Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Mathematics with Applications 50 (2005) 741–752
An International Journal
computers & mathematics
with applications
www.elsevier.com/locate/camwa

# A Fast Algorithm for the Inversion of General Toeplitz Matrices

P. G. MARTINSSON, V. ROKHLIN AND M. TYGERT
Department of Mathematics
Yale University, P.O. Box 208283
New Haven, CT 06520-8283, U.S.A.

*(Received February 2005; revised and accepted March 2005)*

**Abstract**—We propose a "fast" algorithm for the construction of a data-sparse inverse of a general Toeplitz matrix. The computational cost for inverting an $N \times N$ Toeplitz matrix equals the cost of four length-$N$ FFTs plus an $O(N)$-term. This cost should be compared to the $O(N \log^2 N)$ cost of previously published methods. Moreover, while those earlier methods are based on algebraic considerations, the procedure of this paper is analysis-based; as a result, its stability does not depend on the symmetry and positive-definiteness of the matrix being inverted. The performance of the scheme is illustrated with numerical examples. © 2005 Elsevier Ltd. All rights reserved.

**Keywords**—Toeplitz matrix, Fast algorithm, Direct inversion.

## 1. INTRODUCTION

We consider the problem of inverting an $N \times N$ Toeplitz matrix,

$$T = \begin{bmatrix} t_N & t_{N-1} & \cdots & t_1 \\ t_{N+1} & t_N & & t_2 \\ \vdots & & & \vdots \\ t_{2N-1} & & & t_N \end{bmatrix}, \tag{1}$$

with real or complex entries. Methods for computing $T^{-1}$ in $O(N^2)$ operations have been known since the 1960s, see, for example, [1,2]. For the case where $T$ is positive-definite, it is fairly well understood how to construct $T^{-1}$ stably in $O(N \log^2 N)$ operations, see [3] and the references therein. These techniques are based on algebraic properties implied by the Toeplitz structure. When such algebraic techniques are applied to general Toeplitz matrices, stability issues arise; methods for managing such issues (at the cost of sacrificing the $O(N \log^2 N)$ CPU time estimate for certain matrices) are presented in, e.g., [4,5].

In this paper, we use rank considerations, rather than algebraic properties, to construct a fast inversion scheme. Our approach depends crucially on the well-known fact that the representation of $T$ in Fourier space, $\hat{T}$, is semiseparable in the sense that its off-diagonal blocks can be

The first author was supported in part by the Office of Naval Research under Contract #N00014-01-1-0364. The second author was supported in part by the Defense Advanced Research Projects Agency under Contract #MDA972-00-1-0033.

0898-1221/05/$ - see front matter © 2005 Elsevier Ltd. All rights reserved. Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX
doi:10.1016/j.camwa.2005.03.011

approximated by low-rank matrices, see, [6–10]. Combining this fact with the observation that factorizations of these off-diagonal blocks can be obtained very rapidly, we demonstrate that the $O(N)$ inversion algorithm of [11] can be used to compute a compressed representation of $\hat{T}^{-1}$, whence $T^{-1}$ is readily applied via two FFTs. This inversion scheme is in no way dependent on either the positivity or the symmetry of $T$. Numerical experiments on general symmetric Toeplitz matrices indicate that the procedure is very stable. According to [7], similar results should be obtained if the inversion scheme of [12] were applied to the problem of inverting $\hat{T}$.

This paper is structured as follows. Section 2 introduces our notation. Section 3 lists some basic facts about Toeplitz matrices and their Fourier representations. Section 4 contains the proof that the Fourier representation of a Toeplitz matrix possesses the properties required for the fast inversion scheme of [11] to be applicable. Section 5 presents the results of several numerical examples.

## 2. PRELIMINARIES

In this section, we introduce a notational framework.

Let $N$ be a positive integer and let

$$t = [t(1), t(2), \ldots, t(2N-1)] \tag{2}$$

be a sequence of (possibly complex) numbers. We call this sequence the *Toeplitz-vector* that generates the *Toeplitz-matrix* $T \in \mathbb{C}^{N\times N}$ with elements,

$$T(m,n) = t(N+m-n). \tag{3}$$

As an example, for $N = 4$, we have

$$T = \begin{bmatrix} t(4) & t(3) & t(2) & t(1) \\ t(5) & t(4) & t(3) & t(2) \\ t(6) & t(5) & t(4) & t(3) \\ t(7) & t(6) & t(5) & t(4) \end{bmatrix}. \tag{4}$$

Matrices are always named using upper case letters while lower case letters are used for vectors.

We define a discrete Fourier transform $F$ by

$$[Fu](m) = \hat{u}(m) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{-2\pi i m(n-1/2)/N} u(n), \tag{5}$$

which is a unitary operator on $\mathbb{C}^N$. The inverse transform is thus given by

$$u(m) = [F^*\hat{u}](m) = \frac{1}{\sqrt{N}} \sum_{n=1}^{N} e^{2\pi i(m-1/2)n/N} \hat{u}(n). \tag{6}$$

(See Remark 2 for some comments on our choice of a discrete Fourier transform.) Then, the Fourier representation of $T$ is defined by

$$\hat{T} = \mathrm{F}T F^*. \tag{7}$$

We introduce a shift operator $S$, such that, for $u \in \mathbb{C}^N$,

$$[Su](n) = \begin{cases} u(n+1), & n = 1, \ldots, N-1, \\ u(1), & n = N, \end{cases} \tag{8}$$

and a flip operator $P$, such that

$$[Pu](n) = u(N+1-n). \tag{9}$$

As an example, for $N = 4$, $S$ and $P$ have the matrix representations,

$$S = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \text{and} \quad P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{10}$$

