

# A superfast method for solving Toeplitz linear least squares problems

*Marc Van Barel*

*Georg Heinig*

*Peter Kravanja*

*Report TW336, February 2002*



Katholieke Universiteit Leuven  
Department of Computer Science

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# A superfast method for solving Toeplitz linear least squares problems

*Marc Van Barel*

*Georg Heinig*

*Peter Kravanja*

*Report TW 336, February 2002*

Department of Computer Science, K.U.Leuven

## **Abstract**

In this paper we develop a superfast algorithm to solve the linear least squares problem for a Toeplitz matrix. The original problem is first transformed into solving a linear  $2 \times 2$  block system where each block is a Toeplitz matrix. These Toeplitz blocks are extended into circulant blocks introducing new unknowns. Finally the block circulant linear system is transformed into an interpolation problem where a vector polynomial is sought satisfying certain tangential interpolation conditions for the roots of unity and having a specific degree for each of his components. This interpolation problem can be solved in a superfast way by a divide and conquer strategy. 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

**AMS(MOS) Classification :** Primary : 65F20, Secondary : 41A05.

Submitted to Linear Algebra and Its Applications

# A superfast method for solving Toeplitz linear least squares problems\*

Marc Van Barel<sup>†</sup>      Georg Heinig<sup>‡</sup>      Peter Kravanja<sup>§</sup>

## Abstract

In this paper we develop a superfast algorithm to solve the linear least squares problem for a Toeplitz matrix. The original problem is first transformed into solving a linear  $2 \times 2$  block system where each block is a Toeplitz matrix. These Toeplitz blocks are extended into circulant blocks introducing new unknowns. Finally the block circulant linear system is transformed into an interpolation problem where a vector polynomial is sought satisfying certain tangential interpolation conditions for the roots of unity and having a specific degree for each of his components. This interpolation problem can be solved in a superfast way by a divide and conquer strategy. 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  $b$ , 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*. One of the first fast algorithms was introduced by Sweet in his PhD thesis [18]. Other approaches include those by Bojanczyk, Brent and de Hoog [1], Chun, Kailath and Lev-Ari [3], Qiao [17], Cybenko [4, 5], Sweet [19] and many others. None of these algorithms has yet been shown to be numerically stable. For

---

\*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 and by Kuwait University Research Project SM–190. The scientific responsibility rests with the authors.

<sup>†</sup>Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium ([Marc.VanBarel@cs.kuleuven.ac.be](mailto:Marc.VanBarel@cs.kuleuven.ac.be))

<sup>‡</sup>Kuwait University, Department of Mathematics, POB 5969, Safat 13060, Kuwait ([georg@mcs.sci.kuniv.edu.kw](mailto:georg@mcs.sci.kuniv.edu.kw))

<sup>§</sup>Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan 200 A, B-3001 Heverlee, Belgium ([Peter.Kravanja@na-net.ornl.gov](mailto:Peter.Kravanja@na-net.ornl.gov))

several of them examples exist indicating that the approach is actually unstable in certain cases.

Recently, Gu [9] has developed fast algorithms for solving Toeplitz as well as Toeplitz-plus-Hankel linear least squares problems. His algorithm is based on the fact that a Toeplitz or Toeplitz-plus-Hankel matrix can be transformed into a Cauchy-like matrix with the help of the discrete Fourier transformation or a trigonometric transformation. Cauchy-like matrices have the advantage, compared to Toeplitz matrices, that permutations of rows and columns do not destroy the structure, so pivoting strategies are possible in order to stabilize the algorithm. This idea was first mentioned in [10] and further developed in [7, 6, 8, 11, 12, 13, 14, 15, 20] and other papers.

In Gu's paper [9] the Toeplitz least squares problem is transformed into the inversion problem for two Cauchy-like matrices. Numerical experiments indicate that Gu's approach is not only efficient but also numerically 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 [11], [16] and [23]. In the latter paper a superfast Toeplitz solver was proposed. The main aim of the present paper is to apply some modification of the idea in [23].

Our approach can be briefly described as follows. First we reduce in Section 2 the least squares problem for  $T$  to the problem of the solution of a linear system with a Toeplitz block coefficient matrix  $R$  with  $T$  as one of its blocks. This system is somehow related the normal system  $T^H T x = T^H b$  where  $T^H$  denotes the complex conjugate transpose of the matrix  $T$ . 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 in Section 5 standard solution procedures for this kind of problems (see [2, Chapter 7]) can be applied. The inclusion of a divide-and-conquer principle as in [23] speeds up the algorithm from  $O(mn)$  complexity to complexity  $O((m+n)\log^2(m+n))$ . We refrain from presenting the algorithm in all its details as these are given in [23]. The selection of "difficult points" guarantees a certain degree of stability in the process, and iterative refinement improves the result. The numerical issues will be discussed in Section 6.

