A Superfast Solver for Real Symmetric Toeplitz Systems Using Real Trigonometric Transformations

G. Codevico　　*G. Heinig*　　*M. Van Barel*

Report TW 386, March 2004

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

A Superfast Solver for Real Symmetric Toeplitz Systems Using Real Trigonometric Transformations

G. Codevico *G. Heinig* *M. Van Barel*

Report TW 386, March 2004

Department of Computer Science, K.U.Leuven

Abstract

A new superfast $O(n \log^2 n)$ complexity direct solver for real symmetric Toeplitz systems is presented. The algorithm is based on 1. the reduction to symmetric right-hand sides, 2. a polynomial interpretation in terms of Chebyshev polynomials, 3. an inversion formula involving real trigonometric transformations, and 4. an interpretation of the equations as a tangential interpolation problem. The tangential interpolation problem is solved via a divide-and-conquer strategy and fast DCT.

Keywords : symmetric Toeplitz matrix, superfast algorithm, tangential interpolation, cosine and sine transform.
AMS(MOS) Classification : Primary : 65F05, Secondary : 15A09, 65D05.

A Superfast Solver for Real Symmetric Toeplitz Systems Using Real Trigonometric Transformations

G. Codevico[1,*], G. Heinig[2], M. Van Barel[1]

[1] *Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Leuven (Heverlee), Belgium*
[2] *Dept. of Mathematics and Comp. Science, Kuwait University, P. O. Box 5969, Safat 13060, Kuwait*

SUMMARY

A new superfast $O(n \log^2 n)$ complexity direct solver for real symmetric Toeplitz systems is presented. The algorithm is based on 1. the reduction to symmetric right-hand sides, 2. a polynomial interpretation in terms of Chebyshev polynomials, 3. an inversion formula involving real trigonometric transformations, and 4. an interpretation of the equations as a tangential interpolation problem. The tangential interpolation problem is solved via a divide-and-conquer strategy and fast DCT.

KEY WORDS: symmetric Toeplitz matrix, superfast algorithm, tangential interpolation, cosine and sine transform

1. INTRODUCTION

In this paper a new fast algorithm for the solution of linear systems of equations

$$T_n \mathbf{x} = \mathbf{b} \tag{1}$$

with a nonsingular real $n \times n$ symmetric Toeplitz coefficient matrix $T_n = [\, a_{|i-j|} \,]_{i,j=0}^{n-1}$ is presented that computes the solution with computational complexity $O(n \log^2 n)$. Algorithms with this complexity can be found in the literature, in [12], [2], [3], [4], [1], [8], [16] and other

*Correspondence to: Marc Van Barel, Department of Computer Science, K.U.Leuven, Celestijnenlaan 200A, B-3001 Leuven (Heverlee), Belgium

Contract/grant sponsor: The research of the first and third author was partially supported by the Research Council K.U.Leuven, project OT/00/16 (SLAP: Structured Linear Algebra Package), by the Fund for Scientific Research–Flanders (Belgium), projects G.0078.01 (SMA: Structured Matrices and their Applications), G.0176.02 (ANCILA: Asymptotic aNalysis of the Convergence behavior of Iterative methods in numerical Linear Algebra), and G.0455.0 (RHPH: Riemann-Hilbert problems, random matrices and Padé-Hermite approximation), and by the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture, project IUAP V-22 (Dynamical Systems and Control: Computation, Identification & Modelling). The scientific responsibility rests with the authors.

papers (see [13]). There are also algorithms with complexity $O(n\log^3 n)$ (see [14]). All these algorithms are based on the Fast Fourier Transform. In contrast to this, the new algorithm is based on real trigonometric transforms. Since the complexity for trigonometric transforms is essentially less than that one for complex FFT, even less than real FFT (see [15]), the new algorithm should be, if the implementation details are worked out properly, faster than previous ones.

For simplicity of presentation we assume throughout the paper that n is even and $n = 2m$. The case of odd n is analogous.

The approach in this paper is based on the results from [6]. Let us sketch the main ideas. We utilize the fact that a symmetric Toeplitz matrix is also centrosymmetric, which means that $J_nT_nJ_n = T_n$. Here J_n denotes the $n\times n$ matrix of counteridentity, which has ones on the antidiagonal and zeros elsewhere. We set $\widehat{\mathbf{u}} = J_n\mathbf{u}$. Due to centrosymmetry, (1) is equivalent to two systems, one with a symmetric and one with a skewsymmetric right-hand side. These systems can be written as $m\times m$ systems with a Toeplitz-plus-Hankel coefficient matrix

$$R_m^{\pm} = [\, a_{|i-j|} \pm a_{i+j+1}\,]_{i,j=0}^{m-1}\,, \tag{2}$$

respectively.

In Section 2 we show that the system with a skewsymmetric right-hand side can be reduced to two systems with symmetric right-hand sides, so it is sufficient to consider systems with coefficient matrix R_m. In the sequel, we skip the superscript "+" in R_m^+.