Throughout this paper, norms of vectors are $l^2$-norms, $\|v\| = (\sum_{j=1}^{N} |v_j|^2)^{1/2}$, and norms of matrices are the corresponding operator norms. Thus, if $A$ is an $m \times n$ matrix,

$$\|A\| = \max_{u \in \mathbb{C}^n} \frac{\|Au\|}{\|u\|}. \tag{11}$$

Finally, we introduce the concept of neutered block columns and rows. Suppose that $A$ is a matrix composed of $p \times p$ blocks. We say that the submatrix formed by all the blocks in say, the $j^{\text{th}}$ column of blocks, except the block on the diagonal, is the $j^{\text{th}}$ neutered block column. Neutered rows are defined analogously. See Figure 1 for an illustration.

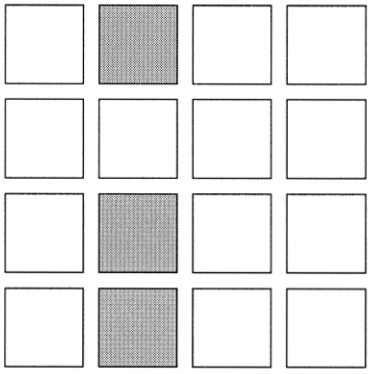

Figure 1. Illustration of the concept of a *neutered block column*. The figure shows a $4 \times 4$ block matrix. The submatrix formed by the three gray blocks is the "second neutered block column".

## 3. PROPERTIES OF THE FOURIER REPRESENTATION OF A TOEPLITZ MATRIX

This section contains two technical results. Lemma 1 gives a formula for the Fourier representation $\hat{T}$ of a Toeplitz matrix $T$, while Lemma 2 lists some its properties.

LEMMA 1. *The entries of the matrix $\hat{T}$ defined by (7) are given by*

$$\hat{T}(m,n) = \begin{cases} (\tilde{v}(m) - \tilde{v}(n))\,\varphi(m-n), & \text{if } m \neq n, \\ \tilde{x}(m), & \text{if } m = n. \end{cases} \tag{12}$$

*where, for $n = 1, \ldots, N$,*

$$v(n) = t(N+n) - t(n), \tag{13}$$

$$\tilde{v}(n) = e^{-\pi i n/N} \hat{v}(n), \tag{14}$$

$$x(n) = nt(n) + (N-n)\,t(N+n), \tag{15}$$

$$\tilde{x}(n) = e^{-\pi i n/N} \frac{\hat{x}(n)}{\sqrt{N}}, \tag{16}$$

*and*

$$\varphi(m) = -\frac{\mathrm{i}}{2\sqrt{N}\sin(\pi(m-n)/N)}. \tag{17}$$

PROOF. The Toeplitz matrix $T$ almost commutes with the shift operator $S$, see (8), in the sense that $ST - TS$ has rank two. To be precise,

$$ST - TS = ve_1^* - e_N v^* P, \tag{18}$$

where $v$ is defined by (13), $P$ is defined by (9), and $e_j \in \mathbb{C}^N$ is the $j^{\text{th}}$ canonical basis vector (so that e.g., $e_1 = [1, 0, \ldots, 0]^*$). On the Fourier side, equation (18) takes the form,

$$\hat{S}\hat{T} - \hat{T}\hat{S} = \hat{v}\hat{e}_1^* - \hat{e}_N \hat{v}^* \hat{P}. \tag{19}$$

Now, we use that $\hat{S}$ is a diagonal matrix with entries,

$$\hat{S}(n,n) = e^{2\pi \mathrm{i} n/N}, \tag{20}$$

and that, for $n = 1, \ldots, N$,

$$\hat{e}_1(n) = \left(\frac{1}{\sqrt{N}}\right) e^{-\pi \mathrm{i} n/N}, \tag{21}$$

$$\hat{e}_N(n) = \left(\frac{1}{\sqrt{N}}\right) e^{\pi \mathrm{i} n/N}, \tag{22}$$

$$\hat{v}^* \hat{P} = \hat{v}^{\mathrm{t}}, \tag{23}$$

to write (19) in component form,

$$\begin{aligned} &e^{2\pi \mathrm{i} m/N}\hat{T}(m,n) - \hat{T}(m,n)\, e^{2\pi \mathrm{i} n/N} \\ &= \left(\frac{1}{\sqrt{N}}\right)\hat{v}(m)\, e^{\pi \mathrm{i} n/N} - \left(\frac{1}{\sqrt{N}}\right) e^{\pi \mathrm{i} m/N}\hat{v}(n). \end{aligned} \tag{24}$$

Equation (12) follows directly from (24) when $m \neq n$.

