Digital Signal Processing 20 (2010) 1711–1722

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Trading-off matrix size and matrix structure: Handling Toeplitz equations by embedding on a larger circulant set ☆

Paulo J.S.G. Ferreira [a], María Elena Domínguez [b,*]

[a] Dept. Electrónica, Telecomunicações e Informática /IEETA, Universidade de Aveiro, Portugal
[b] Dept. Matemática Aplicada, ETSII, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Available online 27 March 2010

Keywords:
Toeplitz matrices
Circulant matrices
Embeddings
Linear equations
FFT

ABSTRACT

This paper explores a seemingly counter-intuitive idea: the possibility of accelerating the solution of certain linear equations by adding even more equations to the problem. The basic insight is to trade-off problem size by problem structure. We test this idea on Toeplitz equations, in which case the expense of a larger set of equations easily leads to circulant structure. The idea leads to a very simple iterative algorithm, which works for a certain class of Toeplitz matrices, each iteration requiring only two circular convolutions. In the symmetric definite case, numerical experiments show that the method can compete with the preconditioned conjugate gradient method (PCG), which achieves $O(n \log n)$ performance. Because the iteration does not converge for all Toeplitz matrices, we give necessary and sufficient conditions to ensure convergence (for not necessarily symmetric matrices), and suggest an efficient convergence test. In the positive definite case we determine the value of the free parameter of the circulant that leads to the fastest convergence, as well as the corresponding value for the spectral radius of the iteration matrix. Although the usefulness of the proposed iteration is limited in the case of ill-conditioned matrices, we believe that the results show that the problem size/problem structure trade-off deserves further study.

1. Introduction

There are many problems that lead to Toeplitz equations, and many methods for solving them. Levinson's algorithm solves n Toeplitz equations in $O(n^2)$ flops [1]. Even the inverse of a Toeplitz can be computed in $O(n^2)$ flops, using the Trench method [1]. Many variants of these methods have been introduced [2], including a number of methods that can handle Toeplitz matrices with singular submatrices [3,4]. Efficient algorithms suited for rationally generated matrices have also been given [5,6]. Quadratic inversion formulas for matrices "close" to Toeplitz have been given [7,8]. A more general approach, described in [9], has lead to $O(n^2)$ algorithms for a class of matrices with recursive structure. This includes Toeplitz, Hankel and other matrices. Block–Toeplitz matrices with Toeplitz entries are also studied in [10]. The fast solver proposed in [11], based on a modified QR algorithm, is backward stable and can be applied to a class of nonsymmetric Toeplitz-like matrices. It also explores displacement structure (see [12] for a review).

There are also $O(n \log^2 n)$ methods. The one in [13] applies to matrices of low displacement rank, the method in [14] is based on Padé approximation, and Ref. [15] uses a generalized Schur algorithm and applies to real positive definite Toeplitz

☆ Some of the results contained in this work were presented at the 109th Annual Meeting of the American Mathematical Society, Baltimore, MD, January 2003, and in the SampTA-03, Strobl, Austria, May 2003.

* Corresponding author. Present address: C/ José Gutiérrez Abascal 2, 28006 Madrid, Spain.
E-mail addresses: pjf@ua.pt (P.J.S.G. Ferreira), edominguez@etsii.upm.es (M.E. Domínguez).

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2010.03.002

matrices. The method described in [16] uses real trigonometric transformations and solves symmetric Toeplitz equations with complexity $O(n\log^2 n)$. Iterative approaches based on Newton's method [17,18] have also been reported.

Hermitian positive definite Toeplitz equations can be solved using the preconditioned conjugate gradient (PCG) method [19,20], implementing the matrix-vector products using the FFT. Adequate preconditioning [21–26] may lead to $O(n\log n)$ performance.

This paper introduces a new idea for solving linear equations: the idea of embedding the original matrix in a larger but more structured matrix. The size of the problem increases, but the more regular structure may allow a net performance gain. We test the concept on Toeplitz equations, and show that the embedding of the Toeplitz matrix in a circulant leads to an efficient and extremely simple iterative algorithm, that can under certain conditions compete with PCG.

In Section 2 we recall some theoretical facts regarding circulant and Toeplitz matrices, that will be used in the rest of the paper. The proposed method is described in Section 3, and its convergence is analyzed in Section 4. Section 5 is concerned with the very important problem of positive definite Toeplitz systems. For these matrices, we provide several theorems which justify the convergence of the algorithm. Finally, some experimental results are shown.