In Section 3 we show that the matrix R_m has a polynomial interpretation in terms of Chebyshev polynomials of third kind. For this reason R_m is called *Chebyshev-Hankel matrix* (of third kind) or briefly *CH-matrix*. In Section 4 we recall an inversion formula from [6] and show how the inverse of R_m can be represented in terms of cosine transforms of two "fundamental polynomials" and diagonal matrices. In Section 5 we show that the fundamental polynomials can be found via the solution of a tangential interpolation problem. The translation into an interpolation problem is different from that in [6]. It uses an idea from [5]. In Section 6 we explain our algorithm and Section 7 reports about the numerical experiments.

2. REDUCTION TO SYMMETRIC SYSTEMS

Let $\mathbb{R}^n_{\pm}$ denote the subspace of all symmetric and skewsymmetric vectors in $\mathbb{R}^n$, respectively. We introduce the matrices $P_{\pm} = \frac{1}{2}(I_n \pm J_n)$. The linear operators corresponding to these matrices are orthoprojectors onto $\mathbb{R}^n_{\pm}$ and $P_+ + P_- = I_n$. Since the symmetric Toeplitz matrix maps $\mathbb{R}^n_{\pm}$ onto itself, the system (1) splits into the two systems $T_n\mathbf{x}_{\pm} = P_{\pm}\mathbf{b}$, where $\mathbf{x} = \mathbf{x}_- + \mathbf{x}_+$. This leads to the following.

Proposition 2.1. *Suppose that* $P_{\pm}\mathbf{b} = \begin{bmatrix} \pm\widehat{\mathbf{c}}_{\pm} \\ \mathbf{c}_{\pm} \end{bmatrix}$ *and* $\mathbf{y}_{\pm}$ *is the solution of* $R_m^{\pm}\mathbf{y}_{\pm} = \mathbf{c}_{\pm}$. *Then the solution of* (1) *is given by*

$$\mathbf{x} = \begin{bmatrix} -\widehat{\mathbf{y}}_- \\ \mathbf{y}_- \end{bmatrix} + \begin{bmatrix} \widehat{\mathbf{y}}_+ \\ \mathbf{y}_+ \end{bmatrix}.$$

We show now that it is sufficient to solve systems with symmetric right-hand side. For this we use polynomial language. If $\mathbf{x} = (x_i)_{i=1}^n$, then $\mathbf{x}(t) := \sum_{i=1}^{n} x_i t^{i-1}$. Let W denote the operator defined by

$$(W\mathbf{x}_-)(t) = \frac{t+1}{t-1}\,\mathbf{x}_-(t) \quad (\mathbf{x}_- \in \mathbb{R}^n_-).$$

W maps $\mathbb{R}^n_-$ into $\mathbb{R}^n_+$. Since n is even, the map is onto. Let $T_\pm$ denote the restriction from T_n to $\mathbb{R}^n_\pm$.

Proposition 2.2. *The operators T_+ and T_- are related via*

$$T_+W - WT_- = 2\mathbf{e}\,\mathbf{a}^T(S - I_n)^{-1},$$

where $\mathbf{a} = [\,a_n\ \dots\ a_1\,]$, $\mathbf{e} = [\,1\ \dots\ 1\,]$ and S is the forward shift.

Proof. Let $T_-\mathbf{x}_- = \mathbf{b}_-$, $\mathbf{z}(t) = \frac{1}{t-1}\,\mathbf{x}_-(t)$. Since $\mathbf{z} \in \mathbb{R}^{n-1}_+$, we have

$$T_n \begin{bmatrix} \mathbf{z} & 0 \\ 0 & \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_+ & r \\ r & \mathbf{r}_+ \end{bmatrix}$$

for some $\mathbf{r}_+ \in \mathbb{R}^{n-1}_+$ and $r = \mathbf{a}^T\mathbf{z}$. Hence

$$\mathbf{b}_-(t) = (t-1)\mathbf{r}_+(t) - r\,(t^{n-1} - 1),$$

and

$$\mathbf{r}_+(t) = \frac{1}{t-1}\,\mathbf{b}_-(t) + r\,\frac{t^{n-1}-1}{t-1}.$$

We conclude

$$(T_n\mathbf{z})(t) = \frac{1}{t-1}\,\mathbf{b}_-(t) + r\,\frac{t^{n-1}-1}{t-1} + r\,t^{n-1} = \frac{1}{t-1}\,\mathbf{b}_-(t) + r\,\frac{t^n-1}{t-1}.$$

Let $\mathbf{x}_+ = W\mathbf{x}_-$. Since $\mathbf{x}_+(t) = \mathbf{x}_-(t) + 2\,\mathbf{z}(t)$, we have

$$(T_+\mathbf{x}_+)(t) = (T_-\mathbf{x}_-)(t) + 2\,(T_n\mathbf{z})(t) = \frac{t+1}{t-1}\,\mathbf{b}_-(t) + 2r\,\frac{t^n-1}{t-1}\ .$$