To derive the expression for the diagonal elements of $\hat{T}$, we insert (3), (5), and (6) into (7) to obtain

$$\hat{T}(n,n) = \frac{1}{N}\sum_{p=1}^{N}\sum_{q=1}^{N} e^{-(2\pi \mathrm{i} n/N)(p-q)} t(N+p-q). \tag{25}$$

Changing summation variables to $r = N + p - q$ and $s = p - q$, we find that

$$\hat{T}(n,n) = \frac{1}{N}\sum_{r=1}^{N}\sum_{p=1}^{r} e^{-(2\pi \mathrm{i} n/N)(r-N)}\, t(r) + \frac{1}{N}\sum_{s=1}^{N-1}\sum_{p=1+s}^{N} e^{-(2\pi \mathrm{i} n/N)s} t(N+s) \tag{26}$$

$$= \frac{1}{N}\sum_{r=1}^{N} e^{-(2\pi \mathrm{i} n/N)r} r t(r) + \frac{1}{N}\sum_{s=1}^{N} e^{-(2\pi \mathrm{i} n/N)s}(N-s)\, t(N+s) \tag{27}$$

$$= \frac{1}{N}\sum_{r=1}^{N} e^{-(2\pi \mathrm{i} n/N)r}\left(r t(r) + (N-r)\, t(N+r)\right). \tag{28}$$

That $\hat{T}(n,n) = \tilde{x}(n)$ now follows from (15), (16), and (5). ∎

REMARK 1. It is frequently useful to write the formula (12) in matrix form. To this end, let us define an $N \times N$ matrix $Y$ by

$$Y(m,n) = \begin{cases} \varphi(m-n), & m \neq n, \\ 0, & m = n, \end{cases} \tag{29}$$

where $\varphi$ is defined by (17). Since $\varphi$ is $N$-periodic, the matrix $Y$ is circulant. Furthermore, let us define $X$ and $V$ as $N \times N$ diagonal matrices with nonzero elements,

$$X(n,n) = \tilde{x}(n), \tag{30}$$

$$V(n,n) = \tilde{v}(n). \tag{31}$$

Then, we can write (12) in matrix form as

$$\hat{T} = VY - YV + X. \tag{32}$$

In particular, using the index notation of [16], the off-diagonal block $\hat{T}(J_2, J_1)$ corresponding to two disjoint index sets $J_1$ and $J_2$, is given by the formula

$$\hat{T}(J_2, J_1) = V(J_2, J_2)\, Y(J_2, J_1) - Y(J_2, J_1)\, V(J_1, J_1). \tag{33}$$

LEMMA 2. *Let $T$ be a Toeplitz matrix (as defined by (3)) and let $\hat{T}$ be its Fourier representation (as defined by (7)). Then,*

(i) *the matrix $\hat{T}$ is always symmetric (but not necessarily Hermitian),*

$$\hat{T} = \hat{T}^{t}, \tag{34}$$

(ii) *if $T$ is real, then $\hat{T}$ is almost persymmetric in the sense that*

$$\hat{T}(m,n) = \overline{\hat{T}(N-n, N-m)}, \qquad \text{for } m,n = 1,\ldots,N-1, \tag{35}$$

$$\hat{T}(N,n) = \overline{\hat{T}(N,N-n)}, \qquad \text{for } n = 1,\ldots,N-1, \tag{36}$$

$$\hat{T}(m,N) = \overline{\hat{T}(N-m,N)}, \qquad \text{for } m = 1,\ldots,N-1, \tag{37}$$

(iii) *if $T$ is both real and symmetric, then $\hat{T}$ is real. In fact,*

$$\hat{T} = -\text{imag}(V)\,\text{imag}(Y) + \text{imag}(Y)\,\text{imag}(V) + \text{real}(X). \tag{38}$$

All of these statements are direct consequences of the definition of a Toeplitz matrix, (3), and the definition of its Fourier representation, (5)–(7).

REMARK 2. While the exact form of the results of Lemma 2 depends on the choice of a Fourier transform, some incarnation of the lemma holds for any reasonable choice. For instance, if we had used the common definition,

$$[Fu](n) = \frac{1}{\sqrt{N}} \sum_{m=1}^{N} e^{-2\pi i n m/N}, \tag{39}$$

then in the case that $T$ is real and symmetric, $\hat{T}$ would not be real, but its imaginary part would have had rank two. Our particular choice, given by (5), was motivated simply by the fact that it resulted in a cleaner formulation of Lemma 2 than the other options we tried.

## 4. THE FOURIER TRANSFORM OF A TOEPLITZ MATRIX IS SEMISEPARABLE

In this section, we will demonstrate that when $T$ is nonsingular, it is possible to compute a compressed representation for the inverse of the $N \times N$ matrix $\hat{T}$ defined by (7) using $O(N)$ arithmetic operations, once $\tilde{v}$ and $\tilde{x}$ (see (14),(16)) have been computed. We do this by using the fast inversion algorithm described in [11]. This algorithm is applicable to any matrix that satisfies the following two conditions,

(i) its off-diagonal blocks have low rank and
(ii) these blocks can be factored cheaply.

In this section, we will state these conditions in detail and prove that the matrix $\hat{T}$, satisfies both of them.

The following lemma states that to precision $\varepsilon$, the rank of any neutered block column of $\hat{T}$ (see Figure 1) is at most $O(\log(1/\varepsilon)\log N)$.