Although the usefulness of the proposed iteration is limited in the case of ill-conditioned matrices, we believe that the results show that the problem size/problem structure trade-off has potential, and deserves further study.

2. Preliminaries

Let us recall some results about Toeplitz and circulant matrices.

If C is a circulant matrix of order $2n$, it is known that its eigenvalues are $\{\lambda_i(C)\}_{0\leqslant i<2n}$, where $\lambda_i(C)$ is the ith element of the discrete Fourier transform (DFT) of the first row of C. On one hand, C can be written as

$$C=\begin{bmatrix} T & S \\ S & T \end{bmatrix}$$

where T, S are Toeplitz matrices. Note that $T+S$ is a circulant, and $T-S$ is skew-circulant. By expressing the DFT of length $2n$ as two DFTs of length n, it is shown in [27] that the eigenvalues of $T+S$ are $\{\lambda_{2i}(C)\}_{0\leqslant i<n}$, and those of $T-S$ are $\{\lambda_{2i+1}(C)\}_{0\leqslant i<n}$.

On the other hand, we can also write

$$C=\frac{1}{2}\begin{bmatrix} I & I \\ I & -I \end{bmatrix}\begin{bmatrix} T+S & 0 \\ 0 & T-S \end{bmatrix}\begin{bmatrix} I & I \\ I & -I \end{bmatrix} \tag{1}$$

so C is invertible if and only if $T\pm S$ are nonsingular. In that case, C^{-1} is equal to

$$\frac{1}{2}\begin{bmatrix} I & I \\ I & -I \end{bmatrix}\begin{bmatrix} (T+S)^{-1} & 0 \\ 0 & (T-S)^{-1} \end{bmatrix}\begin{bmatrix} I & I \\ I & -I \end{bmatrix} \tag{2}$$

which is also a circulant matrix. Hence, multiplications by C or C^{-1} are computed as circular convolutions, which can be efficiently implemented using the FFT algorithm.

For real symmetric matrices C, note that C is positive definite if and only if the real symmetric Toeplitz matrices $T\pm S$ are positive definite.

3. The method

Consider the Toeplitz problem $Tx=y$, where T is a real nonsingular $n\times n$ Toeplitz matrix with elements $T_{ij}=t_{i-j}$. It is easy to realize that T can be embedded (as a principal submatrix) in a circulant matrix of size $2n-1$ or greater. We will often consider circulants of size $2n$, but larger sizes have practical and possibly theoretical advantages (the smallest power of two greater than $2n-1$ is often a good size to use in practice, as FFTs of powers of two are especially efficient).

To build the circulant, define a Toeplitz matrix S, with elements $S_{ij}=s_{i-j}$ determined by $s_i=t_{i-n}$, for $i>0$, and $s_i=t_{i+n}$, for $i<0$. The element $\alpha=s_0$ is arbitrary. Then,

$$C=\begin{bmatrix} T & S \\ S & T \end{bmatrix}$$

is a circulant of size $2n$.

In any case, the Toeplitz linear system can now be replaced with a circulant problem:

$$C\begin{bmatrix} x \\ 0 \end{bmatrix}=\begin{bmatrix} T & S \\ S & T \end{bmatrix}\begin{bmatrix} x \\ 0 \end{bmatrix}=\begin{bmatrix} y \\ z \end{bmatrix}$$

where $y=Tx$ and $z=Sx$. The circulant equations $Ca=b$, with

$$a=\begin{bmatrix} x \\ 0 \end{bmatrix},\qquad b=\begin{bmatrix} y \\ z \end{bmatrix}$$

are equivalent to $Tx = y$. Note the difference between this formulation and the usual one: the left-hand side vector of a traditionally written set of linear equations $Tx = y$ contains all the unknown quantities, whereas in our formulation there are unknown elements in both the right-hand and left-hand side vectors. We may say that both a and b are partially known, since x and z are unknown.

