possible to improve accuracy, if desired, by invoking an iterative method. This is our approach, which we describe below in the event that there is a singular principal submatrix of T_n.

If there exists a singular principal submatrix T_{j+1} for $j \leq n-2$ and the principal submatrices T_i, $1 \leq i \leq j$, are nonsingular, then the Trench algorithm may be carried out until T_{j+1} is encountered, i.e., until $\beta_j = 0$. From (4) it follows that $\alpha_{j-1}^2 = 1$. This difficulty may be interpreted as resulting from the newly introduced element r_j of T_{j+1}, in the expression for α_{j-1} in (3). Our approach now is to perturb r_j by subtracting a small quantity δ, which yields a non-zero β_j, thereby allowing the algorithm to proceed, albeit with only approximate results. (The procedure could be developed, alternatively, as one in which parameters within the algorithm, such as α or β in (3, 4), are perturbed; we follow the approach of perturbing the matrix elements for the numerical experiments discussed here.)

In selecting δ, a balance needs to be maintained between a value small enough so that the perturbed matrix is close to the original one and large enough so that the Trench algorithm will not suffer undue loss of significance resulting from finite machine precision. As the Trench algorithm may require the computation of quantities involving the sum of terms of order unity and order δ^2 when submatrices are nearly singular, a value of δ somewhat larger than the order of $\sqrt{\epsilon_m}$, where ϵ_m is the machine floating point relative accuracy, suggests itself, particularly for matrices having several singular submatrices. These questions and those of scaling, in general, will be pursued further in a subsequent study.

A simple example illustrates the effects of a perturbation. Let

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The principal submatrix (i.e., 0) is singular. Thus, it is necessary to perturb the diagonal, which yields the perturbed matrix

$$\tilde{A} = \begin{bmatrix} -\delta & 1 \\ 1 & -\delta \end{bmatrix}.$$

Scaling gives a unit diagonal Toeplitz matrix,

$$\tilde{A} = -\delta \begin{bmatrix} 1 & -\frac{1}{\delta} \\ -\frac{1}{\delta} & 1 \end{bmatrix} = -\delta T. \tag{8}$$

The Trench algorithm computes the inverse of the unit diagonal Toeplitz matrix in (8),

$$T^{-1} = \frac{\delta^2}{\delta^2 - 1} \begin{bmatrix} 1 & \frac{1}{\delta} \\ \frac{1}{\delta} & 1 \end{bmatrix},$$

from which one obtains

$$\tilde{A}^{-1} = -\frac{1}{\delta} T^{-1} = \frac{1}{1-\delta^2} \begin{bmatrix} \delta & 1 \\ 1 & \delta \end{bmatrix}.$$

The true inverse is

$$A^{-1} = A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Therefore,

$$\|A^{-1} - \tilde{A}^{-1}\|_\infty = |\delta| + O(\delta^2),$$

$$\tilde{A}^{-1}A = \frac{1}{1-\delta^2}\left[\begin{array}{cc} 1 & \delta \\ \delta & 1 \end{array}\right] = I + O(\delta),$$

and

$$\|I - \tilde{A}^{-1}A\|_\infty = |\delta| + O(\delta^2).$$

Considering $\tilde{A}^{-1}$ as a preconditioner for solving linear systems with coefficient matrix A, one has that the eigenvalues of the preconditioned system matrix $\tilde{A}^{-1}A$ are then $1 \pm \delta + O(\delta^2)$.

It may be helpful to explain our points about accuracy in the Trench algorithm by means of a more familiar model. Let us consider Gaussian elimination with no pivoting to see how it is possible to recover some level of accuracy from the unstable floating point computation considered above. Since $A^{-1} = A$, we take $\tilde{A}$ as a perturbation of A^{-1} and carry out the LU decomposition of T^{-1}, which is equivalent to the Trench algorithm, to obtain

$$\tilde{A}^{-1} = \frac{1}{1-\delta^2}\left[\begin{array}{cc} 1 & 0 \\ \frac{1}{\delta} & 1 \end{array}\right]\left[\begin{array}{cc} \delta & 1 \\ 0 & -\frac{1-\delta^2}{\delta} \end{array}\right].$$