LEMMA 3. *Let $\hat{T}$ be the Fourier representation of a Toeplitz matrix $T$ (as defined by (7)), let $J_1 = [j, j+1, j+2, \ldots, j+n-1]$ be an index set and let $\hat{T}_{\text{NC}}$ be the corresponding neutered block column (see Figure 1). Then, for sufficiently small but positive $\varepsilon$, the matrix $\hat{T}_{\text{NC}}$ allows the factorization,*

$$\hat{T}_{\text{NC}} = WR + E, \tag{40}$$

*where the matrix $R$ has dimension $k \times n$ for some integer $k$ satisfying*

$$k \leq C \log N \log\left(\frac{1}{\varepsilon}\right), \tag{41}$$

*the matrix $W$ has dimension $(N-n) \times k$ and satisfies*

$$\|W\| \leq 1, \tag{42}$$

*and the remainder matrix $E$ has dimension $(N-n) \times n$ and satisfies*

$$\|E\| \leq 2\varepsilon \|v\|. \tag{43}$$

SKETCH OF PROOF. Let $J_2$ denote the indices not contained in $J_1$, i.e.,

$$J_2 = [1, 2, \ldots, j-1, j+n, \ldots, N].$$

From Remark 1 we know that

$$\hat{T}_{\text{NC}} = V_2 Y_{21} - Y_{21} V_1, \tag{44}$$

where for $i, j \in \{1, 2\}$,

$$V_i = V(J_i, J_i),$$
$$Y_{ij} = Y(J_i, J_j),$$

with $V$ and $Y$ defined by (31) and (29), respectively.

The entries of the matrix $Y$ are samples of the continuous function $z \mapsto (\sin z)^{-1}$, see (17),(29). Using this fact, calculations similar to those in [13] show that the singular values of $Y_{21}$ decay exponentially. Employing a standard rank-revealing QR-factorization, see [14], then, it follows that $Y_{21}$ admits the factorization,

$$Y_{21} = Q\tilde{R} + \tilde{E}, \tag{45}$$

where $Q$ is an $(N-n) \times k$ matrix with orthonormal columns for some $k$ satisfying (41), $\tilde{R}$ is a $k \times n$ matrix, and $\tilde{E}$ is an $(N-n) \times n$ matrix satisfying,

$$\left\|\tilde{E}\right\| \leq \varepsilon. \tag{46}$$

Combining (44) and (45), we find that

$$\hat{T}_{\text{NC}} = V_2 Q \tilde{R} - Q\tilde{R} V_1 + V_2 \tilde{E} - \tilde{E} V_1 \tag{47}$$

$$= \left[\frac{1}{\|\tilde{v}(J_2)\|} V_2 Q \quad Q\right] \begin{bmatrix} \tilde{R}\|\tilde{v}(J_2)\| \\ \tilde{R} V_1 \end{bmatrix} + V_2 \tilde{E} - \tilde{E} V_1. \tag{48}$$

To convert (47) into (40), we set

$$W = \left[\frac{1}{\|\tilde{v}(J_2)\|} V_2 Q \quad Q\right], \tag{49}$$

$$R = \begin{bmatrix} \tilde{R}\|\tilde{v}(J_2)\| \\ \tilde{R} V_1 \end{bmatrix}, \tag{50}$$

$$E = V_2 \tilde{E} - \tilde{E} V_1. \tag{51}$$

The bound (42) follows from (49) and the fact that $Q$ has orthonormal columns. The bound (43) follows from (46) and (51). ∎

The fast inversion algorithm that we use to invert $\hat{T}$ requires only very limited information about the off-diagonal blocks of the matrix in the compression stage. In fact, it only needs to know the linear dependence relations between the columns (rows) in a neutered block column (row) (see Figure 1). According to the factorization (40), these linear dependence relations are the same for the (small) matrix $R$ as for the (large) matrix $\hat{T}_{\text{NC}}$, to within precision $\varepsilon$. In other words, during the inversion, we can compute $R$ in lieu of $\hat{T}_{\text{NC}}$ and use *it* to determine the linear dependence relations. Now, since the matrix $\tilde{R}$ in (45) can be precomputed, formula (50) shows that once a particular Toeplitz vector $t$ has been specified, it is possible to compute $R$ in $2nk$ floating point operations. We summarize these findings as follows.

OBSERVATION 1. Given a Toeplitz vector $t$, it is possible to compute the factor $R$ in (40) using $2nk$ floating point operations.

REMARK 3. PRECOMPUTATION. The statement of Observation 1 relies on the fact that $\tilde{R}$ has been precomputed for a given $N$ and for a given hierarchical partitioning of the index set $\{1, \ldots, N\}$. This can be accomplished in $O(N^2 \log^2 N)$ arithmetic operations using a standard rank-revealing QR-factorization as described in [14,15]. However, using the fact that $Y$ is circulant, it is possible to perform the precomputation in $O(N \log^2 N)$ operations since all neutered block columns of the same size are identical and since a large neutered block column can be factored by merging and updating the factorizations of two smaller ones.

## 5. NUMERICAL EXAMPLES

In this section, we present some results from numerical examples illustrating the performance of the fast inversion scheme of [11] when applied to four different types of Toeplitz matrices. These types were chosen to illustrate how the performance of the scheme, in terms of speed and accuracy, depends on,

(a) the condition number of the matrix $T$ to be inverted, and
(b) the smoothness of the Toeplitz vector $t$.