## 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

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

Furthermore,

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

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

**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*

$$R \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}. \quad (3)$$

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

$$\begin{bmatrix} \tilde{r} \\ \tilde{x} \end{bmatrix} \quad (4)$$

is at our disposal, with corresponding residuals  $\Delta a$  and  $\Delta b$ :

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

Iterative refinement of (4) is based on the following fact: if  $\Delta r$  and  $\Delta x$  are such that

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

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  $[\tilde{r} \ \tilde{x}]^T$  to  $[r \ x]^T$  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

$$R \begin{bmatrix} \rho \\ \xi \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} \quad (6)$$

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 [23] 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

$$S = [s_{j-k}]_{\substack{k=0,1,\dots,n-1 \\ j=0,1,\dots,m-1}}$$

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  $M \geq m + n - 1$ . If  $M = m + n - 1$ , then we have to choose  $\tilde{S}$  as the  $(n - 1) \times n$  Toeplitz matrix

$$\tilde{S} := [s_{-n+1+j-k}]_{\substack{k=0,1,\dots,n-1 \\ j=0,1,\dots,n-2}}$$

where  $s_{-n-k} := s_{m-k-1}$  for  $k = 0, 1, \dots, n - 1$ . 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

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

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

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

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

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

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  $C_1, \dots, C_7$  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

$$C = \mathcal{F}_M^H \Lambda \mathcal{F}_M$$

where  $\Lambda$  is a  $M \times M$  diagonal matrix containing the eigenvalues of  $C$  and  $\mathcal{F}_M$  denotes the  $M \times M$  Discrete Fourier Transform matrix

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

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

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

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

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

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

## 4 Interpolation interpretation

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

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  $\deg \xi(z) \leq n - 1$ ,  $\deg \zeta(z) \leq M - m - 1$ ,  $\deg \rho(z) \leq m - 1$  and  $\deg \sigma(z) \leq M - n - 1$ , 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  $k = 1, \dots, M$ . In other words, determine the vector polynomial  $p(z) \in \mathbb{C}[z]^{5 \times 1}$  of (component-wise) degree

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

such that

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

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  $k = 1, \dots, M$ .

## 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  $\tau$ -degree. We therefore start by recalling these concepts, adapting the definitions to our specific case.

Let  $\mathcal{S}$  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  $\mathcal{S}$  forms a submodule of the  $\mathbb{C}[z]$ -module  $\mathbb{C}[z]^{5 \times 1}$ . A basis for  $\mathcal{S}$  always consists of exactly five elements [21, Theorem 3.1]. Let  $\{B_1(z), B_2(z), \dots, B_5(z)\}$  be a basis for  $\mathcal{S}$ . Then every element  $p(z) \in \mathcal{S}$  can be written in a unique way as  $p(z) = \sum_{i=1}^5 \alpha_i(z) B_i(z)$  with  $\alpha_i(z) \in \mathbb{C}[z]$ ,  $i = 1, 2, \dots, 5$ . The matrix polynomial  $B(z) := [B_1(z) \ B_2(z) \ \dots \ B_5(z)] \in \mathbb{C}[z]^{5 \times 5}$  is called a *basis matrix*. Basis matrices can be characterized as follows.

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

*Proof.* This follows immediately from [21, Theorem 4.1].  $\square$

Within the submodule  $\mathcal{S}$  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  $\tau$ -degree [21]. Let  $\tau \in \mathbb{Z}^5$ . The  $\tau$ -degree of a vector polynomial  $p(z) = [p_1(z) \ p_2(z) \ \cdots \ p_5(z)]^T \in \mathbb{C}[z]^{5 \times 1}$  is defined as a generalization of the classical degree:

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

with  $\tau\text{-deg } 0 := -\infty$ . The  $\tau$ -highest degree coefficient of a vector polynomial

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

with  $\tau$ -degree  $\delta$  is defined as the vector  $[\omega_1 \ \omega_2 \ \cdots \ \omega_5]^T$  with  $\omega_i$  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  $\tau$ -reduced if the  $\tau$ -highest degree coefficients are linearly independent. Every basis of  $\mathcal{S}$  can be transformed into a  $\tau$ -reduced one. For details, we refer to [21]. Once we have a basis in  $\tau$ -reduced form, the elements of  $\mathcal{S}$  can be parametrized as follows.