(To be precise, the decomposition is the $LU = L(DL^{\mathsf{T}})$ decomposition.) In the Trench algorithm, the inverse is formed from the elements of the LDL^{T} factors of the inverse. If T is well-conditioned, as it is in this example, then the elements of T^{-1} are comparable in magnitude to those of T. To obtain these quantities, the large quantities from the algorithm must be collapsed in the final stages of the algorithm, which is the vulnerable step where accuracy is lost. The product is

$$\tilde{A}^{-1} = \frac{1}{1-\delta^2}\left[\begin{array}{cc} \delta & 1 \\ 1 & \frac{1}{\delta} - \frac{1-\delta^2}{\delta} \end{array}\right]. \tag{9}$$

For the (2, 2) element, where subtraction collapses large quantities to smaller ones, if $|\delta|$ is larger than $\sqrt{\epsilon_m}$, some accuracy is, nevertheless, retained in floating point arithmetic. For example, for a value of δ of about $10\sqrt{\epsilon_m}$, there are approximately two digits of accuracy in the (2, 2) element.

3.1 Perturbation Bounds

In our numerical experiments to be described, we investigate the effect of perturbing the elements along a diagonal of the given matrix A. It is known *a priori* where to make the perturbation for the test matrices. In order to develop perturbation strategies for a general, dynamic algorithm it is necessary to work with the Trench parameters. In such an algorithm, one might monitor these parameters and control them by perturbations to avert an impending singularity or near-singularity. Of the two parameters α and β, related by (4), α is the more convenient to consider, as it is so nicely scaled near a singularity, where $|\alpha| = 1$. We consider now a scaled matrix T_n having unit main diagonal, as required for the Trench algorithm. We obtain an estimate relating the change in T_n to a change in α.

Suppose a principal submatrix T_{j+1}, $0 \le j \le n-2$, is singular or nearly singular. Singularity of T_{j+1} is equivalent to

$$\beta_j = \beta_{j-1}(1 - \alpha_{j-1}^2) = 0,$$

i.e., $\alpha_{j-1}^2 = 1$. For near-singularity there holds $|\alpha_{j-1}| = 1 - \mu$, with $|\mu| \ll 1$. Suppose we perturb α_{j-1} to distance it further in absolute value from 1 so that $\tilde{\alpha}_{j-1}$, the perturbed α_{j-1}, satisfies $\tilde{\alpha}_{j-1} = \alpha_{j-1} - \eta$, where $|\eta| \ll 1$ and also $|\eta| \ge |\mu|$. Denote the corresponding perturbed value of r_j by $r_j - \delta$.

In the following theorem relating δ and η we use the 1-norm. In particular, for a matrix M, the condition number with respect to the 1-norm is $\text{cond}(M) = \|M\|_1 \|M^{-1}\|_1$.

Theorem 2. *Let δ, η, and μ be as above. Then*

$$|\delta| = K\,|\eta|\,[1+O(\eta)], \tag{10}$$

where $K \geq 0$ satisfies

$$K \leq \left[1+\left(1+\frac{|r_{j-1}|}{\|T_{j-1}\|}\right)\text{cond}(T_{j-1})\right](\|T_n\|-1). \tag{11}$$

Proof. From (3) one obtains

$$\eta = \alpha_{j-1} - \tilde{\alpha}_{j-1} = \frac{-\delta}{\beta_{j-1}}.$$

Let ϕ denote the numerator of α_{j-1} in (3). Then

$$|\beta_{j-1}| = \frac{|\phi|}{|\alpha_{j-1}|} = \frac{|\phi|}{1-\mu}.$$

There follows

$$|\delta| = |\phi|\frac{|\eta|}{1-\mu} = |\phi|\,|\eta|\,[1+O(\eta)]. \tag{12}$$

Now consider the terms in

$$|\phi| = |r_j + r^{(j-1)\top}E_{j-1}y^{(j-1)}|.$$

Since E_{j-1} is orthogonal and $y^{(j-1)} = -T_{j-1}^{-1}r^{(j-1)}$, it follows that

$$|r^{(j-1)\top}E_{j-1}y^{(j-1)}| \leq \|r^{(j-1)}\|^2\|T_{j-1}^{-1}\|.$$

But, from simple properties of a Toeplitz matrix,

$$\|r^{(j-1)}\| \leq \|r^{(n-1)}\| = \|T_n\|-1$$

and

$$\|r^{(j-1)}\| = \|T_{j-1}\| + |r_{j-1}|.$$

Therefore,

$$\|r^{(j-1)}\|^2\|T_{j-1}^{-1}\| \leq \left[\left(1+\frac{|r_{j-1}|}{\|T_{j-1}\|}\right)\text{cond}(T_{j-1})\right](\|T_n\|-1). \tag{13}$$