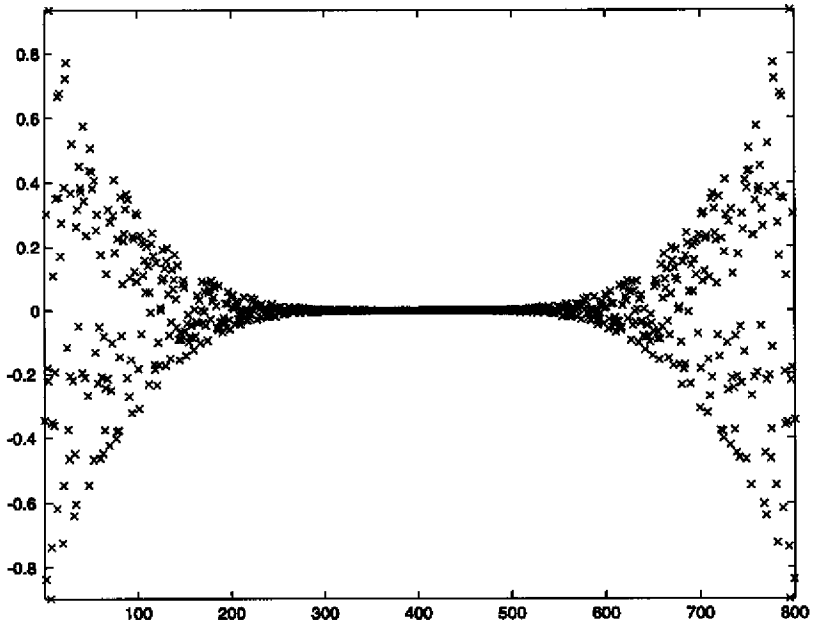


Figure 2. A nonsmooth Toeplitz vector used to generate an ill-conditioned Toeplitz matrix of "Type III".

The four types are as follows.

(1) *Nonsmooth t, well-conditioned T.* The Toeplitz vector consists of numbers randomly distributed between $-1$ and 1. Additionally, a constant $10\sqrt{N}$ was added to $t(N)$ to force the matrix to be well-conditioned.
(2) *Smooth t, well-conditioned T.* $t(j) = 1/(1 + |j - N|)$.
(3) *Non-smooth t, ill-conditioned T.* The Toeplitz vector was constructed from a sequence of randomly distributed numbers by suppressing those numbers close to the diagonal, see Figure 2.
(4) *Smooth t, ill-conditioned T.* $t(j) = 0.05/(1 + 0.02|j - N|^2)$.

For each of the four classes of matrices, the program was set to produce an approximate inverse with a residual error of at most $10^{-10}$. For the two well-conditioned classes of matrices, experiments were also performed at the lower accuracy $10^{-5}$.

The algorithm was implemented in Fortran 77 and run on a Pentium IV desktop with a 2.8 GHz processor and 512 Mb of RAM.

The results of these experiments are given in Tables 1–4. As a reference for the timings, Table 5 gives the computational times required to invert the matrices using Trench's algorithm (as described in [16]).

Table 1. Computational results for Toeplitz matrices of Type I. The top block reports experiments run at an accuracy of $10^{-10}$ and the lower block experiments run at $10^{-5}$.

| $N$ | $\kappa$ | $t_{\text{inv}}$ | $t_{\text{solve}}$ | $E_{\text{F}}$ | $M$ | $E_{\text{Trench}}$ |
|---|---|---|---|---|---|---|
| 401 | 1.3 | 4.2e − 2 | 1.0e − 3 | 1.0e − 9 | 3.6e − 1 | 8.4e − 15 |
| 801 | 1.3 | 9.2e − 2 | 3.0e − 3 | 3.3e − 9 | 7.1e − 1 | 1.2e − 14 |
| 1601 | 1.4 | 1.8e − 1 | 7.0e − 3 | 3.5e − 9 | 1.3 | 8.4e − 15 |
| 3201 | 1.4 | 3.5e − 1 | 1.4e − 2 | 4.0e − 9 | 2.6 | 1.2e − 14 |
| 6401 | 1.4 | 7.2e − 1 | 2.9e − 2 | 1.2e − 8 | 5.0 | 1.2e − 14 |
| 12801 | 1.4 | 1.4 | 6.0e − 2 | 1.9e − 8 | 1.0e − 1 | — |
| 401 | 1.3 | 1.6e − 2 | 5.0e − 4 | 8.8e − 5 | 1.7e − 1 | 8.4e − 15 |
| 801 | 1.3 | 3.0e − 2 | 1.2e − 3 | 1.5e − 4 | 3.2e − 1 | 1.2e − 14 |
| 1601 | 1.4 | 5.8e − 2 | 3.2e − 3 | 3.1e − 4 | 6.1e − 1 | 8.4e − 15 |
| 3201 | 1.4 | 1.2e − 1 | 47.2e − 3 | 7.6e − 4 | 1.2 | 1.2e − 14 |
| 6401 | 1.4 | 2.3e − 1 | 1.4e − 2 | 1.1e − 3 | 2.4 | 1.2e − 14 |
| 12801 | 1.4 | 4.6e − 1 | 3.1e − 2 | 2.9e − 3 | 4.9 | — |

Table 2. Computational results for Toeplitz matrices of Type II. The top block reports experiments run at an accuracy of $10^{-10}$ and the lower block experiments run at $10^{-5}$.

