Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Linear Algebra and its Applications 366 (2003) 441–457
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

A superfast method for solving Toeplitz linear least squares problems

Marc Van Barel [a,*], Georg Heinig [b], Peter Kravanja [a]

[a]*Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium*
[b]*Department of Mathematics and Computer Science, Kuwait University, P.O. Box 5969, Safat 13060, Kuwait*

Received 16 May 2001; accepted 10 June 2002

Submitted by D.A. Bini

Abstract

In this paper we develop a superfast $\mathcal{O}((m+n)\log^2(m+n))$ complexity algorithm to solve a linear least squares problem with an $m \times n$ Toeplitz coefficient matrix. The algorithm is based on the augmented matrix approach. The augmented matrix is further extended to a block circulant matrix and DFT is applied. This leads to an equivalent tangential interpolation problem where the nodes are roots of unity. This interpolation problem can be solved by a divide and conquer strategy in a superfast way. To avoid breakdowns and to stabilize the algorithm pivoting is used and a technique is applied that selects "difficult" points and treats them separately. The effectiveness of the approach is demonstrated by several numerical examples.

Keywords: Least squares problem; Toeplitz matrices; Structured matrices; Superfast algorithm; Block circulant matrices; Vector polynomial interpolation; Divide and conquer strategy

1. Introduction

Let $T = [\, t_{i-j}\,] \in \mathbb{C}^{m\times n}$ be an $m \times n$ Toeplitz matrix with $m > n$ and full column rank n, and let $b \in \mathbb{C}^m$. We consider the least squares problem: given T and

* Corresponding author.
E-mail addresses: marc.vanbarel@cs.kuleuven.ac.be, (M. Van Barel), georg@mcs.sci.kuniv.edu.kw (G. Heinig), peter.kravanja@na-net.ornl.gov (P. Kravanja).

0024-3795/03/$ - see front matter
doi:10.1016/S0024-3795(02)00495-0

b, determine the (unique) vector $x \in \mathbb{C}^n$ such that $\|Tx - b\|$ is minimal. Here $\|\cdot\|$ denotes the (Euclidian) 2-norm.

Standard algorithms for solving linear least squares problems require $\mathcal{O}(mn^2)$ flops. The arithmetic complexity can be reduced by taking into account the Toeplitz structure. Algorithms that require only $\mathcal{O}(mn)$ flops are called *fast*, algorithms that require less operations are called *superfast*. As far as we know, the first fast algorithm for Toeplitz least squares problems was proposed by Sweet in his Ph.D. thesis [25]. Other approaches include those by Bojanczyk et al. [5], Chun et al. [9], Qiao [24], Cybenko [10,11], Sweet [26] and many others.

The aim of the present paper is to propose a superfast algorithm with a complexity of $\mathcal{O}((m+n)\log^2(m+n))$. For the solution of Toeplitz systems algorithms with this complexity are well known (see [1–4,6,12,30] and other papers).

The problem with many fast and superfast algorithms is that their numerical performance heavily depends on the condition of certain submatrices. They even may breakdown if some of these submatrices are singular. One of several ways out of such situations is to transform the Toeplitz matrix into a Cauchy-like matrix with the help of the discrete Fourier or other trigonometric transformations. Cauchy-like matrices have the advantage that, in contrast to Toeplitz matrices, permutations of rows and columns do not destroy the structure. That means that the application of pivoting strategies is possible in order to stabilize the algorithm. For the solution of Toeplitz systems this approach was first proposed in [17] and further developed in [13–15,18–22,27] and other papers. Let us point out that “stabilize” does not mean that the resulting algorithm is stable in a strong sense, but only that it is stable “in practice”, like it is true for Gaussian elimination for unstructured systems with partial pivoting.

In Gu’s paper [16] the transformation idea was applied to the solution of Toeplitz least squares problems. Numerical experiments indicate that Gu’s approach is not only efficient but also fairly stable for many examples, even for ill-conditioned problems.

The transformation of Toeplitz into Cauchy-like has the disadvantage that there seems to be no freedom in the choice of the transformation length. Differently, such a freedom does exist if the matrix is transformed into a Vandermonde-like matrix, which means, in the language of matrix functions, into a tangential interpolation problem. This approach was proposed in [18,23,30]. In the latter paper a superfast Toeplitz solver was proposed. The main aim of the present paper is to apply some modification of the ideas in [30]. Also the same stabilizing techniques are used to enhance the numerical properties of the algorithm. The experiments described in Section 6 show that these stabilizing techniques lead to accurate results.

Our approach can be briefly described as follows. It is based on the augmented matrix idea that relates the $m \times n$ least squares problem with an equivalent $(m+n) \times (m+n)$ linear system. If the coefficient matrix T is Toeplitz then the augmented matrix R will be a Toeplitz block matrix with T as one of its blocks. This will be explained in Section 2.

In Section 3 the system for R will be extended to a circulant block system. This system will be transformed using the discrete Fourier transform into a Vandermonde block system. The latter one will be interpreted, in Section 4, as a homogeneous tangential interpolation problem, so that standard solution procedures for this kind of problems (see [7, Chapter 7]) can be applied. The inclusion of a divide-and-conquer principle as in [30] speeds up the algorithm from $\mathrm{O}(mn)$ complexity to complexity $\mathrm{O}((m+n)\log^2(m+n))$. The basic features of the superfast solver of tangential interpolation problems are explained in Section 5. We refrain from presenting the algorithm in all its details as these are given in [30].