The proposed algorithm produces two sequences a_k and b_k which iteratively approximate a and b, respectively. At each step, the known elements of a and b are reinserted in the approximations, and multiplication by C or C^{-1} leads to the next approximation. More precisely, let

$$a_k = \begin{bmatrix} x_k \\ o_k \end{bmatrix}, \qquad b_k = \begin{bmatrix} y_k \\ z_k \end{bmatrix} \tag{3}$$

be the approximations to a and b, and hence to the unknowns x and z, available at the end of step k. Step $k+1$ is defined by the following sequence of operations:

(1) Insert the known elements of b into b_k, that is, replace y_k by y and compute the matrix-vector product:

$$C^{-1} \begin{bmatrix} y \\ z_k \end{bmatrix} = \begin{bmatrix} x_{k+1} \\ o_{k+1} \end{bmatrix} = a_{k+1}. \tag{4}$$

(2) Insert the known elements of a into a_{k+1}, that is, replace o_{k+1} by the zero vector, and compute the matrix-vector product:

$$C \begin{bmatrix} x_{k+1} \\ 0 \end{bmatrix} = \begin{bmatrix} y_{k+1} \\ z_{k+1} \end{bmatrix} = b_{k+1}. \tag{5}$$

(3) Stop if the result is acceptable. Otherwise repeat.

The matrix-vector products are $2n$-dimensional circular convolutions that can be performed in $O(n \log n)$ flops using standard FFT or fast convolution algorithms. The FFT of the first column of C needs to be computed only once, and immediately determines the FFT of the first column of C^{-1}. Both steps (1) and (2) of the algorithm require one FFT and one IFFT of (at least) $2n$ points, and hence $O(n \log n)$ flops. More specific speedups are possible but will not be discussed for brevity (for example, it is well known that the bit-reversal stages can be avoided by properly selecting the FFT and IFFT algorithms, and that the FFT of $2n$ real data can be computed using one n-point complex FFT).

It is interesting to compare our work and [23,22], which explore preconditioning by embedding. These are based on equations similar to

$$\begin{bmatrix} T & S \\ S & T \end{bmatrix} \begin{bmatrix} x \\ x \end{bmatrix} = \begin{bmatrix} y \\ y \end{bmatrix}$$

of size $m = 2n$. They are not equivalent to $Tx = y$ but to $(T + S)x = b$, and lead to the (circulant) preconditioner $T + S$. This is the same as our $T + S$ when the size of the embedding is $m = 2n$. Another (skew-circulant) preconditioner obtained by embedding is $T - S$ (note its appearance in (1)).

It is important to note that the our approach can in principle be tried even when T is not Toeplitz, provided that it can be embedded as a principal submatrix in a circulant of sufficiently small size. Note that principal submatrices of Toeplitz or circulant matrices are not necessarily Toeplitz. For example,

$$\begin{bmatrix} a & b & d \\ d & a & c \\ b & c & a \end{bmatrix}$$

is not Toeplitz, but it can be embedded as a principal submatrix in the circulant

$$\begin{bmatrix} a & b & \mathbf{c} & d \\ d & a & \mathbf{b} & c \\ \mathbf{c} & \mathbf{d} & \mathbf{a} & \mathbf{b} \\ b & c & \mathbf{d} & a \end{bmatrix}$$

(note the row and column that must be added, in bold). For an example of a problem that leads to such non-Toeplitz matrices see [28,29]. In fact, the circulant matrices considered in [28] have one interesting property: they have k zero eigenvalues, but every principal submatrix obtained by deleting at least k of its rows and columns is nonsingular.

Since the proposed method only works if C is invertible, this suggests the following question: is it always possible to embed T in a nonsingular circulant, by the described process? The following result shows that this is possible. We denote by C_0 the circulant embedding matrix obtained by choosing $\alpha = 0$.

Theorem 1. *A Toeplitz matrix T can always be embedded in an infinite set of nonsingular circulant matrices C: the ones obtained by choosing*

$$\alpha \notin \{-\lambda_{2i}(C_0),\ \lambda_{2i+1}(C_0),\ 0 \leqslant i < n\}.$$

Proof. Given T, let us build as explained C_0 and S_0 such that $\alpha = s_0 = 0$. Hence, any other matrix S of that kind is $S = S_0 + \alpha I$. Hence, $T \pm S = T \pm (S_0 + \alpha I)$. Now, C is invertible if and only if its eigenvalues are nonzero, but its eigenvalues are those of $T + S_0 + \alpha I$ (say, $\lambda_{2i}(C_0) + \alpha$), and those of $T - S_0 - \alpha I$ (say, $\lambda_{2i+1}(C_0) - \alpha$). It suffices to impose that these $2n$ numbers are not zero. □

4. Analysis of the algorithm

This section discusses the convergence of the algorithm, in the general case of invertible matrices T. The first theorem provides a necessary and sufficient condition for convergence.