| $N$ | $\kappa$ | $t_{\text{inv}}$ | $t_{\text{solve}}$ | $E_{\text{F}}$ | $M$ | $E_{\text{Trench}}$ |
|---|---|---|---|---|---|---|
| 401 | 2.0 | 1.7e − 2 | 4.4e − 4 | 1.3e − 10 | 1.7e − 1 | 8.3e − 15 |
| 801 | 2.2 | 3.4e − 2 | 1.4e − 3 | 1.1e − 10 | 3.4e − 1 | 1.2e − 14 |
| 1601 | 2.3 | 6.5e − 2 | 3.4e − 3 | 2.4e − 10 | 6.5e − 1 | 8.0e − 15 |
| 3201 | 2.5 | 1.3e − 1 | 7.4e − 3 | 9.0e − 10 | 1.3 | 1.1e − 14 |
| 6401 | 2.6 | 2.6e − 1 | 1.5e − 2 | 6.8e − 10 | 2.6 | 1.1e − 14 |
| 12801 | 2.7 | 5.3e − 1 | 3.1e − 2 | 1.2e − 9 | 5.1 | — |
| 401 | 2.0 | 1.1e − 2 | 2.6e − 4 | 1.8e − 6 | 1.0e − 1 | 8.3e − 15 |
| 801 | 2.2 | 2.1e − 2 | 8.0e − 4 | 2.2e − 6 | 2.0e − 1 | 1.2e − 14 |
| 1601 | 2.3 | 3.6e − 2 | 2.2e − 3 | 3.6e − 6 | 3.9e − 1 | 8.0e − 15 |
| 3201 | 2.5 | 7.4e − 2 | 4.8e − 3 | 5.4e − 6 | 7.8e − 1 | 1.1e − 14 |
| 6401 | 2.6 | 1.5e − 1 | 9.6e − 3 | 8.7e − 6 | 1.5 | 1.1e − 14 |
| 12801 | 2.7 | 3.0e − 1 | 2.1e − 2 | 1.7e − 5 | 3.2 | — |

Table 3. Computational results for Toeplitz matrices of Type III.

| $N$ | $\kappa$ | $t_{\text{inv}}$ | $t_{\text{solve}}$ | $E_{\text{F}}$ | $M$ | $E_{\text{Trench}}$ |
|---|---|---|---|---|---|---|
| 401 | 1.4e4 | 5.0e − 2 | 1.4e − 3 | 1.8e − 7 | 4.0e − 1 | 3.7e − 10 |
| 801 | 2.1e4 | 1.1e − 1 | 3.4e − 3 | 2.7e − 7 | 7.7e − 1 | 9.9e − 9 |
| 1601 | 2.8e5 | 2.2e − 1 | 7.4e − 3 | 6.8e − 6 | 1.5 | 3.2e − 7 |
| 3201 | 7.5e5 | 4.4e − 1 | 1.7e − 2 | 3.0e − 5 | 2.9 | 3.9e − 6 |
| 6401 | 3.1e6 | 8.8e − 1 | 3.3e − 2 | 8.2e − 4 | 5.7 | 3.1e − 5 |
| 12801 | 5.0e6 | 1.6 | 6.5e − 2 | 7.5e − 3 | 11 | — |

Table 4. Computational results for Toeplitz matrices of Type IV.

| $N$ | $\kappa$ | $t_{\text{inv}}$ | $t_{\text{solve}}$ | $E_{\text{F}}$ | $M$ | $E_{\text{Trench}}$ |
|---|---|---|---|---|---|---|
| 401 | 2.1e9 | 1.7e − 2 | 4.0e − 4 | 2.1e − 3 | 1.7e − 1 | 1.9e − 6 |
| 801 | 2.2e9 | 3.3e − 2 | 1.4e − 3 | 1.8e − 3 | 3.4e − 1 | 6.0e − 6 |
| 1601 | 2.2e9 | 6.5e − 2 | 3.4e − 3 | 8.1e − 3 | 6.5e − 1 | 7.4e − 6 |
| 3201 | 2.2e9 | 1.3e − 1 | 7.6e − 3 | 2.0e − 2 | 1.3 | 7.8e − 6 |
| 6401 | 2.2e9 | 2.6e − 1 | 1.6e − 2 | 1.1e − 2 | 2.6 | 7.9e − 6 |
| 12801 | 2.2e9 | 5.1e − 1 | 3.2e − 2 | 1.4e − 2 | 5.1 | — |

Table 5. Performance numbers for the Trench algorithm.

| $N$ | $t_{\text{inv}}$ | $t_{\text{apply}}$ | $M$ |
|---|---|---|---|
| 401 | 1.0e − 3 | 7.5e − 4 | 3.1e − 1 |
| 801 | 5.0e − 3 | 3.8e − 3 | 1.2 |
| 1601 | 2.4e − 2 | 1.7e − 2 | 4.9 |
| 3201 | 1.0e − 1 | 6.5e − 2 | 2.0e1 |
| 6401 | 4.4e − 1 | 2.6e − 1 | 7.8e1 |