In matrix language this means

$$T_+W\mathbf{x}_- = WT_-\mathbf{x}_- + 2r\,\mathbf{e}.$$

It remains to mention that $\mathbf{z} = (Z - I_n)^{-1}\mathbf{x}_-$. ∎

This proposition can be applied to transform $T_n\mathbf{x}_- = \mathbf{b}_-$ into systems with symmetric right-hand sides. In fact, let $T_n\mathbf{x}_+ = \mathbf{b}_+$ and $T_n\mathbf{v}_+ = \mathbf{e}$. From Proposition 2.2 we conclude that $T_+(W\mathbf{x}_- - 2r\,\mathbf{v}_+) = W\mathbf{b}_-$ for some constant r, which implies $\mathbf{x}_- = W^{-1}(\mathbf{x}_+ + 2r\,\mathbf{v}_+)$. The constant r can be found by multiplying one row of T_n by $\mathbf{x}_-$.

3. POLYNOMIAL INTERPRETATION

CH-matrices can be interpreted as matrices of multiplication operators in bases of Chebyshev polynomials. We need Chebyshev polynomials of first and third kind (cf. [11]),

$$\mathbf{t}_k(\cos\theta) = \cos k\theta, \quad \mathbf{h}_k(\cos\theta) = \frac{\cos(k+1/2)\theta}{\cos\theta/2} \quad (k \in \mathbb{Z})\ .$$

It is important to mention the symmetry properties

$$\mathbf{t}_{-k}(t) = \mathbf{t}_k(t), \quad \text{and} \quad \mathbf{h}_{-k}(t) = \mathbf{h}_{k-1}(t)\ .$$

and the relations

$$2\,\mathbf{t}_i(t)\mathbf{h}_k(t) = \mathbf{h}_{i+k}(t) + \mathbf{h}_{k-i}(t) = \mathbf{h}_{i+k}(t) + \mathbf{h}_{i-k-1}(t)\ , \tag{3}$$

which immediately follow from the corresponding trigonometric identities.

Proposition 3.1. *Suppose that $f(t) = a_0 + 2\sum_{k=1}^{2m-1} a_k\,\mathbf{t}_k(t)$. Then matrix M_m of the operator of multiplication $\mathbf{x}(t) \to f(t)\mathbf{x}(t)$, $\mathbf{x} \in \mathbb{R}^m$, with respect to the basis $\{\mathbf{h}_k(t)\}$ is given by*

$$M_m = [\, a_{|i-j|} + a_{i+j+1}\,]_{i=0,\ \ j=0}^{3m-2,\ m-1}\ ,$$

where we set $a_i = 0$ for $i \geq 2m$.

Proof. In view of (3), we have for $j \neq 0$

$$\begin{aligned} f(t)\mathbf{h}_j(t) &= a_0\mathbf{h}_j(t) + \sum_{k=1}^{2m-1} a_k(\mathbf{h}_{j+k}(t) + \mathbf{h}_{j-k}(t)) \\ &= a_0\mathbf{h}_j(t) + \sum_{i=j+1}^{2m+j-1} a_{i-j}\mathbf{h}_i(t) + \sum_{i=0}^{j-1} a_{j-i}\mathbf{h}_i(t) + \sum_{i=0}^{2m-j-2} a_{i+j+1}\mathbf{h}_i(t)\ . \end{aligned}$$

It remains to collect the coefficients corresponding to each $\mathbf{h}_i(t)$. ■

The CH-matrix R_m is given by $R_m = [\, I_m \ \ 0\,]\ M_m$.

We introduce a notation. If $\mathbf{u} = (u_k)_{k=0}^m$, then $\mathbf{u}^+(t)$ will denote the polynomial $\mathbf{u}^+(t) = \sum_{k=0}^{m} u_k\mathbf{h}_k(t)$. From Proposition 3.1 we conclude the following.

Corollary 3.2. *The vector $\mathbf{y}$ is the solution of the system $R_m\mathbf{y} = \mathbf{c}$ if and only if $f(t)\mathbf{y}^+(t) = \mathbf{c}^+(t) + \xi^+(t)$ for some $\xi^+(t) = \sum_{k=0}^{2m-2} \xi_k\,\mathbf{h}_{k+m}(t)$.*

4. INVERSION FORMULA

In this section we present a formula for the inverse of the CH-matrix R_m.