Theorem 2. *If C is invertible, $(T+S)^{-1}$ exists, and the algorithm converges if and only if the eigenvalues λ of $A = (T+S)^{-1}(T-S)$ lie in the subset of the complex plane described by*

$$4\cos\theta - 2\sqrt{2} < |\lambda + 1| < 4\cos\theta + 2\sqrt{2} \tag{6}$$

where θ denotes the polar angle of λ, centered in $\lambda = -1$.

The complex domain defined by (6) is named Pascal's snail, and is depicted in Fig. 1.

Proof. It follows from (3), (4) and (5) that

$$a_{k+1} = C^{-1} Q C P a_k + C^{-1} \begin{bmatrix} y \\ 0 \end{bmatrix}$$

where

$$P = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \qquad Q = I - P.$$

It will be convenient to set

$$C^{-1} = \begin{bmatrix} C_1 & C_2 \\ C_2 & C_1 \end{bmatrix}. \tag{7}$$

A brief computation now shows that

$$C^{-1} Q C P = \begin{bmatrix} C_2 S & 0 \\ C_1 S & 0 \end{bmatrix},$$

and therefore x_k converges if and only if the spectral radius of $C_2 S$ satisfies $\rho(C_2 S) < 1$. To determine C_2, it suffices to compare Eqs. (7) and (2), and a straightforward computation shows that

$$C_2 = \frac{1}{2}\big[(T+S)^{-1} - (T-S)^{-1}\big].$$

The product $C_2 S$ can now be written as

$$\begin{aligned} C_2 S &= \frac{1}{4}\big[(T+S)^{-1} - (T-S)^{-1}\big]\big[(T+S) - (T-S)\big] \\ &= \frac{1}{4}\big(2I - A - A^{-1}\big), \end{aligned}$$

with

$$A = (T+S)^{-1}(T-S).$$

Any eigenvalue λ of A–they are not zero–corresponds to an eigenvalue $f(\lambda)$ of the iteration matrix, with

$$f(\lambda) = \frac{1}{4}\big(2 - \lambda - \lambda^{-1}\big),$$

and so the algorithm converges if and only if $|f(\lambda)| < 1$. To finish, we find the set of complex numbers λ such that

$$\left|\lambda+\lambda^{-1}-2\right|<4 \quad\Leftrightarrow\quad \left|\lambda^2+1-2\lambda\right|<4|\lambda| \quad\Leftrightarrow\quad |\lambda-1|^2<4|\lambda|.$$

By writing $\lambda=-1+re^{i\theta}$, this is equivalent to

$$\left|re^{i\theta}-2\right|^2<4\left|re^{i\theta}-1\right| \quad\Leftrightarrow\quad r^2+4-4r\cos\theta<4\sqrt{r^2+1-2r\cos\theta};$$

we square this expression, and arrive to the inequality

$$r^2-8r\cos\theta+8\left(2\cos^2\theta-1\right)<0$$

whose quadratic polynomial has roots $4\cos\theta\pm2\sqrt{2}$, so the final solution is

$$4\cos\theta-2\sqrt{2}<r<4\cos\theta+2\sqrt{2}$$

where $r=|\lambda+1|$. □

Remark. The iteration matrix is $C_2S=(2I-A-A^{-1})/4$, not A. The theorem assures the possibility of studying the eigenvalues of the simpler matrix A, for convergence purposes.

5. Analysis for positive definite matrices

In many practical problems T is a positive definite Toeplitz matrix, that is, $x^*Tx>0$ for all nonzero complex vectors x, a condition that implies that T is Hermitian [30]. In this case, the first question that arises is whether T can embedded in a circulant positive definite matrix C.

To this aim, let us recall the circulant matrix C_0; we will denote

$$\begin{aligned} L_0&=\min_{0\leqslant i<n}\lambda_{2i}(C_0), \qquad L_{2n-2}=\max_{0\leqslant i<n}\lambda_{2i}(C_0),\\ L_1&=\min_{0\leqslant i<n}\lambda_{2i+1}(C_0), \qquad L_{2n-1}=\max_{0\leqslant i<n}\lambda_{2i+1}(C_0) \end{aligned} \tag{8}$$

which correspond to the minimum and maximum eigenvalues of $T+S_0$ and $T-S_0$, respectively. Note that they are easily computed by means of a $2n$-point FFT of the first row of C_0, and that this FFT has to be computed for the purpose of efficiently implementing the circular convolutions required by the algorithm. Therefore, assuming that these four quantities are known implies negligible computational overhead.