Since $|r_j| \leq \|T_n\|-1$, there holds

$$|\phi| \leq |r_j| + |r^{(j-1)\top}E_{j-1}y^{(j-1)}| \leq \left[1+\left(1+\frac{|r_{j-1}|}{\|T_{j-1}\|}\right)\text{cond}(T_{j-1})\right](\|T_n\|-1),$$

from which (10, 11) follow. □

Remark 2. If T_{j-1} is well conditioned, then the bound for the relative change $|\delta|/\|T_n\|$ in T_n is comparable with $|\eta|$, but if T_{j-1} is poorly conditioned, then the relative change might be much larger.

3.2 Effect of the Perturbation on the Algorithm

We shall use an example of Bunch [4, p. 353] to consider further the effect of a perturbation on the Trench algorithm. Our concern is that the algorithm should not only proceed to completion but also that the matrix it yields be a reasonable approximation to the inverse of the given Toepltiz matrix. Let

$$T_3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}. \tag{14}$$

The given Toeplitz matrix T_3 is well-conditioned; its condition number with respect to the l_1-norm is 9. However, since T_3 is indefinite, it is possible that the principal submatrix,

$$T_2 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

is singular, which, indeed, is the case. A perturbation yields the matrix

$$\tilde{T}_3 = \begin{bmatrix} 1 & 1-\delta & 0 \\ 1-\delta & 1 & 1-\delta \\ 0 & 1-\delta & 1 \end{bmatrix}.$$

For small $|\delta|$ the principal submatrix $\tilde{T}_2$ of $\tilde{T}_3$ is merely badly conditioned, rather than singular. Normally this is still regarded as an alarming fact, but we shall consider how it is possible for the Trench algorithm to proceed, nevertheless, to a satisfactory result.

The Trench algorithm yields $\beta_1(\delta) = \delta(2-\delta)$ and

$$\tilde{T}_2^{-1} = \frac{1}{\delta(2-\delta)} \begin{bmatrix} 1 & -(1-\delta) \\ -(1-\delta) & 1 \end{bmatrix},$$

for which the condition is approximately $1/\delta$. This matrix is used in the Trench algorithm to form the 2×2 principal submatrix of $\tilde{T}_3^{-1}$, which we shall denote by M. As is given by (5), the expression for M is

$$\beta_2(\delta) M = \beta_2(\delta)\tilde{T}_2^{-1} + E_2 y^{(2)} (E_2 y^{(2)})^{\mathsf{T}}. \tag{15}$$

Here $\beta_2(\delta) = \frac{4\delta-1}{2\delta} + O(\delta)$ is of large magnitude, which contributes to the well-conditioning of M, constructed from the badly conditioned inverse of $\tilde{T}_2$. Since the elements of M are approximately 9 times larger than those of $\tilde{T}_2$, the elements of $\tilde{T}_2^{-1}$, which are of magnitude $O(\delta^{-1})$, must cancel with the roughly equal magnitude elements of $E_2 y^{(2)} (E_2 y^{(2)})^{\mathsf{T}}$ in order to collapse to the small magnitude elements of M. Indeed, this can be seen from the (1, 1) element of $\beta_2(\delta)M$, which is equal to 1, although it is the difference of two terms, each of which is $O(1/\delta^2)$.

4 Test Matrices

For our test matrices, we use (nonsingular) indefinite symmetric Toeplitz matrices A constructed so that the Trench algorithm will break down because a principal submatrix is singular. The first test matrices are obtained from those used in [10] by applying diagonal shifts so that successive submatrices become singular. The second group of test matrices is one for which the nonzero elements are a unit main diagonal and an additional unit off-diagonal band, symmetrically placed above and below the main diagonal. For the third group, matrices with zero diagonal are considered, with unit off-diagonal bands as in the second group. As in [10], we take $n = 16$ for all the tests.