Let us note, that in order to achieve a complexity less than $\mathcal{O}(mn)$ it is not possible to apply full partial pivoting, since already this part of the algorithm would need $\mathcal{O}(mn)$ operations. Instead of this we only select out "difficult" interpolation points during the algorithm and treat them in the standard fast (i.e. not superfast) way at the end. This avoids breakdowns of the algorithm guaranteeing a certain degree of stability in the process. Iterative refinements at intermediate steps and at the end improve the results. In Section 6 we discuss the numerical performance of the algorithm. It turns out that our algorithm gives good results for well-conditioned and mildly ill-conditioned matrices. For highly ill-conditioned matrices the results are not satisfactory.

Let us finally mention the paper of Chandrasekaran and Sayed [8]. In this paper, a fast algorithm is designed to solve structured linear systems, in particular Toeplitz systems. The approach is based on matrix augmentation and the application of the Schur algorithm. The authors also prove the backward stability of their algorithm. As they suggest in the conclusions of their paper, this approach can be, in principle, also be applied to Toeplitz least squares problems. It would be interesting to have also a superfast version of the proposed algorithm and to compare the numerical performance with our algorithm.

2. Extension approach

We explain in this section the familiar extension approach for solving least squares problems. Let A be any $m \times n$ matrix with $m > n$ and full column rank n. It is well known that the solution x of the least squares problem to minimize the norm $\|r\|$ of the residual $r = b - Ax$ can be characterized by the condition $A^H r = 0$. Introducing the $(m+n) \times (m+n)$ matrix

$$R = \begin{bmatrix} I_m & A \\ A^H & 0 \end{bmatrix}$$

the least squares problem is equivalent to the linear system

$$R \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

We describe the relation between A and R more completely. Note that the solution of the least squares problem can be written in the form $x = (A^H A)^{-1} A^H b$.

Lemma 2.1. *If A has full column rank, then R is nonsingular and*

$$R^{-1} = \begin{bmatrix} P & (A^\dagger)^H \\ A^\dagger & -(A^H A)^{-1} \end{bmatrix},$$

where P denotes the orthoprojection onto the kernel of A^H.

Proof. Let $\begin{bmatrix} r \\ x \end{bmatrix}$ be a solution of the linear system

$$R \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ a \end{bmatrix}.$$

Then $r + Ax = b$, which implies $A^H Ax = A^H b - a$. Since A has full column rank, $A^H A$ is nonsingular and we obtain

$$x = (A^H A)^{-1} A^H b - (A^H A)^{-1} a = A^\dagger b - (A^H A)^{-1} a. \tag{1}$$

Furthermore,

$$r = (I_m - AA^\dagger) b + A(A^H A)^{-1} a = Pb + (A^\dagger)^H a. \tag{2}$$

The relations (1) and (2) provide the formula for R^{-1}. $\square$

Corollary 2.1. *The solution of the least squares problem to minimize $\|Ax - b\|$ is the second block component x of the solution of the system*

$$R \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \tag{3}$$

For the least squares problem we will always have a right-hand side with a zero second block component. But we will need the system for a general right-hand side if we want to apply iterative refinement. Let us explain this.

Let us assume that an approximate solution

$$\begin{bmatrix} \tilde{r} \\ \tilde{x} \end{bmatrix} \tag{4}$$

is at our disposal, with corresponding residuals Δa and Δb:

$$\begin{bmatrix} \Delta b \\ \Delta a \end{bmatrix} := \begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I_m & A \\ A^H & 0 \end{bmatrix} \begin{bmatrix} \tilde{r} \\ \tilde{x} \end{bmatrix}.$$

Iterative refinement of (4) is based on the following fact: if Δr and Δx are such that

$$\begin{bmatrix} I_m & A \\ A^H & 0 \end{bmatrix} \begin{bmatrix} \Delta r \\ \Delta x \end{bmatrix} = \begin{bmatrix} \Delta b \\ \Delta a \end{bmatrix}, \tag{5}$$

then

$$\begin{bmatrix} I_m & A \\ A^H & 0 \end{bmatrix} \begin{bmatrix} \tilde{r} + \Delta r \\ \tilde{x} + \Delta x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Theoretically, it is therefore possible to move from $[\tilde{r} \ \ \tilde{x}]^{\mathrm{T}}$ to $[r \ \ x]^{\mathrm{T}}$ in a single step. In numerical calculations (5) will only be solved approximately, of course, and hence the process will have to be repeated. The linear systems (3) and (5) differ only in their right-hand sides.

Thus, to cover also iterative refinement, we will henceforth consider the linear system

$$R \begin{bmatrix} \rho \\ \xi \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix}, \tag{6}$$

where initially $\alpha = 0$, $\beta = b$ and $\rho = r$, $\xi = x$.

Another way to do iterative refinement is to use a formula for the inverse of R, as it was done in [30] for the solution of Toeplitz systems. This will be dicussed in a forthcoming paper.

3. Transformation into a block circulant system

Henceforth we consider Toeplitz matrices. It is clear that any $m \times n$ Toeplitz matrix

$$S = \left[s_{j-k} \right]_{j=0,1,\ldots,m-1}^{k=0,1,\ldots,n-1}$$

can be extended at its top with a matrix $\tilde{S}$ such that the extended $M \times n$ matrix

$$C := \begin{bmatrix} \tilde{S} \\ S \end{bmatrix}$$

is just the left $M \times n$ submatrix of an $M \times M$ circulant. Here M is any integer satisfying $M \geqslant m + n - 1$. If $M = m + n - 1$, then we have to choose $\tilde{S}$ as the $(n-1) \times n$ Toeplitz matrix

$$\tilde{S} := \left[s_{-n+1+j-k} \right]_{j=0,1,\ldots,n-2}^{k=0,1,\ldots,n-1},$$

where $s_{-n-k} := s_{m-k-1}$ for $k = 0, 1, \ldots, n-1$. It might be convenient to choose M larger than $m + n - 1$ in order to make sure that the discrete Fourier transform of the columns of C can be computed efficiently.