We first provide the theorem of existence in the positive definite case, analogous to Theorem 1 in the general case:

Theorem 3. *A real positive definite Toeplitz matrix T can be embedded in a real positive definite circulant matrix C by the described process if and only if*

$$L_0+L_1>0.$$

Moreover, C is positive definite if and only if α is chosen in the interval

$$\alpha\in(-L_0,L_1). \tag{9}$$

Proof. Recalling the preliminaries and the proof of Theorem 1, we now impose that the eigenvalues of $T\pm S=T\pm(S_0+\alpha I)$ be positive. But the eigenvalues of $T+S_0+\alpha I$ are $\lambda_{2i}(C_0)+\alpha$, and those of $T-S_0-\alpha I$ are $\lambda_{2i+1}(C_0)-\alpha$. It suffices to impose that these $2n$ numbers be positive. As C_0 is real and symmetric, its eigenvalues $\lambda_i(C_0)$ are real; so we just choose

$$-\lambda_{2i}(C_0)<\alpha<\lambda_{2k+1}(C_0), \quad 0\leqslant i,k<n,$$

or, in an equivalent way,

$$-\min_{0\leqslant i<n}\lambda_{2i}(C_0)=-L_0<\alpha<\min_{0\leqslant k<n}\lambda_{2k+1}(C_0)=L_1$$

which concludes the proof. □

Next, we apply the general convergence theorem to the particular case of positive definite matrices:

Theorem 4. *If T is a positive definite Toeplitz matrix and $\alpha\in(-L_0,L_1)$, then the eigenvalues of $A=(T+S)^{-1}(T-S)$ are positive. In this case, the algorithm converges if and only if they belong to the interval $(1/c,c)$, where $c=3+2\sqrt{2}$ (hence, $1/c=3-2\sqrt{2}$).*

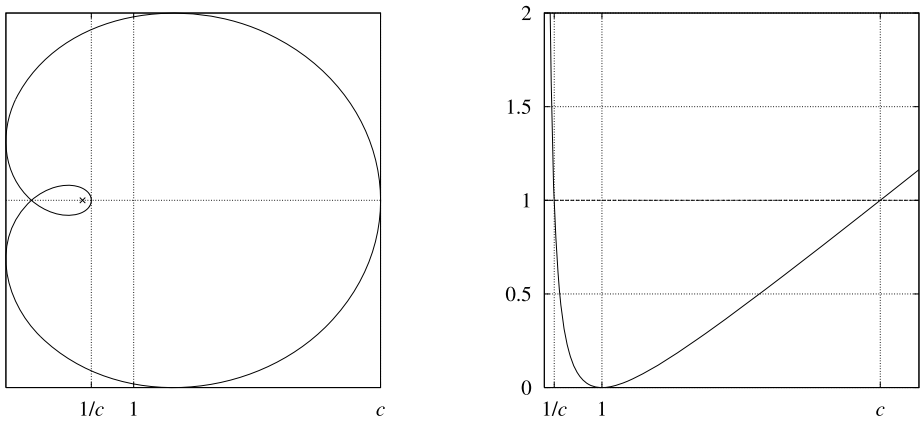

Fig. 1. The curve $|f(z)| = 1$ in the complex plane, and the graph of $|f(z)|$ for real z (as in Theorem 4, $c = 3 + 2\sqrt{2}$).

Proof. The previous theorem guarantees that C is positive definite; hence, it follows from (1) that both $T + S$ and $T - S$ are also positive definite matrices, and so is $(T + S)^{-1}$. We can therefore write $(T + S)^{-1} = XX^H$, and hence $A = XX^H(T - S) = XX^H(T - S)XX^{-1}$. This shows that A is similar to $B = X^H(T - S)X$ (henceforth, they share eigenvalues). Moreover, B is congruent to $T - S$, but $T - S$ is also positive definite, and so is B; thus, A has positive eigenvalues.

Now let us impose the convergence condition (6) to the eigenvalues λ of A. We now know that they are positive, so $|\lambda + 1| = \lambda + 1$, and their polar angle is $\theta = 0$; the new necessary and sufficient condition is

$$c^{-1} = 3 - 2\sqrt{2} < \lambda < 3 + 2\sqrt{2} = c$$

which concludes the proof. □

The behavior of the function $f(x)$ that maps eigenvalues of A to eigenvalues of the iteration matrix, and the rôle of the interval $(1/c, c)$, are depicted in Fig. 1.