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

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

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

*Proof.* See Van Barel and Bultheel [21, Theorem 3.2].  $\square$

We will solve the interpolation problem (7) by constructing a  $5 \times 5$  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  $\tau$ -reduced with  $\tau = (m, 1, M - n, n, M - m)$ . Then this basis matrix will have just one column having  $\tau$ -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  $\tau$ -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  $\kappa$  and  $K - \kappa$ , respectively. Given a  $\tau$ -reduced basis matrix for the problem of size  $\kappa$ , the theorem shows how the interpolation data for the remaining problem of size  $K - \kappa$  needs to be modified, such that the  $\tau$ -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  $\tau$ -reduced basis matrix for the problem of size  $\kappa$  and the  $\tau$ -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  $\phi_k \neq [0 \ 0 \ \dots \ 0]^T$  for  $k = 1, \dots, K$ . Let  $1 \leq \kappa \leq K$ . Let  $\tau_K \in \mathbb{Z}^5$ . Suppose that  $B_\kappa(z) \in \mathbb{C}[z]^{5 \times 5}$  is a  $\tau_K$ -reduced basis matrix with basis vectors having  $\tau_K$ -degree  $\delta_i$  for  $i = 1, 2, \dots, 5$ , corresponding to the interpolation data*

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

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

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

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

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

*Proof.* See Van Barel and Bultheel [22, Theorem 3]. □

In [23], 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 [23]. Note that in [23] the basis matrix is of size  $2 \times 2$  whereas in the present paper it is of size  $5 \times 5$ .

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

The presence of zeros in  $F_k$  can be exploited to obtain the  $5 \times 5$  basis matrix  $B_{FG}(z)$  as the product of the basis matrices  $B_F(z)$  and  $B_{F \rightarrow FG}(z)$  where  $B_F(z)$  can be derived from a  $4 \times 4$  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  $5 \times 5$  basis matrix as the product of  $B_G(z)$  and  $B_{G \rightarrow FG}(z)$  where  $B_G(z)$  can be obtained from a basis matrix of size  $3 \times 3$ .

## 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  $u \approx 1.11 \times 10^{-16}$ .

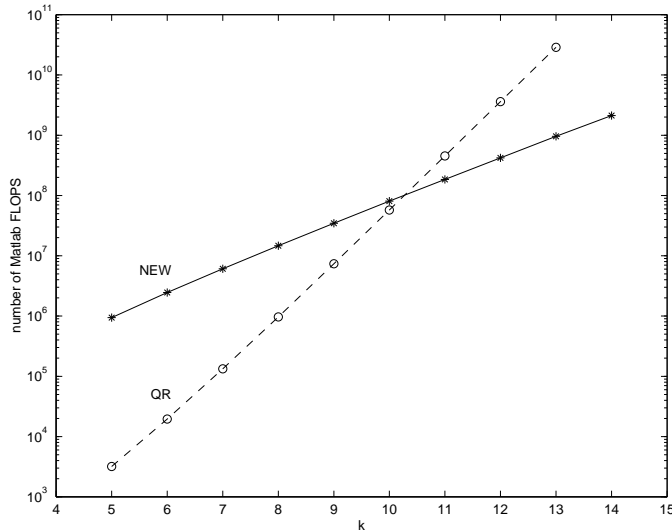


Figure 1: Computational complexity (NEW versus QR)

## 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  $k = 5, \dots, 14$ . We choose  $m = M/2$  and  $n = M/4$ . Figure 1 plots the number of flops required by NEW and by QR, respectively. It follows that NEW is indeed superfast (as long as the 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  $k = 5, \dots, 14$ . We choose  $m = M/2$  and  $n = M/4$ . The entries of  $T$  are chosen uniformly at random in the interval  $(0, 1)$ . Figure 2 plots the 2-condition number of a sample of five such Toeplitz matrices for the cases  $k = 5, 6, \dots, 11$ . 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 2-condition number of these matrices is  $\mathcal{O}(10^7)$ . The extended matrix that appears in Equation (3) is even more ill-conditioned: its 2-condition number is  $\mathcal{O}(10^{12})$ .
3. The prolate matrix (see [24]) of size  $64 \times 32$  whose entries are determined by  $\omega := 0.44$ ,  $t_0 := 2\omega$  and  $t_k := \frac{\sin(2\pi\omega k)}{\pi k}$  for  $k \neq 0$ . The number of interpolation points  $M$  is equal to 128. This matrix is well-conditioned: its 2-condition number is  $\approx 3 \cdot 10^2$ . The 2-condition number of the extended matrix is  $\approx 2 \cdot 10^5$ .