We extend the Toeplitz matrices T and T^H into the circulant matrices

$$\begin{bmatrix} \bar{T} \\ T \end{bmatrix} \in \mathbb{C}^{M \times n} \quad \text{and} \quad \begin{bmatrix} \tilde{T} \\ T^H \end{bmatrix} \in \mathbb{C}^{M \times m}$$

respectively, where $\bar{T} \in \mathbb{C}^{(M-m) \times n}$ and $\tilde{T} \in \mathbb{C}^{(M-n) \times m}$. Define

$$\sigma := \tilde{T} \rho \in \mathbb{C}^{M-n} \quad \text{and} \quad \zeta := \bar{T} \xi \in \mathbb{C}^{M-m}.$$

With these definitions and notations we can extend (6) into the following homogeneous linear system of equations:

$$\begin{bmatrix} 0 & 0 & 0 & \bar{T} & -I_{M-m} \\ I_m & -\beta & 0 & T & 0 \\ \tilde{T} & 0 & -I_{M-n} & 0 & 0 \\ T^H & -\alpha & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ 1 \\ \sigma \\ \xi \\ \zeta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

This system can be written in block form as

$$\begin{bmatrix} C_1 & C_2 & 0 & C_3 & C_4 \\ C_5 & C_6 & C_7 & 0 & 0 \end{bmatrix} \begin{bmatrix} \rho \\ 1 \\ \sigma \\ \xi \\ \zeta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Note that the matrices $C_1, \ldots, C_7$ all have row size M and correspond to the first columns of circulant matrices.

A $M \times M$ circulant matrix C can be factorized as

$$C = \mathscr{F}_M^H \Lambda \mathscr{F}_M,$$

where Λ is a $M \times M$ diagonal matrix containing the eigenvalues of C and $\mathscr{F}_M$ denotes the $M \times M$ discrete Fourier transform matrix

$$\mathscr{F}_M := \left[\omega_M^{jk}\right]_{j,k=0,1,\ldots,M-1},$$

where $\omega_M := \mathrm{e}^{-2\pi \mathrm{i}/M}$ and i denotes the imaginary unit. Similarly, if C is of size $M \times q$, where $q \leqslant M$, then C can be factorized as

$$C = \mathscr{F}_M^H \Lambda \mathscr{F}_{M,q},$$

where Λ is again a $M \times M$ diagonal matrix and $\mathscr{F}_{M,q}$ denotes the $M \times q$ submatrix of $\mathscr{F}_M$ containing the first q columns of $\mathscr{F}_M$. It follows that

$$\begin{aligned} &\begin{bmatrix} \mathscr{F}_M & 0 \\ 0 & \mathscr{F}_M \end{bmatrix} \begin{bmatrix} C_1 & C_2 & 0 & C_3 & C_4 \\ C_5 & C_6 & C_7 & 0 & 0 \end{bmatrix} \\ &\quad = \begin{bmatrix} \Lambda_1 \mathscr{F}_{M,m} & \Lambda_2 \mathscr{F}_{M,1} & 0 & \Lambda_3 \mathscr{F}_{M,n} & \Lambda_4 \mathscr{F}_{M,M-m} \\ \Lambda_5 \mathscr{F}_{M,m} & \Lambda_6 \mathscr{F}_{M,1} & \Lambda_7 \mathscr{F}_{M,M-n} & 0 & 0 \end{bmatrix}, \end{aligned}$$

where $\Lambda_j =: \mathrm{diag}(\lambda_1^{(j)}, \ldots, \lambda_M^{(j)})$ is a $M \times M$ diagonal matrix for $j = 1, \ldots, 7$.

4. Interpolation interpretation

Let us now translate the homogeneous linear system into polynomial language. The discrete Fourier transform matrix $\mathscr{F}_M$ can be interpreted as the Vandermonde matrix based on the nodes $z_k := \omega_M^{k-1}$, $k = 1, \ldots, M$. Multiplying a vector with (leading columns of) $\mathscr{F}_M$ hence corresponds to evaluating a polynomial at the Mth roots of unity $z_1, \ldots, z_M$.

Let $\rho =: [\rho_k]_{k=0}^{m-1}$ and define the polynomial $\rho(z)$ as

$$\rho(z) := \sum_{k=0}^{m-1} \rho_k z^k.$$

The polynomials $\sigma(z)$, $\xi(z)$ and $\zeta(z)$ are defined in a similar way.

The previous considerations now enable us to reformulate the original Toeplitz linear least squares problem as the following equivalent interpolation problem: determine the polynomials $\xi(z)$, $\zeta(z)$, $\rho(z)$ and $\sigma(z)$, where $\deg \xi(z) \leqslant n-1$, $\deg \zeta(z) \leqslant M-m-1$, $\deg \rho(z) \leqslant m-1$ and $\deg \sigma(z) \leqslant M-n-1$, such that

$$\lambda_k^{(1)} \rho(z_k) + \lambda_k^{(2)} 1 + \lambda_k^{(3)} \xi(z_k) + \lambda_k^{(4)} \zeta(z_k) = 0$$

and

$$\lambda_k^{(5)} \rho(z_k) + \lambda_k^{(6)} 1 + \lambda_k^{(7)} \sigma(z_k) = 0$$

for $k = 1, \ldots, M$. In other words, determine the vector polynomial $p(z) \in \mathbb{C}[z]^{5\times 1}$ of (componentwise) degree

$$\deg p(z) < \begin{bmatrix} m \\ 1 \\ M-n \\ n \\ M-m \end{bmatrix}$$

such that

$$\begin{bmatrix} G_k \\ F_k \end{bmatrix} p(z_k) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad k = 1, \ldots, M, \tag{7}$$

where

$$F_k := \begin{bmatrix} \lambda_k^{(1)} & \lambda_k^{(2)} & 0 & \lambda_k^{(3)} & \lambda_k^{(4)} \end{bmatrix}$$

and

$$G_k := \begin{bmatrix} \lambda_k^{(5)} & \lambda_k^{(6)} & \lambda_k^{(7)} & 0 & 0 \end{bmatrix}$$

for $k = 1, \ldots, M$.