If $B = [\, b_{ij}\,]_{i,j=0}^{m-1}$ is any $m \times m$ matrix, then we denote by $B^+(t,s)$ the bivariate polynomial

$$B^+(t,s) = \sum_{i,j=0}^{m-1} b_{ij}\,\mathbf{h}_i(t)\mathbf{h}_j(s)\ .$$

Let R_m be a nonsingular CH-matrix as above and $\mathbf{u}'$ and $\mathbf{v}'$ be the solutions of

$$R_m\mathbf{u}' = -\mathbf{a}, \quad R_m\mathbf{v}' = \mathbf{e}_m,$$

where $\mathbf{e}_m$ denotes the last unit vector in the standard basis of $\mathbb{R}^m$ and $\mathbf{a} = (a_{n-i}+a_{i+n+1})_{i=0}^{m-1}$, and let the vectors $\mathbf{u}$, $\mathbf{v}$ be defined by

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}' \\ 1 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}' \\ 0 \end{bmatrix} .$$

The following is proved in [6].

Theorem 4.1. *The inverse of R_m is given by*

$$(R_m^{-1})^+(t,s) = \frac{1}{2}\,\frac{\mathbf{u}^+(t)\mathbf{v}^+(s) - \mathbf{v}^+(t)\mathbf{u}^+(s)}{t-s} . \tag{4}$$

Matrices like on the right-hand side of (4) are called *CH-Bezoutian* (of the third kind). The polynomials $\mathbf{u}^+(t)$ and $\mathbf{v}^+(t)$ occurring in (4) can also be characterized as solutions of

$$[\, f(t) \quad -1\,] \begin{bmatrix} \mathbf{u}^+(t) & \mathbf{v}^+(t) \\ \mathbf{p}^+(t) & \mathbf{q}^+(t) \end{bmatrix} = 0 \tag{5}$$

for some $\mathbf{p}^+(t) = \sum_{k=0}^{2m-1} p_k \mathbf{h}_{m-1+k}(t)$, $\mathbf{q}^+(t) = \sum_{k=0}^{2m-1} q_k \mathbf{h}_{m-1+k}(t)$ with normalization

$$\mathbf{e}_{m+1}^T\,[\,\mathbf{u} \;\; \mathbf{v}\,] = [\,1 \;\; 0\,], \quad p_0 = 0,\; q_0 = 1 .$$

The importance of Theorem 4.1 for practical calculation consists in the fact that CH-Bezoutians can be represented with the help of trigonometric transforms. Here we present only one of several possibilities. More relations can be deduced from the representations in [9].

Let $N \geq n$ be a power of 2 and $M = N/2$. We use the cosine transforms of third and fourth kind which are defined as multiplication by

$$\mathcal{C}_3 = \left[\cos\frac{(2i+1)j\pi}{2M} \right]_{i,j=0}^{M-1}, \quad \mathcal{C}_4 = \left[\cos\frac{(2i+1)(2j+1)\pi}{4M} \right]_{i,j=0}^{M-1},$$

respectively. For these transforms and their inverses $O(M \log M)$ complexity algorithms do exist (see, for example [15]).

We introduce the numbers

$$\rho_j = \cos\frac{j\pi}{2M}, \quad \sigma_j = \cos\frac{(2j+1)\pi}{4M}.$$

and the diagonal matrices

$$\Lambda_\rho = \mathrm{diag}\,(\rho_i)_{i=0}^{M-1} \quad \text{and} \quad \Lambda_\sigma = \mathrm{diag}\,(\sigma_i)_{j=0}^{M-1} .$$

Then we have for a vector $\mathbf{w} \in \mathbb{R}^M$

$$(\mathbf{w}^+(\rho_{2i+1}))_{i=0}^{M-1} = \Lambda_\sigma^{-1}\mathcal{C}_4\mathbf{w} \quad \text{and} \quad (\mathbf{w}^+(\rho_{2j}))_{j=0}^{M-1} = \Lambda_\rho^{-1}\mathcal{C}_3^T\mathbf{w}\,,$$

and for a $m \times m$ matrix B,

$$[\,B^+(\rho_{2i+1}, \rho_{2j})\,]_{i,j=0}^{M-1} = \Lambda_\sigma^{-1}\mathcal{C}_4 B \mathcal{C}_3 \Lambda_\rho^{-1} .$$

Hence formula (4) can be written as

$$\Lambda_\sigma^{-1}\mathcal{C}_4 R_m^{-1}\mathcal{C}_3\Lambda_\rho^{-1} = \frac{1}{2}\left[\frac{\mathbf{u}(\rho_{2i+1})\mathbf{v}(\rho_{2j}) - \mathbf{v}(\rho_{2i+1})\mathbf{u}(\rho_{2j})}{\rho_{2i+1}-\rho_{2j}}\right]_{i,j=0}^{M-1}$$

or

$$R_m^{-1} = \frac{1}{2}\,\mathcal{C}_4^{-1}\Lambda_\sigma(D_o(\mathbf{u})\Omega D_e(\mathbf{v}) - D_o(\mathbf{v})\Omega D_e(\mathbf{u}))\Lambda_\rho\mathcal{C}_3^{-1}\ ,$$