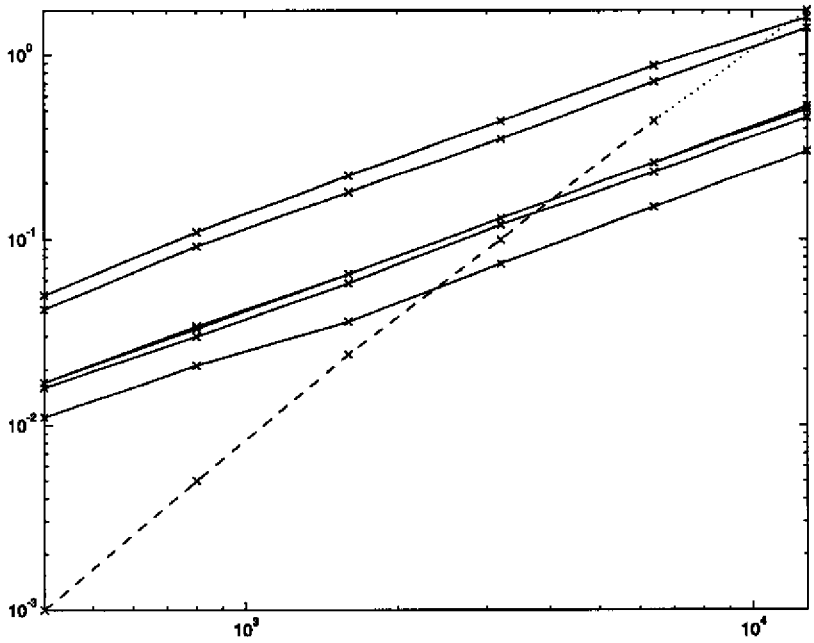


Figure 3. Time in seconds for inverting $T$ versus problem size. The timings for the compressed algorithm applied to different types of Toeplitz vectors are drawn with solid lines while the time required for the Trench algorithm is drawn with a dashed line (the last data point is extrapolated).

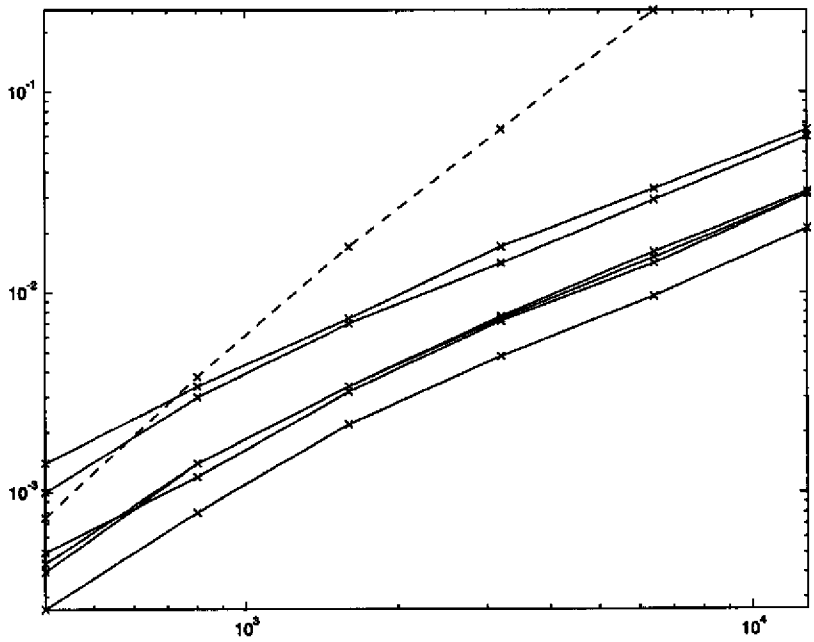


Figure 4. Time in seconds for applying the inverse of $T$ versus problem size. The timings for the compressed algorithm applied to different types of Toeplitz vectors are drawn with solid lines while the time required for the Trench algorithm is drawn with a dashed line.

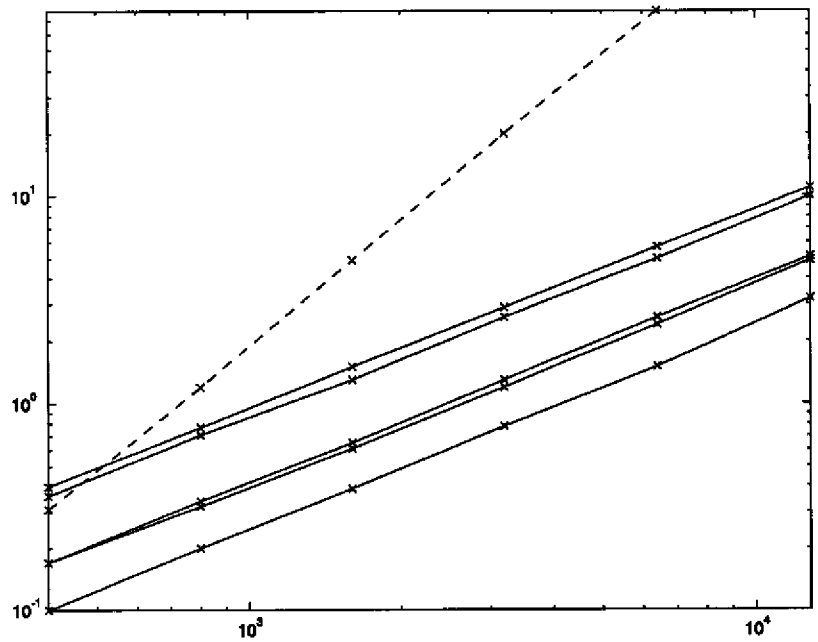


Figure 5. The amount of memory required (in Mb) for inverting $T$ versus problem size. Solid lines corresponds to accelerated algorithms while the dashed line corresponds to the Trench algorithm.