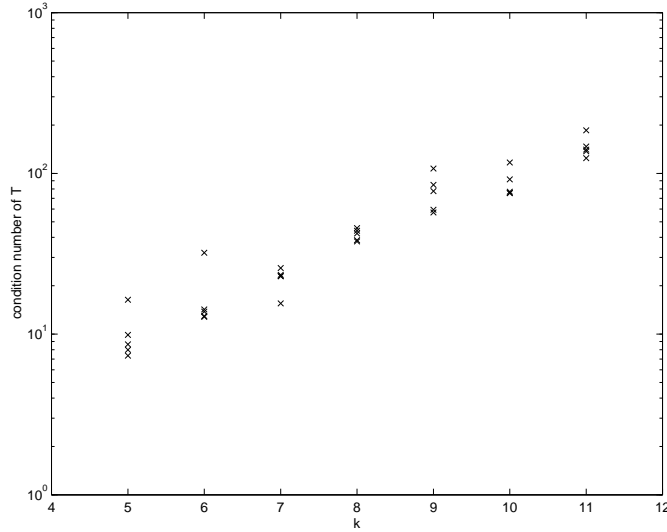


Figure 2: Condition number (Toeplitz matrices of type 1)

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  $\Delta a$  and  $\Delta b$ .
2. The entries of  $b$  are generated uniformly at random in  $(0, 1)$ . This choice generally leads to large residuals  $\Delta a$  and  $\Delta b$ .

### 6.2.1 Matrices of type 1

In Figures 3 and 4 we plot the relative norm of the residual

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

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

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

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

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.

### 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  $\eta$  is chosen

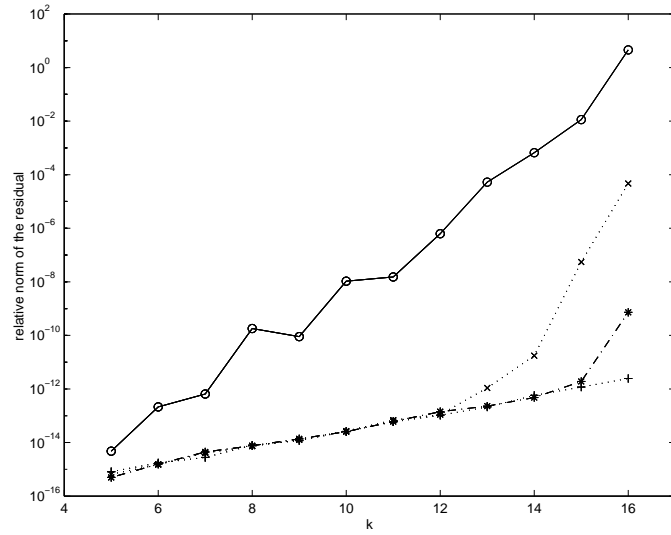


Figure 3: Relative norm of the residual (right-hand sides of type 1)

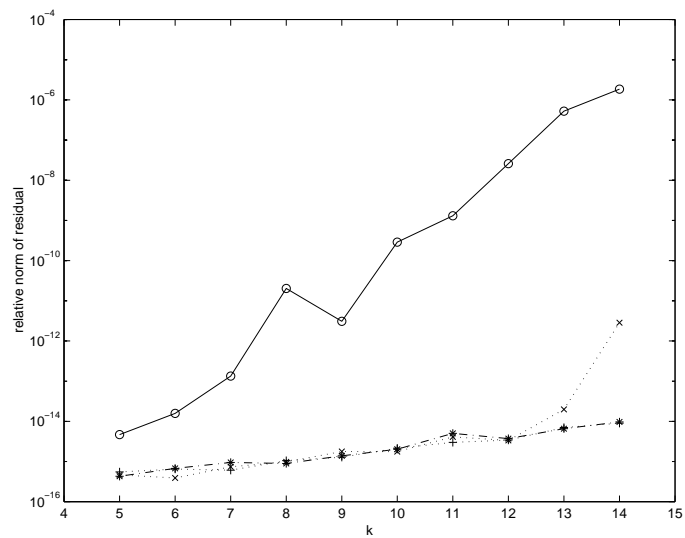


Figure 4: Relative norm of the residual (right-hand sides of type 2)