The first test matrices are taken from positive definite examples used in [8], which are then shifted by a constant diagonal matrix to make some of the eigenvalues negative. All of the original matrices have distinct eigenvalues. If $\hat{A}$ denotes the original positive definite matrix, then the indefinite one $\breve{A}$ for these examples is obtained by $\breve{A} = \hat{A} - \alpha I$, where α is the average of the fifth and sixth eigenvalues of $\hat{A}$, ordered increasingly. The test matrices A used here are then obtained by the set of shifts $A = A^{(\ell)} = \breve{A} - c_\ell I$, $\ell = 1, 2, \ldots, n-2$, where c_ℓ is the number of smallest magnitude so that the $(\ell+1) \times (\ell+1)$ principal submatrix of $A^{(\ell)}$ will have a zero eigenvalue.

The underlying positive definite matrices $\hat{A}$, before shifting, are the following.

1. $\hat{a}_k = \left(\frac{1}{2}\right)^k$, $k = 0, 1, \ldots, n-1$. For this case $\alpha \approx 0.4320$, and $\breve{a}_0 \approx 0.5680$.
2. $\hat{a}_k = 1/(k+1)$, $k = 0, 1, \ldots, n-1$. For this case $\alpha \approx 0.4582$, and $\breve{a}_0 \approx 0.5418$.
3. $\hat{a}_k = \frac{\cos k}{k+1}$, $k = 0, 1, \ldots, n-1$. For this case $\alpha \approx 0.5878$, and $\breve{a}_0 \approx 0.4122$.
4. $\hat{a}_0 = 2$, $\hat{a}_1 = -1$, $\hat{a}_2 = \hat{a}_3 = \cdots = \hat{a}_{n-1} = 0$. For this case $\alpha \approx 0.9516$, and $\breve{a}_0 \approx 1.0484$.

For cases (1), (2), and (3), all principal submatrices of the $A^{(\ell)}$, except the $(\ell+1) \times (\ell+1)$ one, are nonsingular. For case (4) several principal submatrices of $A^{(\ell)}$ other than the $(\ell+1) \times (\ell+1)$ one may be singular.

The second group of test matrices is given by

$$a_k = \begin{cases} 1, & k = 0 \\ 1, & k = j \\ 0, & \text{otherwise,} \end{cases}$$

where the value j is such that $1 \le j \le n-2$. For $k < j$, the principal $(k+1) \times (k+1)$ submatrices are nonsingular (they are simply identity matrices); the $(j+1) \times (j+1)$ principal submatrix

$$\begin{bmatrix} 1 & 0 & \ldots & 0 & 1 \\ 0 & 1 & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 & 0 \\ 1 & 0 & \ldots & 0 & 1 \end{bmatrix}$$

is singular. The subsequent submatrices for $j < k < n-1$ and the entire $n \times n$ matrix A may or may not be singular. We take $j = 1$, 4, and 5, which are the cases for which A is nonsingular. For these cases A is indefinite, and generally there are several singular principal submatrices.

For the final test matrices, we consider ones with zero on the diagonal. They are obtained from the second group by setting the main diagonal to zeros instead of ones,

$$a_k = \begin{cases} 1, & k = j \\ 0, & \text{otherwise.} \end{cases}$$

In this case the matrices that are nonsingular, the ones taken for our tests, are for $j = 1, 2, 4$, and 8. They are all indefinite, with eight positive and eight negative eigenvalues, and have numerous singular principal submatrices, eight for the $j = 1$ case to fifteen for the $j = 8$ case. The inverses can be expressed simply; for example the inverse of the $j = 2$ matrix is the block Toeplitz matrix

$$\begin{bmatrix} D & -E & E & -E \\ -E^\top & D & -E & E \\ E^\top & -E^\top & D & -E \\ -E^\top & E^\top & -E & D \end{bmatrix},$$

where D and E are the Toeplitz matrices

$$D = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The matrix D, which is the block diagonal of the original matrix, has the property that $D = D^{-1}$.

Inverses of the other matrices in the third group can be expressed in a similar fashion. For $j = 8$ there holds simply that $A = A^{-1}$. The inverses of the (nonsingular) test matrices from the second group, and of $A^{(1)} = A^{(4)} = A^{(7)} = A^{(10)} = A^{(13)}$ of case (4) for the first group, for which $c_\ell = 1$, can be expressed also as patterns of 0's, 1's, and -1's.

5 Numerical Results

In conducting our numerical experiments, our procedure was as follows. If a test matrix A had a singular $(j+1)\times(j+1)$ principal submatrix and nonsingular $(k+1)\times(k+1)$ principal submatrices for $0 \le k < j$, then we perturbed the elements a_j of A by an amount $-\delta$ along diagonal $j+1$. This yielded a perturbed matrix $\tilde{A}$, for which the elements on diagonal $j+1$ became $a_j - \delta$.