where $D_o(\mathbf{u}) = \operatorname{diag}(\mathbf{u}(\rho_{2i+1}))_{i=0}^{M-1}$, $D_e(\mathbf{u}) = \operatorname{diag}(\mathbf{u}(\rho_{2i}))_{i=0}^{M-1}$ (similarly for $D_o(\mathbf{v})$ and $D_e(\mathbf{v})$) and

$$\Omega = \left[\frac{1}{\rho_{2i+1}-\rho_{2j}}\right]_{i,j=0}^{M-1}.$$

The matrix Ω, in its turn, can be represented with the help of trigonometric transforms. One of the possibilities is (see [9]))

$$\Omega = -\widetilde{\Lambda}_\sigma^{-1}\mathcal{C}_4\mathcal{C}_3\widetilde{\Lambda}_\rho^{-1},$$

where

$$\widetilde{\Lambda}_\sigma = \operatorname{diag}\left((-1)^i\sigma_i\right)_{i=0}^{M-1},\quad \widetilde{\Lambda}_\rho = \operatorname{diag}\left((-1)^i\rho_i\right)_{i=0}^{M-1}\ .$$

In that way a vector can be multiplied by a CH-Bezoutian with the help of 6 transforms of length M plus $O(M)$ operations. For preprocessing the data, assuming that $\mathbf{u}$ and $\mathbf{v}$ are given, 4 transforms of length M are required.

5. INTERPOLATION INTERPRETATION

We translate now equation (5) into an interpolation problem.

Let ρ_j $(j = 1,\dots,N)$ be as in the previous section. It follows from trigonometric identities that

$$\mathbf{h}_{N+k}(\rho_j) = \mathbf{h}_{N-k-1}(\rho_j) = (-1)^j\mathbf{h}_k(\rho_j)\quad (k = 0,1,\dots)\ . \tag{6}$$

Let $\mathbf{p}^+(t)$ and $\mathbf{q}^+(t)$ be taken from the solution of (5). We form the vectors

$$\mathbf{r} = (\,p_{N-m-k} + p_{N-m+k+1}\,)_{k=0}^{N-m},\quad \mathbf{s} = (\,q_{N-m-k} + q_{N-m+k+1}\,)_{k=0}^{N-m}.$$

Then, in view of (6), the polynomials $\mathbf{r}^+(t)$ satisfy

$$\mathbf{r}^+(\rho_j) = (-1)^j\mathbf{p}^+(\rho_j),\quad \mathbf{s}^+(\rho_j) = (-1)^j\mathbf{q}^+(\rho_j)\quad (j = 0,\dots,N-1)\ .$$

Hence, taking in (5) the values at ρ_j, the polynomials $\mathbf{p}^+(t)$ and $\mathbf{q}^+(t)$ can be replaced by polynomials of degree $N-m$.

We form the matrix polynomial

$$\Phi_N(t) = \begin{bmatrix}\mathbf{u}^+(t) & \mathbf{v}^+(t)\\ \mathbf{r}^+(t) & \mathbf{s}^+(t)\end{bmatrix}. \tag{7}$$

If we take the highest order coefficients in each row of $\Phi_N(t)$ we obtain I_2. Let us call a matrix polynomial with this property *quasi-monic.*

Theorem 5.1. *Let $\mathbf{u}^+(t)$ and $\mathbf{v}^+(t)$ be the polynomials in Theorem 4.1. Then there exist $\mathbf{r},\mathbf{s}\in\mathbb{R}^{N-m+1}$ such that $\Phi_N(t)$ defined by (7) is quasi-monic and satisfies the tangential interpolation conditions*

$$[\,f(\rho_j)\ \ (-1)^{j+1}\,]\,\Phi_N(\rho_j) = [\,0\ \ 0\,]\quad (j = 0,\dots,N-1)\ . \tag{8}$$

6. DESCRIPTION OF THE ALGORITHM

We describe now an algorithm that solves (8) with $O(N\log^2 N)$ complexity. We assume, for simplicity, that n is already a power of 2 and $N = n$. The idea is to split (8) into 2 subproblems of half the size. First we find a quasi-monic matrix polynomial $\Phi_m(t)$ of the same form like $\Phi_n(t)$ the entries of which are of degree $\leq m$ and which satisfies the interpolation conditions

$$[\, f(\rho_{2j}) \quad -1\,]\, \Phi_m(\rho_{2j}) = [\, 0 \;\; 0\,] \quad (j = 0, \ldots, m-1)\,.$$

Then we compute the residues

$$[\, z_j^1 \;\; z_j^2\,] = [\, f(\rho_{2j+1}) \;\; 1\,]\, \Phi_m(\rho_{2j+1}) \quad (j = 0, \ldots, m-1)\,.$$

After that we find a quasi-monic matrix polynomial $\Psi_m(t)$ the entries of which are of degree $\leq m$ and which satisfies the interpolation conditions

$$[\, z_j^1 \;\; z_j^2\,]\, \Psi_m(\rho_{2j+1}) = [\, 0 \;\; 0\,] \quad (j = 0, \ldots, m-1)\,.$$

Using cosine transforms this problem can be reformulated as an interpolation problem at ρ_{2j}. Finally we compute

$$\Phi_n(t) = \Phi_m(t)\Psi_m(t)$$

using fast algorithms for the cosine transforms. The splitting is recursively repeated until a size is reached for which the $O(n^2)$ complexity algorithm from [10] is applied.