5. A superfast algorithm for solving polynomial interpolation problems

In this section we will show how the interpolation problem (7) can be solved in a superfast way. Our algorithm is of divide and conquer type and involves “stabilizing” techniques to enhance the computational accuracy. To obtain the algorithm, we rely on the theoretical framework of basis matrices and τ-degree. We therefore start by recalling these concepts, adapting the definitions to our specific case.

Let $\mathscr{S}$ be the set of all the column vector polynomials $p(z) \in \mathbb{C}[z]^{5\times 1}$ that satisfy the interpolation conditions (7). If $p(z) \in \mathbb{C}[z]^{5\times 1}$ is an arbitrary vector polynomial, then the left-hand side of (7) is called the *residual* with respect to $p(z)$ at the interpolation point z_k.

The set $\mathscr{S}$ forms a submodule of the $\mathbb{C}[z]$-module $\mathbb{C}[z]^{5\times 1}$. A basis for $\mathscr{S}$ always consists of exactly five elements [28, Theorem 3.1]. Let $\{B_1(z), B_2(z), \ldots, B_5(z)\}$ be a basis for $\mathscr{S}$. Then every element $p(z) \in \mathscr{S}$ can be written in a unique way as $p(z) = \sum_{i=1}^{5} \alpha_i(z)B_i(z)$ with $\alpha_i(z) \in \mathbb{C}[z]$, $i = 1, 2, \ldots, 5$. The matrix polynomial $B(z) := [B_1(z)\ B_2(z)\ \cdots\ B_5(z)] \in \mathbb{C}[z]^{5\times 5}$ is called a *basis matrix*. Basis matrices can be characterized as follows.

Theorem 5.1. *A matrix polynomial $C(z) = [C_1(z)\ C_2(z)\ \cdots\ C_5(z)] \in \mathbb{C}[z]^{5\times 5}$ is a basis matrix if and only if $C_i(z) \in \mathscr{S}$ for $i = 1, 2, \ldots, 5$ and* $\deg\det C(z) = 2M$.

Proof. This follows immediately from [28, Theorem 4.1]. $\square$

Within the submodule $\mathscr{S}$ we want to be able to consider solutions $p(z)$ that satisfy additional conditions concerning their degree-structure. To describe the degree-structure of a vector polynomial, we use the concept of τ-degree [28]. Let $\tau \in \mathbb{Z}^5$. The τ-degree of a vector polynomial $p(z) = [p_1(z)\ p_2(z)\ \cdots\ p_5(z)]^{\mathrm{T}} \in \mathbb{C}[z]^{5\times 1}$ is defined as a generalization of the classical degree:

$$\tau\text{-deg}\, w(z) := \max_i \{\deg p_i(z) - \tau_i\}$$

with τ-deg $0 := -\infty$. The τ-highest degree coefficient of a vector polynomial

$$[p_1(z)\ p_2(z)\ \cdots\ p_5(z)]^{\mathrm{T}}$$

with τ-degree δ is defined as the vector $[\omega_1\ \omega_2\ \cdots\ \omega_5]^{\mathrm{T}}$ with ω_i the coefficient of $z^{\delta+\tau_i}$ in $p_i(z)$. A set of vector polynomials in $\mathbb{C}[z]^{5\times 1}$ is called *τ-reduced* if the τ-highest degree coefficients are linearly independent. Every basis of $\mathscr{S}$ can be transformed into a τ-reduced one. For details, we refer to [28]. Once we have a basis in τ-reduced form, the elements of $\mathscr{S}$ can be parametrized as follows.

Theorem 5.2. *Let $\{B_1(z), B_2(z), \ldots, B_5(z)\}$ be a τ-reduced basis for $\mathscr{S}$. Define $\delta_i := \tau\text{-deg}\, B_i(z)$ for $i = 1, 2, \ldots, 5$. Then every element $p(z) \in \mathscr{S}$ having τ-degree $\leqslant \delta$ can be written in a unique way as*

$$p(z) = \sum_{i=1}^{5} \alpha_i(z)B_i(z)$$

with $\alpha_i(z) \in \mathbb{C}[z]$, $\deg\alpha_i(z) \leqslant \delta - \delta_i$.

Proof. See Van Barel and Bultheel [29,Theorem 3.2]. □

We will solve the interpolation problem (7) by constructing a 5×5 basis matrix $B_{FG}(z)$ such that

$$\begin{bmatrix} G_k \\ F_k \end{bmatrix} B_{FG}(z_k) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad k = 1, \dots, M.$$

The matrix polynomial $B_{FG}(z)$ is τ-reduced with $\tau = (m, 1, M-n, n, M-m)$. Then this basis matrix will have just one column having τ-degree equal to -1. By normalizing the second component of this vector polynomial to one, we derive the solution of the interpolation problem (7).

How to compute a τ-reduced basis matrix? The following theorem immediately leads to a (recursive) divide and conquer approach (and hence, a superfast algorithm). The theorem shows how basis matrices can be coupled in case the degree-structure is important.

The idea behind the theorem is to split up a "big" interpolation problem of size K into two "smaller" problems of size κ and $K-\kappa$, respectively. Given a τ-reduced basis matrix for the problem of size κ, the theorem shows how the interpolation data for the remaining problem of size $K-\kappa$ needs to be modified, such that the τ-reduced basis matrix for the problem of (full) size K is obtained simply by multiplying (which can be done via FFT as the product involves polynomial matrices) the τ-reduced basis matrix for the problem of size κ and the τ-reduced basis matrix for the problem of size $K-\kappa$. How the theorem then leads to a divide and conquer algorithm is self-evident.