Next, we will determine the values of α for which the method converges. To this aim, we provide a result concerning the eigenvalues of matrix A as a function of α.

Theorem 5. *For $\alpha \in (-L_0, L_1)$, all the eigenvalues of A are contained in the positive interval*

$$\left(\frac{L_1 - \alpha}{L_{2n-2} + \alpha}, \frac{L_{2n-1} - \alpha}{L_0 + \alpha}\right).$$

Proof. On one hand, Theorem 4 guarantees that A has positive eigenvalues; consequently,

$$\lambda_{\min}(A) = \frac{1}{\lambda_{\max}(A^{-1})}.$$

On the other hand, Theorem 3 assures that C is positive definite. By applying standard results to $A = (T + S)^{-1}(T - S)$, we get

$$\lambda_{\max}(A) \leqslant \frac{\max_i \lambda_{2i+1}(C)}{\min_i \lambda_{2i}(C)} = \frac{L_{2n-1} - \alpha}{L_0 + \alpha}. \tag{10}$$

Similarly, considering $A^{-1} = (T - S)^{-1}(T + S)$ rather than A, yields

$$\lambda_{\min}(A) \geqslant \frac{\min_i \lambda_{2i+1}(C)}{\max_i \lambda_{2i}(C)} = \frac{L_1 - \alpha}{L_{2n-2} + \alpha} > 0. \tag{11}$$

Together, (10) and (11) imply to the result. □

From Theorems 4 and 5 we deduce a very important sufficient condition of convergence:

Theorem 6. *Let T be a positive definite Toeplitz matrix. If*

$$\frac{L_{2n-1} + L_{2n-2}}{L_0 + L_1} < c, \tag{12}$$

then the algorithm converges for any α in the convergence interval

$$I_{con} = \left(\frac{L_{2n-1} - cL_0}{c+1}, \frac{cL_1 - L_{2n-2}}{c+1} \right).$$

Proof. The idea is to impose that the eigenvalues of A be in $(1/c, c)$. As they belong to the interval given in the previous theorem, it suffices to embed that interval into $(1/c, c)$:

$$\frac{1}{c} < \frac{L_1 - \alpha}{L_{2n-2} + \alpha}, \qquad \frac{L_{2n-1} - \alpha}{L_0 + \alpha} < c.$$

We obtain α from these inequalities:

$$\frac{L_{2n-1} - cL_0}{c+1} < \alpha < \frac{cL_1 - L_{2n-2}}{c+1}$$

which provide the sufficient interval of convergence I_{con}. To finish, I_{con} is nonempty whenever its length is > 0; and this condition is equivalent to (12). □

This sufficient condition is used to check the convergence of the algorithm: it suffices to compute the quantity

$$d = \frac{L_{2n-1} + L_{2n-2}}{L_0 + L_1}, \tag{13}$$

and see whether it is smaller than c or not (Eq. (12)).

Note that the computation of d implies negligible overhead, because the L_j are eigenvalues of C_0, and these eigenvalues, as we have noted before, are precisely the DFT elements needed to efficiently implement the circular convolutions.

If this convergence check is fulfilled, then there are infinitely many ways of choosing the value α. But which one yields the best convergence rate of the algorithm? The answer to this question is in our final result:

Theorem 7. *If Eq. (12) is satisfied, then the best possible convergence rate is achieved by choosing*

$$\alpha_{best} = \frac{L_1 L_{2n-1} - L_0 L_{2n-2}}{L_0 + L_1 + L_{2n-2} + L_{2n-1}}.$$

Moreover, the best spectral radius of the iteration matrix (obtained when $\alpha = \alpha_{best}$) is

$$\rho_{best} \leqslant \frac{(d-1)^2}{4d} < 1$$

where d is defined as in Eq. (13).