7. NUMERICAL EXPERIMENTS

We consider double precision symmetric Toeplitz matrices T_n whose entries are real and random uniformly distributed in $[0, 1]$ with $n = 2^k$ for $k = 1, \ldots, 15$. Interpolation problems of size less than or equal to 2^7 are solved by our fast-only algorithm and for each value of k we consider 10 samples.

The right-hand sides $b_n \in \mathbb{R}^n$ of the samples are calculated such that $x_n := T_n^{-1} b_n = [1 \;\cdots\; 1]^T$. The calculations were done on a Pentium IV running at 2.66GHz having 1GByte of RAM. Our software is written in C++ making use of the FFTW 3.0 package and the ATLAS package. It is available at `http://www.cs.kuleuven.ac.be/~marc/`.

Figure 1 shows the relative norm of the residual using the fast-only and superfast algorithm without any kind of refinement. For a dimension bigger than 2^{12}, the relative norm of the residual becomes of the order 1. For dimensions smaller than 2^{12}, we can decrease the relative residual norm by applying some iterative refinement steps at the Toeplitz level, i.e., when the generators of the inverse of the CH-Bezoutian are computed and so they can be used to refine the final soulution with fast Bezoutian multiplications requiring $O(n \log n)$ floating point operations. When using three of these steps, the results are given in Figure 2.

Figure 3 shows the execution time (in seconds) of the solution computed using the fast interpolation algorithm (symbol $+$) and the superfast with (symbol $*$) and without iterative refinement (symbol $\circ$) at the Toeplitz level.

In Figure 1, we have seen that for dimension larger than 2^{12} the solution to the interpolation problem should be computed with a smaller norm of the residual enabling the iterative refinement at the Toeplitz level to work with success. To derive a solution of the interpolation

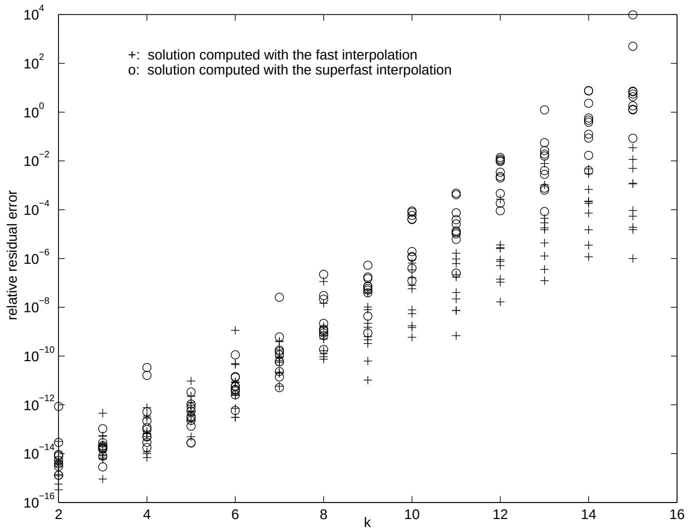

Figure 1. $\frac{\|b_n - T_n \hat{x}_n\|_1}{\|b_n\|_1}$ versus $k = \log_2 n$ for $k = 1, \ldots, 15$.

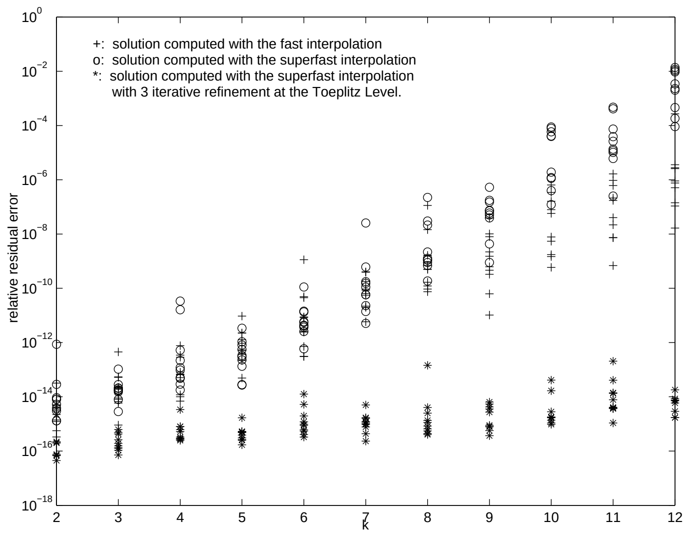

Figure 2. $\frac{\|b_n - T_n \hat{x}_n\|_1}{\|x_n\|_1}$ versus $k = \log_2 n$ for $k = 1, \ldots, 12$.