Theorem 5.3. *Suppose K is a positive integer. Let $\sigma_1, \dots, \sigma_K \in \mathbb{C}$ be mutually distinct and let $\phi_1, \dots, \phi_K \in \mathbb{C}^{5\times 1}$. Suppose that $\phi_k \neq [0\ 0\ \cdots\ 0]^{\mathrm{T}}$ for $k = 1, \dots, K$. Let $1 \leqslant \kappa \leqslant K$. Let $\tau_K \in \mathbb{Z}^5$. Suppose that $B_\kappa(z) \in \mathbb{C}[z]^{5\times 5}$ is a τ_K-reduced basis matrix with basis vectors having τ_K-degree δ_i for $i = 1, 2, \dots, 5$, corresponding to the interpolation data*

$$\{(\sigma_k, \phi_k) : k = 1, \dots, \kappa\}.$$

Let $\tau_{\kappa\to K} := -[\delta_1, \delta_2, \dots, \delta_5]$. Let $B_{\kappa\to K}(z) \in \mathbb{C}[z]^{5\times 5}$ be a $\tau_{\kappa\to K}$-reduced basis matrix corresponding to the interpolation data

$$\left\{(\sigma_k, B_\kappa^{\mathrm{T}}(\sigma_k)\phi_k) : k = \kappa+1, \dots, K\right\}.$$

Then $B_K(z) := B_\kappa(z)B_{\kappa\to K}(z)$ is a τ_K-reduced basis matrix corresponding to the interpolation data

$$\{(\sigma_k, \phi_k) : k = 1, \dots, K\}.$$

Proof. See Van Barel and Bultheel [29, Theorem 3]. □

In [30], we developed a superfast algorithm based on this theorem for vector polynomials having two instead of five components. To enhance the numerical stability pivoting is used and if it turns out that updating the basis matrix to satisfy an additional interpolation condition would lead to an ill-conditioned subproblem, these interpolation conditions are skipped and only handled at the very end of the algorithm. The corresponding interpolation points are called "difficult points". For further details we refer the reader to [30]. Note that in [30] the basis matrix is of size 2×2 whereas in the present paper it is of size 5×5.

To obtain $B_{FG}(z)$ we proceed as follows. First we apply our superfast interpolation algorithm to the data (z_k, F_k), $k = 1, \ldots, M$, to obtain a 5×5 basis matrix $B_F(z)$ and a (possibly empty) set of difficult points z_j, $j \in \mathscr{D}_F$. Next, we apply this algorithm to the data $(z_k, G_k B_F(z_k))$, $k = 1, \ldots, M$, to obtain a 5×5 basis matrix $B_{F\to FG}(z)$ and another (again possibly empty) set of difficult points z_j, $j \in \mathscr{D}_{F\to FG}$. We get $B'_{FG} := B_F(z)B_{F\to FG}(z)$ via FFT and by applying the fast interpolation algorithm to $B'_{FG}(z)$ we add the difficult points z_j, $j \in \mathscr{D}_F \cup \mathscr{D}_{F\to FG}$ to obtain $B_{FG}(z)$.

The presence of zeros in F_k can be exploited to obtain the 5×5 basis matrix $B_{FG}(z)$ as the product of the basis matrices $B_F(z)$ and $B_{F\to FG}(z)$ where $B_F(z)$ can be derived from a 4×4 basis matrix. If we handle first the interpolation conditions connected to G_k and then those corresponding to F_k we can compute the final 5×5 basis matrix as the product of $B_G(z)$ and $B_{G\to FG}(z)$ where $B_G(z)$ can be obtained from a basis matrix of size 3×3.

6. Numerical examples

We have implemented our algorithm in Matlab (version 5.3.0.10183 (R11) on LNX86). In the following numerical experiments the computations have been performed in double precision arithmetic with unit roundoff $u \approx 1.11 \times 10^{-16}$.

6.1. Computational complexity

Let us start by comparing the computational complexity of our approach (algorithm NEW) with that of the algorithm based on the QR factorization (algorithm QR), the classical algorithm for solving general dense linear least squares problems. We consider Toeplitz matrices whose entries are chosen uniformly at random in the interval (0,1). The number of interpolation points $M = 2^k$ for $k = 5, \ldots, 14$. We choose $m = M/2$ and $n = M/4$. Fig. 1 plots the number of flops required by NEW and by QR, respectively. It follows that NEW is indeed superfast (as long as the

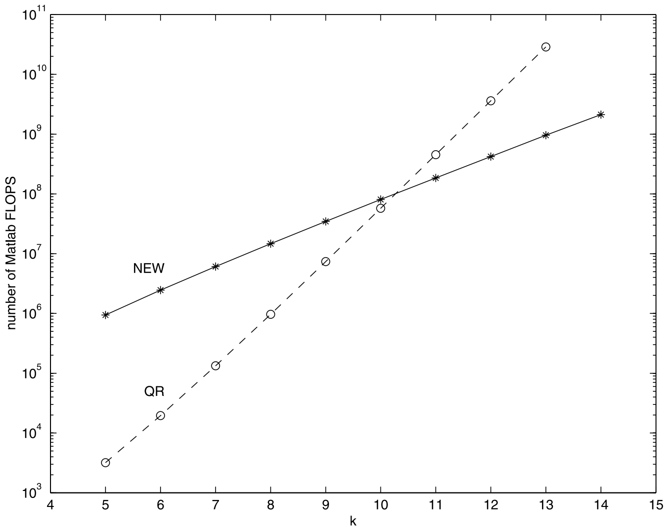

Fig. 1. Computational complexity (NEW versus QR).

number of difficult points is small compared to the total number M of interpolation points).