The computational error is reported in the $l^2$-operator norm; to be precise, if we denote the computed inverse by $T_\varepsilon^{-1}$, what we report is

$$E_{\mathrm{F}} = \max_{f \in \mathbb{R}^N} \frac{\left\| T_\varepsilon^{-1} f - T^{-1} f \right\|}{\left\| T^{-1} f \right\|} = \max_{u \in \mathbb{R}^N} \frac{\left\| T_\varepsilon^{-1} T u - u \right\|}{\left\| u \right\|} = \left\| T_\varepsilon^{-1} T - I \right\|. \tag{52}$$

This quantity was computed by applying a power iteration to the operator $(T_\varepsilon^{-1}T-I)^*(T_\varepsilon^{-1}T-I)$ (we note that $T_\varepsilon^{-1}T$ is typically not entirely symmetric). Similarly, letting $T_{\text{Trench}}^{-1}$ denote the inverse computed using the Trench algorithm, we report $E_{\text{Trench}} = \|T_{\text{Trench}}^{-1}T - I\|$ for each of the matrices investigated. In the tables, the following numbers are also reported.

| | |
|---|---|
| $\kappa$ | The condition number of the Toeplitz matrix $T$. |
| $t_{\text{inv}}$ | Time required to compute the compressed inverse (in seconds). |
| $t_{\text{solve}}$ | Time required to apply the compressed inverse (in seconds). |
| $M$ | Memory required to compute the compressed inverse (in Mb). |

The data given in the tables is presented graphically in Figures 3–5, in which, respectively, $t_{\text{inv}}$, $t_{\text{solve}}$, and $M$, are plotted against $N$.

The numerical experiments show that both the computational complexity and the memory requirements of the fast inversion scheme presented here scale more or less linearly with problem size. Moreover, both of these quantities depend on the structure of the matrix to be inverted; when the Toeplitz vector $t$ is smooth, the algorithm requires only 35% as many arithmetic operations as in the worst case.

The size of a matrix at which the procedure of this paper compares favorably with the Trench scheme depends on what is being compared, what accuracy is required, and the properties of the Toeplitz vector. Thus, if the times for inverting the matrix are compared, the break-even point in our experiments lies somewhere between 2000 and 10000; if the time to apply the inverse to a vector is compared, the break-even point is 800 or less; and if the memory requirements are compared, the break-even point is 500 or less.

## REFERENCES


1. W.R. Trench, An algorithm for the inversion of finite Toeplitz matrices, *J. Soc. Indust. Appl. Math.* **12**, 515–522, (1964).
2. S. Zohar, Toeplitz matrix inversion: The algorithm of W.F. Trench, *J. Assoc. Comput. Mach.* **16**, 592–601, (1969).
3. G. Ammar, Classical foundations of algorithms for solving positive definite Toeplitz equations, *Calcolo* **33**, 99–113, (1996).
4. M. Van Barel, G. Heinig and P. Kravanja, A stabilized superfast solver for nonsymmetric Toeplitz systems, *SIAM Journal on Matrix Analysis and Applications* **23** (2), 494–510, (2001).
5. M. Stewart, A superfast Toeplitz solver with improved numerical stability, *SIAM Journal on Matrix Analysis and Applications* **25** (3), 669–693, (2003).
6. J.-P. Cardinal, On a property of Cauchy-like matrices, *C. R. Acad. Sci. Paris Sér. I Math.* **328** (11), 1089–1093, (1999).
7. S. Chandrasekaran, T. Pals, M. Gu, P. Dewilde and A.-J. van der Veen, Fast and stable direct solvers for PDEs and integral equations, In *Colloquium Talk, Summer, Lawrence Livermore National Laboratory*, (2002).
8. T. Kailath and V. Olshevsky, Diagonal pivoting for partially reconstructible Cauchy-like matrices, with applications to Toeplitz-like linear equations and to boundary rational matrix interpolation problems, in *Proceedings of the Fifth Conference of the International Linear Algebra Society* (Atlanta, GA, 1995), Special Issue of *Linear Algebra Appl.* **254**, 251–302, (1997).
9. V. Olshevksy, Editor, Fast algorithms for structured matrices: Theory and applications, In *Contemporary Mathematics, Volume 323*, Papers from the AMS-IMS-SIAM Joint Summer Research Conference on Fast Algorithms in Mathematics, Computer Science and Engineering held at Mount Holyoke College, South Hadley, MA, August 5–9, 2001 pp. viii–433, American Mathematical Society, Providence, RI, U.S.A., (2003).
10. V.Y. Pan and A. Zheng, Superfast algorithms for Cauchy-like matrix computations and extensions, *Linear Algebra Appl.* **310** (1-3), 83–108, (2000).
11. P.G. Martinsson and V. Rokhlin, A fast direct solver for boundary integral equations in two dimensions, *J. Comp. Phys.* **205** (1), 1–23, (2005).
12. S. Chandrasekaran and M. Gu, A fast and stable solver for recursively semi-separable systems of equations, In *Structured Matrices in Mathematics, Computer Science, and Engineering, Contemporary Mathematics, Volume II*, Proceedings of an AMS-IMS-SIAM Joint Summer Research Conference, University of Colorado, Boulder, June 27–July 1, 1999, (Edited by V. Olshevsky),, AMS Publications, Providence, RI, U.S.A., (2001).
13. A. Dutt and V. Rokhlin, Fast Fourier transforms for nonequispaced data. II, *Appl. Comput. Harmon. Anal.* **2** (1), 85–100, (1995).
14. M. Gu and S.C. Eisenstat, Efficient algorithms for computing a strong rank-revealing QR factorization, *SIAM J. Sci. Comput.* **17** (4), 848–869, (1996).

15. H. Cheng, Z. Gimbutas, P.G. Martinsson and V. Rokhlin, On the compression of low-rank matrices, *SIAM J. Sci. Comp.* **26**, 1389–1404, (2005).
16. G.H. Golub and C.F. Van Loan, *Matrix Computations, Johns Hopkins Studies in the Mathematical Sciences*, Third Edition, Johns Hopkins University Press, Baltimore, MD, U.S.A., (1996).