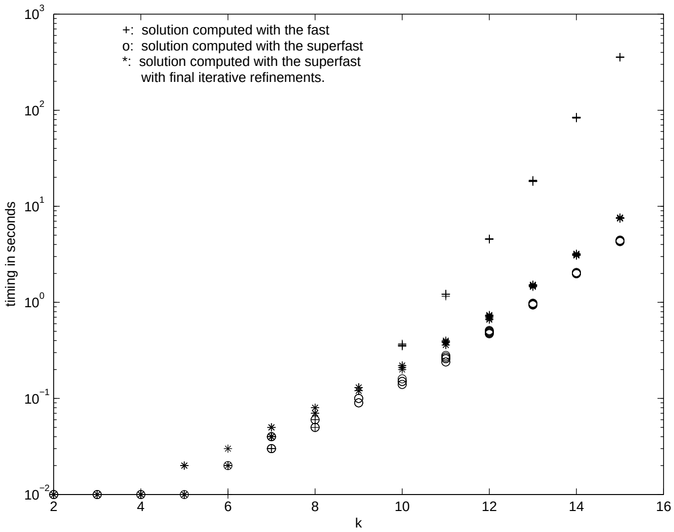

Figure 3. Timings of the fast and the superfast applying iterative refinements at the Toeplitz level.

problem having a smaller residual norm, we followed several strategies. We tried iterative refinement at intermediate levels of the interpolation problem using inversion formulas for coupled Vandermonde matrices (see [7]). However, it turned out that these inversion formulas could not be evaluated in a numerically stable way. Therefore, we developed a slower but effective refinement technique. In theory, the solution $\Phi_N(t)$ of the interpolation problem has to satisfy the following equations

$$[\, f(\rho_j) \;\; (-1)^{j+1} \,]\, \Phi_N(\rho_j) = [\, 0 \;\; 0 \,] \quad (j = 0, \ldots, N-1) \; .$$

In practice we do not get the zero vector at the right-hand-side but a certain residual vector $[r_{1,j} \qquad r_{2,j}]$. Hence, when we can compute $\tilde{\Phi}_N(t)$ satisfying

$$[\, f(\rho_j) \;\; (-1)^{j+1} \;\; r_{1,j} \;\; r_{2,j} \,] \begin{bmatrix} \tilde{\Phi}_N(\rho_j) \\ I_2 \end{bmatrix} = [\, 0 \;\; 0 \,] \quad (j = 0, \ldots, N-1) \qquad (9)$$

with $\deg \tilde{\Phi}_N(t) < \deg \Phi_N(t)$, we get the solution of the interpolation problem as $\Phi_N(t) + \tilde{\Phi}_N(t)$.

In practice, the right-hand-side of 9 will also not be the zero vector but it turns out that its norm is smaller than the norm of $[\, r_{1,j} \;\; r_{2,j} \,]$ under weak conditions. The computation of $\tilde{\Phi}_N(t)$ requires again the divide and conquer approach, now for a 4×4 polynomial matrix instead of a 2×2 one. This refinement approach can be applied at any level of the interpolation problem. However, it is clear that it is an expensive refinement procedure. To decide at which point in the recursive procedure, we had to apply this refinement procedure, we used a heuristic based on the size of the interpolation problem at that point as well as on the norm of the corresponding residuals. As is shown in Figure 4, it is an expensive strategy. So, research is still needed to find a better heuristic and/or an inversion formula that is numerically stable to apply for coupled Vandermonde matrices based on Chebyshev polynomials. The relative residual norm before applying iterative refinement at the Toeplitz level is now much better as is shown in Figure 5.

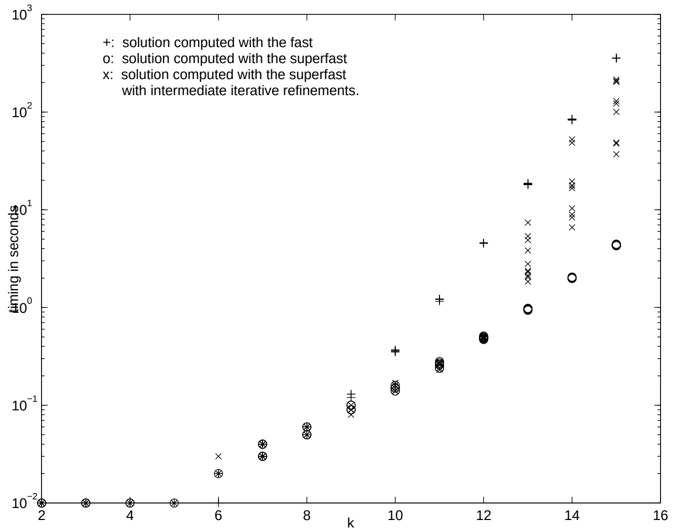

Figure 4. Timings of the fast and the superfast applying iterative refinements at each level except at the Toeplitz one.