6.2. Accuracy

To investigate the accuracy obtained by NEW we consider the following three types of Toeplitz matrices:

1. The number of interpolation points $M = 2^k$ for $k = 5, \ldots, 14$. We choose $m = M/2$ and $n = M/4$. The entries of T are chosen uniformly at random in the interval (0,1). Fig. 2 plots the two-condition number of a sample of five such Toeplitz matrices for the cases $k = 5, 6, \ldots, 11$. These Toeplitz matrices appear to be generically well conditioned.
2. We consider ill-conditioned circulant Toeplitz matrices. Such matrices are rather easy to generate. The parameters M, m and n take the same values as for matrices of type 1. The two-condition number of these matrices is $\mathcal{O}(10^7)$. The extended matrix that appears in Eq. (3) is even more ill conditioned: its two-condition number is $\mathcal{O}(10^{12})$.
3. The prolate matrix (see [31]) of size 64×32 whose entries are determined by $\omega := 0.44$, $t_0 := 2\omega$ and $t_k := \sin(2\pi\omega k)/\pi k$ for $k \neq 0$. The number of interpolation points M is equal to 128. This matrix is well conditioned: its two-condition number is $\approx 3 \times 10^2$. The two-condition number of the extended matrix is $\approx 2 \times 10^5$.

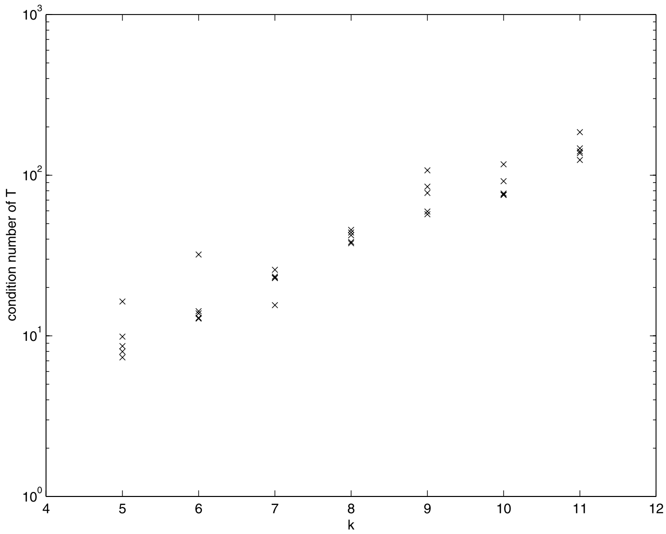

Fig. 2. Condition number (Toeplitz matrices of type 1).

We consider two types of right-hand side vectors:

1. The entries of b are computed such that $b = Tx$ where the entries of x are chosen uniformly at random in (0, 1). In this case we obtain small residuals Δa and Δb.
2. The entries of b are generated uniformly at random in (0, 1). This choice generally leads to large residuals Δa and Δb.

6.2.1. Matrices of type 1

In Figs. 3 and 4 we plot the relative norm of the residual

$$\left\| \begin{bmatrix} \Delta b \\ \Delta a \end{bmatrix} \right\| \Big/ \left\| \begin{bmatrix} \tilde{r} \\ \tilde{x} \end{bmatrix} \right\|$$

for right-hand sides of type 1 (small residuals) and type 2 (large residuals), respectively.

In Fig. 5 we plot $\|\Delta a\|$ for the case of right-hand sides of type 1.

The different lines in Figs. 3–7 correspond to the different stages of iterative refinement applied to the normal equations (3), as explained at the end of Section 2.

6.2.2. Matrices of type 2

In Figs. 6 and 7 we plot the relative norm of the residual and $\|\Delta a\|$, respectively, for right-hand sides of type 1 (small residuals).

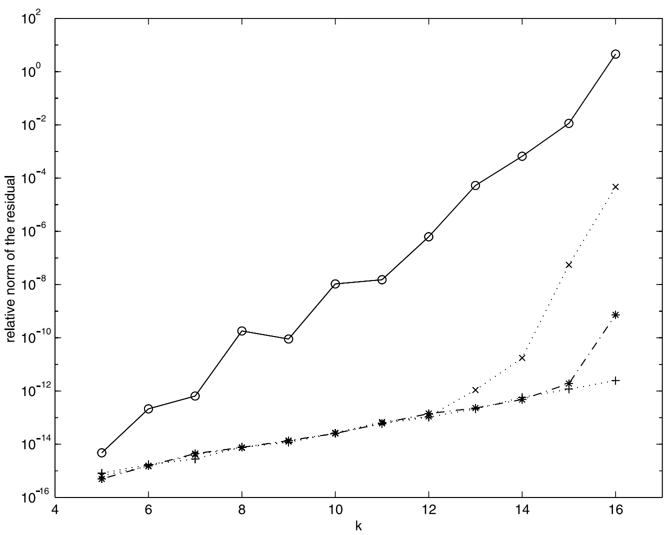

Fig. 3. Relative norm of the residual (right-hand sides of type 1).

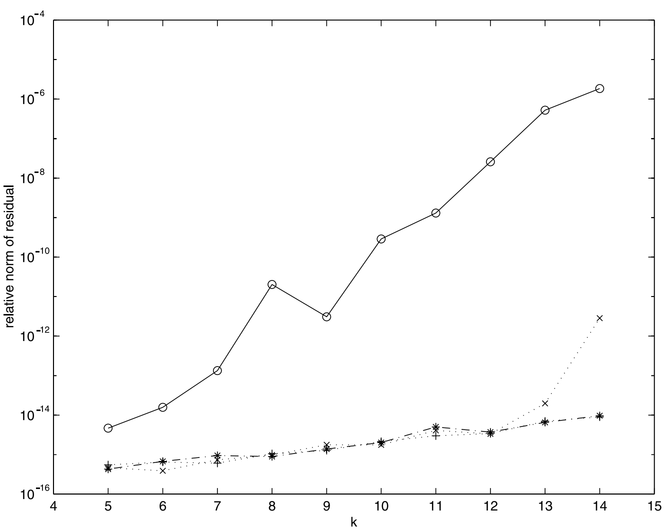

Fig. 4. Relative norm of the residual (right-hand sides of type 2).