For the experiments reported here, we selected as the value of primary interest $\delta = 10^{-7}$, which is approximately $10\sqrt{\epsilon_m}$ for the computer on which the calculations were carried out (in double precision). We then compared the results with those for $\delta = 10^{-8}$ and $\delta = 10^{-6}$.

The inverse C of the perturbed matrix $\tilde{A}$ was computed by the Trench algorithm using MATLAB, with the ordering of operations as given in [4, pp. 356–357]. (The matrix $\tilde{A}$ was scaled to have unit diagonal prior to entering the Trench algorithm, and then the matrix obtained was scaled back, to yield a computed inverse to the original $\tilde{A}$.) In all cases, except for matrices in the final group with $\delta = 10^{-8}$, the algorithm completed successfully. In those cases for which subsequent principal submatrices of the unperturbed matrix were singular, the second and third groups and example (4) of the first group, perturbation of the earlier elements removed the later singularities as well. In this connection, we note that for the matrices in the third group, the perturbations resulted simply in replacement of the zero diagonal elements with $-\delta$.

For $\delta = 10^{-7}$, the computed eigenvalues of the preconditioned matrix CA for the first and second group of test matrices were all equal to unity to MATLAB short precision format (four decimal places), as indicated in the Table. Thus, for these test problems, C can be considered as an excellent preconditioner for solving a system of equations, approximating A^{-1} so well that iteration may not even be necessary. For this δ, agreement with unity of the eigenvalues of $C\tilde{A}$, a measure of the accuracy in carrying out the Trench algorithm, was several decimal places better for the first group of test matrices and one or two places better for the second group. Comparison with the computations for $\delta = 10^{-8}$ and $\delta = 10^{-6}$ show that for the smaller δ the Trench algorithm accuracy was less and for the larger δ greater. For example, for $\delta = 10^{-6}$ the eigenvalues of $C\tilde{A}$ were unity to eight decimal places or more and for $\delta = 10^{-8}$ to about six places or more. For $\delta = 10^{-8}$, there was a reasonable balance between the errors committed by the perturbation and rounding errors in the Trench algorithm for the first and second group of test problems. Except for a few places in the last digit for the first group for $\delta = 10^{-6}$, all three values of δ yielded the overall result that for the first and second group of test problems CA was the identity matrix to MATLAB short precision format.

For the final group of test matrices, the loss of significance within the Trench algorithm was generally greater. As mentioned above, for $\delta = 10^{-8}$ significance was lost altogether for $j = 1$, 2,

Table

Accuracy of the Calculated Trench Inverse

	Max $\left|1 - |\text{eigenvalues of } CA|\right|$			Max $\left|1 - |\text{eigenvalues of } C\tilde{A}|\right|$		
δ	10^{-6}	10^{-7}	10^{-8}	10^{-6}	10^{-7}	10^{-8}
First Group						
1.	1.6E−4	1.6E−5	1.7E−6	1.7E−9	1.4E−8	3.3E−7
2.	1.9E−4	1.9E−5	1.9E−6	4.4E−9	1.4E−8	9.5E−8
3.	3.9E−4	3.9E−5	1.1E−6	2.1E−9	1.7E−7	2.7E−6
4.	1.9E−5	1.9E−6	2.6E−7	1.3E−9	2.7E−8	2.6E−7
Second Group						
$j = 1$	8.2E−6	8.4E−7	1.5E−7	7.4E−10	2.0E−8	6.7E−8
$j = 4$	2.6E−6	2.6E−7	2.7E−8	1.5E−10	3.5E−9	2.4E−9
$j = 5$	3.4E−6	3.4E−7	3.6E−8	1.5E−10	3.5E−9	2.4E−9
Third Group						
$j = 1$	2.5E−4	1.2E−2	NaN	2.5E−4	1.2E−2	NaN
$j = 2$	9.5E−5	8.5E−3	NaN	9.5E−5	8.5E−3	NaN
$j = 4$	6.7E−5	5.8E−3	NaN	6.6E−5	5.8E−3	NaN
$j = 8$	1.0E−6	1.0E−7	1.0E−8	0	0	1.1E−16

and 4; the algorithm stopped on an attempted divide by zero in computing β. However, for larger δ the results were satisfactory. The eigenvalues of CA were unity to two decimal places or better for $\delta = 10^{-7}$ and to four decimal places or better for $\delta = 10^{-6}$, except possibly for one or two places in the last digit. For $\delta = 10^{-6}$, the eigenvalues of C times $\tilde{A}$ were unity to the same number of places as C times the unperturbed A. The problem for $j = 8$ is a special case for which the computations are robust with respect to rounding errors.