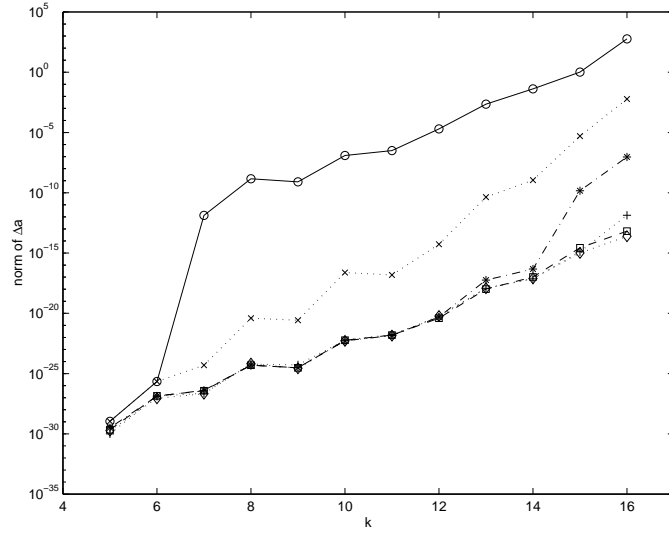


Figure 5:  $\|\Delta a\|$  (right-hand sides of type 1)

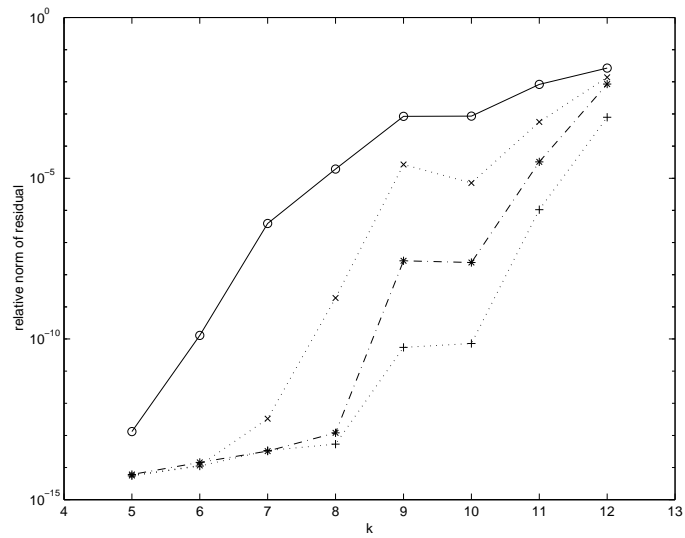


Figure 6: Relative norm of the residual (right-hand sides of type 1)

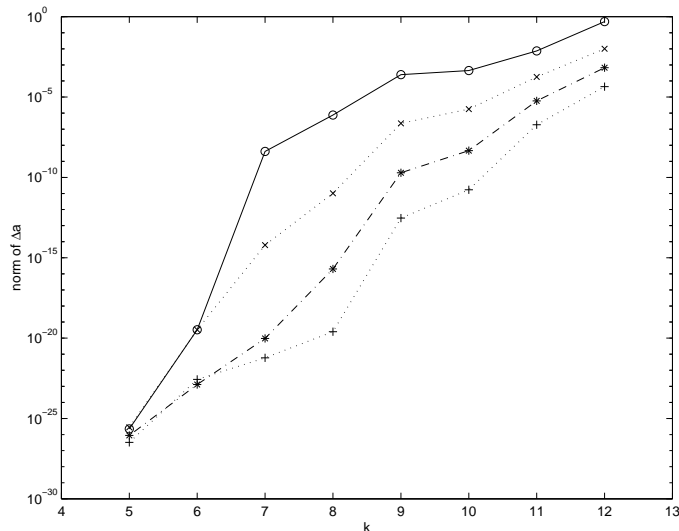


Figure 7:  $\|\Delta a\|$  (right-hand sides of type 1)

uniformly at random in the interval  $(0, 1)$ . The calculations are performed on the perturbed matrix. The residuals  $\Delta a$  and  $\Delta b$  are calculated 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.