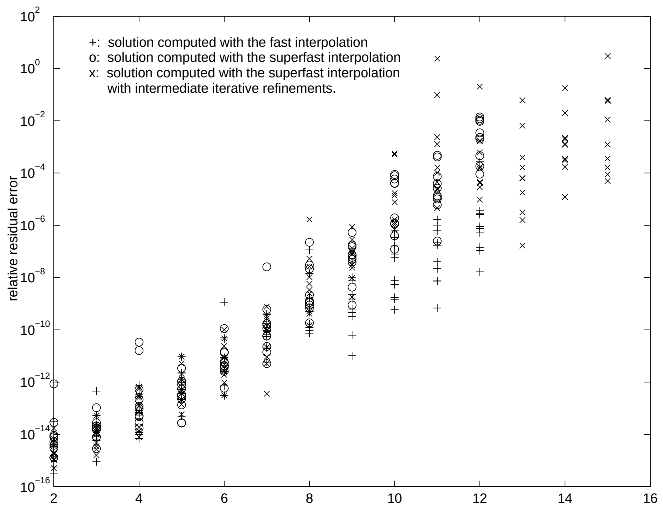

Figure 5. Residual of the fast and the superfast applying iterative refinements at each level except at the Toeplitz one.

8. CONCLUSIONS

In this paper we developed a superfast algorithm to solve real symmetric Toeplitz systems using real fast trigonometric transformations. The numerical experiments indicate that our method is indeed superfast for matrices up to size 2^{12} of that specific class. However, for bigger dimensions an efficient way to refine the solution at intermediate levels when solving the interpolation problem is still missing.

REFERENCES

1. G.S. Ammar, W.B. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM J. Matrix Analysis Appl.* 9: 61-76 (1988).
2. R.R. Bitmead, B.D.O. Anderson, Asymptotically fast solution of Toeplitz and related systems of linear equations, *Linear Algebra Appl.* 34: 103-116 (1980).
3. R. Brent, F. Gustavson, D. Yun, Fast solution of Toeplitz systems of equations and and computation of Padé approximants, *J.Algorithms* 1: 259-295 (1980).
4. F. de Hoog, A new algorithm for solving Toeplitz systems of equations, *Linear Algebra Appl.* 88/89: 123-138 (1987).
5. G. Heinig Transformation Approaches for Fast and Stable Solution of Toeplitz Systems and Polynomial Equations, *Proceedings of RAAM Workshop*, Kuwait University, May 1996, 223–238.
6. G. Heinig, Chebyshev-Hankel matrices and the splitting approach for centrosymmetric Toeplitz-plus-Hankel matrices, *Linear Algebra Appl.* **327**, 1-3 (2001), 181–196.
7. G. Heinig, F. Al-Musallam, Lagrange's formula for tangential interpolation with application to structured matrices, *Integral Equ. and Operator Theory*, 30 (1998), 83-100.
8. G. Heinig, P. Jankowski, Fast algorithms for the solution of general Toeplitz systems, *Wiss. Zeitschr. TU Karl-Marx-Stadt*, **32**, 1 (1990), 12–16.
9. G. Heinig, K. Rost, Representations of Cauchy matrices with Chebyshev nodes using trigonometric transforms, In: D. Bini, E. Tyrtyshnikov, P. Yalamov (Eds.), *Structured Matrices: Recent Developments in Theory and Computation*, Advances in Computation: Theory and Practice, vol.4, Nova Svience Inc., 2001, pp. 135–148.
10. P. Kravanja, M. Van Barel, A fast Hankel solver based on an inversion formula for block Loewner matrices, *Calcolo*, 33 (1996), 147–164.
11. J.C. Mason, Chebyshev polynomials of the second, third and fourth kind in approximation, indefinite integration, and integral transforms, *J.of Comp.and Appl.Math.* 49 (1993), 169-178.
12. M. Morf, Doubling algorithms for Toeplitz and related equations, *IEEE Internat.Conf. on Acoustics, Speech and Signal Processing, Denver 1980*, pp.954-959.
13. V. Y. Pan, *Structured Matrices and Polynomials*, Birkhäuser Verlag, Boston and Springer–Verlag, New York, 2001.
14. M. Stewart, A superfast Toeplitz solver with improved numerical stability, *SIAM J. Matrix Anal. Appl.*, 25 (3) 2003, 669-693.
15. M. Tasche, Fast algorithms for discrete Chebyshev-Vandermonde transforms and applications, *Numerical Algorithms* 5: 453-464 (1993).
16. M. Van Barel, G. Heinig, P. Kravanja, A stabilized superfast solver for nonsymmetric Toeplitz systems, *SIAM J. Matrix Anal. Appl.*, 23 (2) (2001), 449–510.