For a given Toeplitz matrix of type 2, the results are in general better for right-hand sides of type 2 than for right-hand sides of type 1.

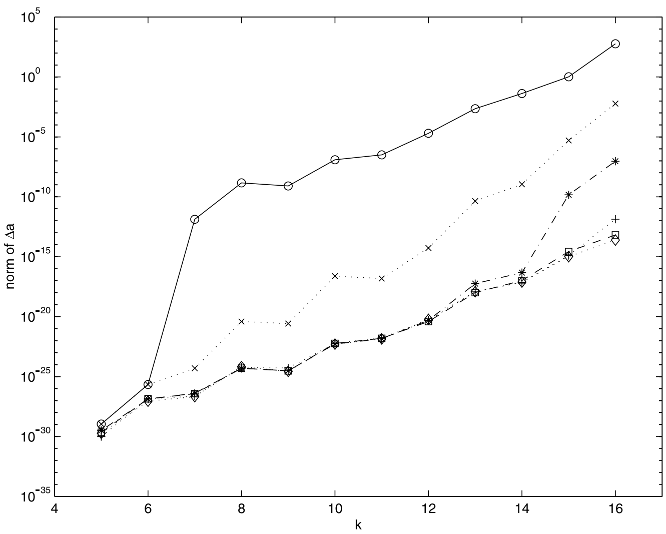

Fig. 5. $\|\Delta a\|$ (right-hand sides of type 1).

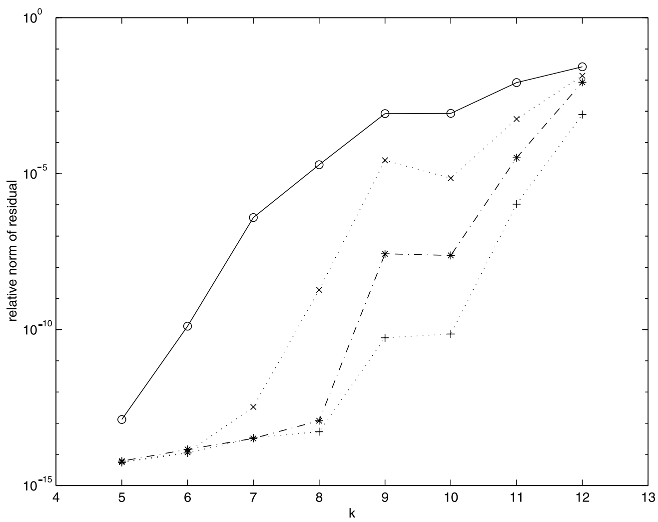

Fig. 6. Relative norm of the residual (right-hand sides of type 1).

6.2.3. The prolate matrix

In this case the algorithm NEW leads to unacceptable results. We proceed by putting some relative noise on the elements of the matrix: we replace t_k by $t_k(1 + \eta 10^{-4})$ where η is chosen uniformly at random in the interval (0, 1). The calculations are performed on the perturbed matrix. The residuals Δa and Δb are calculated

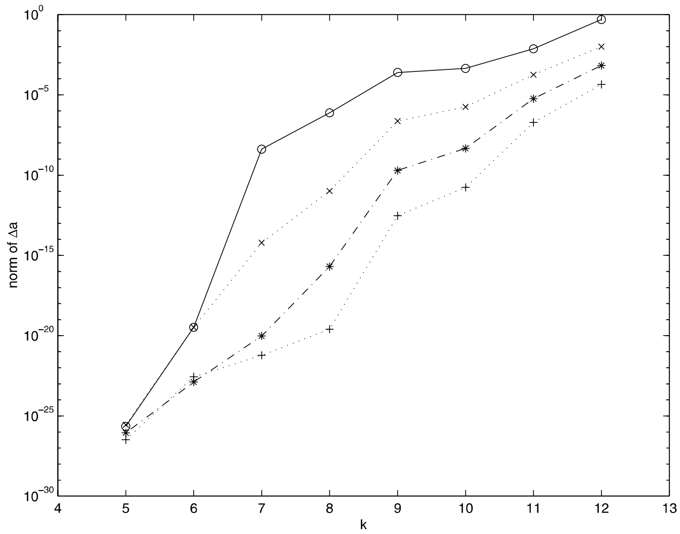

Fig. 7. $\|\Delta a\|$ (right-hand sides of type 1).

from the original matrix, though. After j steps of iterative refinement, we obtain the following results:

j	$\|\Delta b\|/\|\tilde{x}\|$	$\|\Delta a\|/\|\tilde{r}\|$
0	3×10^{-5}	5×10^{-6}
1	7×10^{-8}	2×10^{-7}
2	3×10^{-9}	1×10^{-9}
3	1×10^{-11}	9×10^{-12}
4	9×10^{-12}	2×10^{-13}

7. Conclusion

The previous numerical experiments show that for several large-size Toeplitz linear least squares problems the algorithm NEW leads to an accurate approximation of the solution. Moreover, the algorithm is generically superfast. However, it still needs to be investigated why our approach does not lead to acceptable results in case of the prolate matrix, unless its elements are perturbed.

Acknowledgments

This research was partially supported by the Fund for Scientific Research-Flanders (FWO-V), project "Orthogonal systems and their applications", grant

#G.0278.97, and project “Structured Matrices and their Applications”, grant #G.0078.01, by the K.U. Leuven (Bijzonder Onderzoeksfonds), project “SLAP: Structured Linear Algebra Package”, grant #OT/00/16, by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister’s Office for Science, Technology and Culture. The second author was supported by Kuwait University Research Grant SM 07/00 “Least squares and related problems for Toeplitz matrices”. The scientific responsibility rests with the authors.