## References

- [1] A. W. Bojanczyk, R. P. Brent, and F. R. de Hoog. *QR* factorization of Toeplitz matrices. *Numer. Math.*, 49(1):81–94, July 1986.
- [2] A. Bultheel and M. Van Barel. *Linear Algebra, Rational approximation and Orthogonal polynomials*, volume 6 of *Studies in Computational Mathematics*. North-Holland, Elsevier Science, Amsterdam, 1997. (464 pages) ISBN: 0-444-82872-9.
- [3] J. Chun, T. Kailath, and H. Lev-Ari. Fast parallel algorithms for *QR* and triangular factorization. *SIAM J. Sci. Stat. Comput.*, 8(6):899–913, November 1987.

- [4] G. Cybenko. A general orthogonalization technique with applications to time series analysis and signal processing. *Math. Comput.*, 40:323–336, 1983.
- [5] G. Cybenko. Fast Toeplitz orthogonalization using inner products. *SIAM J. Sci. Stat. Comput.*, 8(5):734–740, September 1987.
- [6] K. A. Gallivan, S. Thirumalai, P. Van Dooren, and V. Vermaut. High performance algorithms for Toeplitz and block Toeplitz matrices. *Linear Algebr. Appl.*, 241–243:343–388, 1996.
- [7] I. Gohberg, T. Kailath, and V. Olshevsky. Fast Gaussian elimination with partial pivoting for matrices with displacement structure. *Math. Comput.*, 64(212):1557–1576, 1995.
- [8] M. Gu. Stable and efficient algorithms for structured systems of equations. *SIAM J. Matrix Analysis Appl.*, 19, 2:279–306, 1997.
- [9] M. Gu. New fast algorithms for structured linear least squares problems. *SIAM J. Matrix Anal. Appl.*, 20(1):244–269, September 1998.
- [10] G. Heinig. Inversion of generalized Cauchy matrices and other classes of structured matrices. In *Linear Algebra in Signal Processing*, volume 69 of *IMA volumes in Mathematics and its Applications*, pages 95–114. IMA, 1994.
- [11] G. Heinig. Solving Toeplitz systems after extension and transformation. *CALCOLO*, 33:115–129, 1996.
- [12] G. Heinig and A. Bojanczyk. Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices: I. Transformations. *Linear Algebr. Appl.*, 254:193–226, March 1997.
- [13] G. Heinig and A. Bojanczyk. Transformation techniques for Toeplitz and Toeplitz-plus-Hankel matrices: II. Algorithms. *Linear Algebr. Appl.*, 278(1–3):11–36, 1998.
- [14] P. Kravanja and 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.
- [15] P. Kravanja and M. Van Barel. A fast Hankel solver based on an inversion formula for Loewner matrices. *Linear Algebr. Appl.*, 282(1–3):275–295, 1998.
- [16] P. Kravanja and M. Van Barel. A fast Hankel solver based on an inversion formula for Loewner matrices. *Linear Algebr. Appl.*, 282(1–3):275–295, September 1998.
- [17] S. Qiao. Hybrid algorithm for fast Toeplitz orthogonalization. *Numer. Math.*, 53:351–366, 1988.
- [18] D. Sweet. *Numerical Methods for Toeplitz Matrices*. PhD thesis, University of Adelaide, Adelaide, Australia, 1982.
- [19] D. R. Sweet. Fast Toeplitz orthogonalization. *Numer. Math.*, 43(1):1–21, January 1984.

- [20] D. R. Sweet and R. P. Brent. Error analysis of a fast partial pivoting method for structured matrices. In T. Luk, editor, *Advanced Signal Processing Algorithms, Proceedings of SPIE-1995*, volume 2563, pages 266–280, 1995.
- [21] M. Van Barel and A. Bultheel. A general module theoretic framework for vector M-Padé and matrix rational interpolation. *Numer. Algorithms*, 3:451–462, 1992.
- [22] M. Van Barel and A. Bultheel. The “look-ahead” philosophy applied to matrix rational interpolation problems. In U. Helmke, R. Mennicken, and J. Saurer, editors, *Systems and networks: Mathematical theory and applications, Volume II: Invited and contributed papers*, volume 79 of *Mathematical Research*, pages 891–894. Akademie Verlag, 1994.
- [23] M. Van Barel, G. Heinig, and P. Kravanja. A stabilized superfast solver for nonsymmetric Toeplitz systems. *SIAM J. Matrix Anal. Appl.*, 23(2):494–510, 2001.
- [24] J. M. Varah. The prolate matrix. *Linear Algebr. Appl.*, 187:269–278, July 1993.