6 Conclusions and Future Work

In this paper, we have presented preliminary results on an $O(n^2)$ technique to compute an approximate inverse to a symmetric indefinite Toeplitz matrix. Our approach, which preserves the Toeplitz structure, is simply to perturb the given Toeplitz matrix so as to allow the Trench algorithm to proceed. Our experiments show that the perturbation results in only small errors for the test problems. We give a relationship between the magnitude of the perturbation and the magnitude of perturbed quantities in the algorithm that relate to the condition number. A topic for future research is a dynamic algorithm to determine the magnitude of the perturbation so as to balance perturbation and resulting rounding errors. Some issues that relate to this topic overlap with those treated in [9].

7 Acknowledgments

This work was supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research of the U. S. Department of Energy under Contract DE-AC03-76SF00098 and by NSF grant DMS87-03226. We are indebted to the Swiss Federal Institute of Technology, Zürich, for hospitality during a phase of the research. The work of the second author was supported in part by a grant from the Schlumberger Foundation.

References

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

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

[3] R. P. Brent, G. F. Gustavson, and D. Y. Y. Yun. Fast solution of Toeplitz systems of equations and computation of padé approximants. *J. Algorithms*, 1:259–295, 1980.

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

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

[6] R. H. Chan. Circulant preconditioners for hermitian Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 10:542–550, 1989.

[7] R. H. Chan and G. Strang. Toeplitz equations by conjugate gradients with circulant preconditioner. *SIAM J. Sci. Statist. Comput.*, 10:104–119, 1989.

[8] T. F. Chan. An optimal circulant preconditioner for Toeplitz systems. *SIAM J. Sci. Statist. Comput.*, 9:766–771, 1988.

[9] T. F. Chan and P. C. Hansen. A look-ahead Levinson algorithm for indefinite Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 13(2):490–506, 1992.

[10] P. Concus and P. Saylor. Preconditioned iterative methods for indefinite symmetric toeplitz systems. In L. Hayes and D. Kincaid, editors, *Iterative Methods for Large Linear Systems*, pages 309–332. Academic Press, New York, 1990.

[11] G. Cybenko. The numerical stability of the Levinson-Durbin algorithm for Toeplitz systems of equations. *SIAM J. Sci. Statist. Comput.*, 1:303–310, 1980.

[12] J. Durbin. The fitting of time series models. *Review of the International Statisical Institute*, 28:233–243, 1960.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, Maryland, 1989.

[14] T. Kailath, B. Levy, L. Ljung, and M. Morf. The factorization and representation of operators in the algebra generated by Toeplitz operators. *SIAM J. Appl. Math.*, 37:467–484, 1979.

[15] T. Kailath, A. Vieira, and M. Morf. Inverses of Toeplitz operators, innovations and orthogonal polynomials. *SIAM Rev.*, 20:106–119, 1978.

[16] N. Levinson. The Wiener RMS error criterion in filter design and prediction. *J. Math. Phys.*, 25:261–278, 1947.

[17] W. F. Mitchell and J. D'Angelo. A nonsymmetric non-hermitian complex matrix. In *Proceedings of the Copper Mountain Conference on iterative methods.* The University of Colorado at Denver, April 1990.

[18] J. A. Olkin. *Linear and nonlinear deconvolution models.* PhD thesis, Dept. of Math. Sci., Rice University, 1986. Available as Technical Report 86-10.

[19] G. Strang. A proposal for Toeplitz matrix calculations. *Stud. Appl. Math.*, 74:171–176, 1986.

[20] G. Strang and A. Edelman. The Toeplitz-circulant eigenvalue problem $Ax = \lambda Cx$. In L. Bragg and J. Dettman, editors, *Oakland Conf. on PDE's*, pages 109–117. Longman, 1987.

[21] S. Zohar. Toeplitz matrix inversion: The algorithm of W.F. Trench. *J. Assoc. Comp. Mach.*, 16:592–601, 1969.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720