References

[1] G.S. Ammar, W.B. Gragg, The generalized Schur algorithm for the superfast solution of Toeplitz systems, in: J. Gilewicz, M. Pindor, W. Siemaszko (Eds.), Rational Approximation and its Applications in Mathematics and Physics, Lecture Notes in Mathematics, vol. 1237, Springer, 1987, pp. 315.

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

[3] G.S. Ammar, W.B. Gragg, Numerical experience with a superfast real Toeplitz solver, Linear Algebra Appl. 121 (1989) 185–206.

[4] R.R. Bitmead, B.D.O. Anderson, Asymptotically fast solution of Toeplitz and related systems of linear equations, Linear Algebra Appl. 34 (1980) 103–116.

[5] A.W. Bojanczyk, R.P. Brent, F.R. de Hoog, *QR* factorization of Toeplitz matrices, Numer. Math. 49 (1) (1986) 81–94.

[6] R. Brent, F. Gustavson, D. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximants, J. Algorithms 1 (3) (1980) 259–295.

[7] A. Bultheel, M. Van Barel, in: Linear Algebra, Rational approximation and Orthogonal polynomials, Studies in Computational Mathematics, vol. 6, North-Holland, Elsevier Science, Amsterdam, 1997 (464 pages) ISBN: 0-444-82872-9.

[8] S. Chandrasekaran, 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.

[9] J. Chun, T. Kailath, H. Lev-Ari, Fast parallel algorithms for *QR* and triangular factorization, SIAM J. Sci. Statist. Comput. 8 (6) (1987) 899–913.

[10] G. Cybenko, A general orthogonalization technique with applications to time series analysis and signal processing, Math. Comp. 40 (1983) 323–336.

[11] G. Cybenko, Fast Toeplitz orthogonalization using inner products, SIAM J. Sci. Statist. Comput. 8 (5) (1987) 734–740.

[12] F. de Hoog, A new algorithm for solving Toeplitz systems of equations, Linear Algebra Appl. 88/89 (1987) 123–138.

[13] K.A. Gallivan, S. Thirumalai, P. Van Dooren, V. Vermaut, High performance algorithms for Toeplitz and block Toeplitz matrices, Linear Algebra Appl. 241–243 (1996) 343–388.

[14] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian elimination with partial pivoting for matrices with displacement structure, Math. Comp. 64 (212) (1995) 1557–1576.

[15] M. Gu, Stable and efficient algorithms for structured systems of equations, SIAM J. Matrix Anal. Appl. 19 (2) (1997) 279–306.

[16] M. Gu, New fast algorithms for structured linear least squares problems, SIAM J. Matrix Anal. Appl. 20 (1) (1998) 244–269.

[17] G. Heinig, Inversion of generalized Cauchy matrices and other classes of structured matrices, In: Linear Algebra in Signal Processing, IMA Volumes in Mathematics and its Applications, vol. 69, IMA, 1995, pp. 95–114.

[18] G. Heinig, Solving Toeplitz systems after extension and transformation, CALCOLO 33 (1996) 115–129.
[19] G. Heinig, A. Bojanczyk, Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices. I. Transformations, Linear Algebra Appl. 254 (1997) 193–226.
[20] G. Heinig, A. Bojanczyk, Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices. II. Algorithms, Linear Algebra Appl. 278 (1–3) (1998) 11–36.
[21] P. Kravanja, M. Van Barel, A fast block Hankel solver based on an inversion formula for block Loewner matrices. CALCOLO, 33(1–2):147–164, January–June 1996. Proceedings of the workshop Toeplitz Matrices: Structure, Algorithms and Applications, Cortona (Italy), September 9–12, 1996.
[22] P. Kravanja, M. Van Barel, A fast Hankel solver based on an inversion formula for Loewner matrices, Linear Algebra Appl. 282 (1–3) (1998) 275–295.
[23] P. Kravanja, M. Van Barel, A fast Hankel solver based on an inversion formula for Loewner matrices, Linear Algebra Appl. 282 (1–3) (1998) 275–295.
[24] S. Qiao, Hybrid algorithm for fast Toeplitz orthogonalization, Numer. Math. 53 (1988) 351–366.
[25] D. Sweet, Numerical Methods for Toeplitz Matrices, Ph.D. thesis, University of Adelaide, Adelaide, Australia, 1982.
[26] D.R. Sweet, Fast Toeplitz orthogonalization, Numer. Math. 43 (1) (1984) 1–21.
[27] D.R. Sweet, R.P. Brent, Error analysis of a fast partial pivoting method for structured matrices, in: T. Luk (Ed.), Advanced Signal Processing Algorithms Proceedings of SPIE, vol. 2563, 1995, pp. 266–280.
[28] M. Van Barel, A. Bultheel, A general module theoretic framework for vector M-Padé and matrix rational interpolation, Numer. Algorithms 3 (1992) 451–462.
[29] M. Van Barel, A. Bultheel, The "look-ahead" philosophy applied to matrix rational interpolation problems. In: U. Helmke, R. Mennicken, J. Saurer (Eds.), Systems and Networks: Mathematical Theory and Applications, Volume II: Invited and Contributed Papers, Mathematical Research, vol. 79, Akademie Verlag, 1994, pp. 891–894.
[30] M. Van Barel, G. Heinig, P. Kravanja, A stabilized superfast solver for nonsymmetric Toeplitz systems, SIAM J. Matrix Anal. Appl. 23 (2) (2001) 494–510.
[31] J.M. Varah, The prolate matrix, Linear Algebra Appl. 187 (1993) 269–278.