Proof. The iteration matrix of the algorithm is $C_2 S$; our aim is to find the value of the parameter α which minimizes its spectral radius. For any α in the convergence interval I_{con}, Theorem 5 states that eigenvalues λ of A are real and belong to the interval

$$\big(c_1(\alpha), c_2(\alpha)\big) = \left(\frac{L_1 - \alpha}{L_{2n-2} + \alpha}, \frac{L_{2n-1} - \alpha}{L_0 + \alpha} \right) \subset \left(\frac{1}{c}, c \right).$$

Recall that the spectral radius is

$$\rho(C_2 S) = \max\big\{ \big| f(\lambda) \big|,\ \lambda \text{ eigenvalue of } A \big\}$$

and this maximum should be minimized. Notice in Fig. 1 that $|f|$ is convex in $(1/c, c)$, so it reaches its maximum at the edges of each interval. Hence,

$$\rho(C_2 S) \leqslant \max\big\{ \big| f\big(c_1(\alpha)\big) \big|, \big| f\big(c_2(\alpha)\big) \big| \big\}.$$

Observing the growth of $|f|$, we get that this maximum is $|f(c_1(\alpha))|$ if $c_1(\alpha) c_2(\alpha) \leqslant 1$, and $|f(c_2(\alpha))|$ otherwise. This maximum is minimized when $c_1(\alpha) c_2(\alpha) = 1$. Direct manipulations show that the only value of the parameter which verifies this condition is $\alpha = \alpha_{best}$. Moreover, it turns out that $c_2(\alpha_{best}) = d \in [1, c)$ so finally

$$\rho(C_2 S) \leqslant \big| f\big(c_2(\alpha_{best})\big) \big| = \big| f(d) \big| = \frac{|d-1|^2}{4|d|}$$

which is smaller than 1, because d belongs to Pascal's snail. □

Remark. By choosing this parameter, the relative error between two consecutive iterations of the algorithm turns out to be

$$\|x_k - x\| = \|(C_2 S)(x_{k-1} - x)\| \leqslant \rho_{best}\|x_{k-1} - x\|$$

so after k steps we have

$$\|x_k - x\| \leqslant (\rho_{best})^k \|x_0 - x\| \tag{14}$$

where

$$\rho_{best} \leqslant \frac{(d-1)^2}{4d} < 1. \tag{15}$$

This confirms that, regardless the initial guess vector x_0, the algorithm always converges to the solution x, and gives an expression for the rate of convergence.

5.1. An additional result about the convergence

As we have pointed out, the proposed algorithm may not always converge because the eigenvalues of the iteration matrix $C_2 S$ may have modulus greater than 1. For instance, for ill-conditioned matrices T, the sufficient condition for convergence (12) is no longer fulfilled, and $\rho(C_2 S) \geqslant 1$; in other words, for *some* initial guess vectors, the iterative algorithm may not converge.

Nevertheless, not everything is lost in that case; here we give an additional general result about the convergence of this method:

Theorem 8. *For positive definite Toeplitz matrices T of large enough size n, most of the eigenvalues of the iteration matrix $C_2 S$ are clustered around* 0.

Proof. By using the expression of C^{-1} given by Eq. (7), and the fact that $C^{-1}C = I$, we get $C_2 S = I - C_1 T$, so its eigenvalues λ are

$$\lambda(C_2 S) = 1 - \lambda(C_1 T).$$

On the other hand, for positive definite Toeplitz matrices T, [24] guarantees that the eigenvalues of $C_1 T$ are clustered around 1, except for some outliers. Moreover, in [26] it is proven that they are even more clustered around 1 than the eigenvalues of matrix $(T+S)^{-1}T$, which appears in the approach described in [23].

More precisely, for $0 < \epsilon < 1$ and n large enough, we can assure that

$$\big|\lambda(C_2 S)\big| = \big|1 - \lambda(C_1 T)\big| \leqslant \epsilon < 1$$

for all eigenvalues $\lambda(C_2 S)$ except for some outliers. □

Remark. We have proven a general result: *most* of the eigenvalues of $C_2 S$ have small enough moduli. This means that our algorithm in many cases can get to the solution, even for ill-conditioned large matrices T, whenever the initial guess vector turns out to be properly chosen.

The choice of the initial guess vector x_0 is a difficult theoretical problem. To ensure convergence, it suffices that the initial error vector $x_0 - x$ belongs to the span of the eigenspaces associated to eigenvalues $|\lambda(C_2 S)| < 1$. If the maximum of these moduli is $|\lambda_0| \leqslant \epsilon < 1$ then, after k iterations, the error is

$$(x_k - x) = (C_2 S)^k (x_0 - x),$$
$$\|x_k - x\| \leqslant |\lambda_0|^k \|x_0 - x\| \leqslant \epsilon^k \|x_0 - x\|$$

which gives an idea of the rate of convergence of the algorithm, even in the ill-conditioned case.

5.2. Numerical results and discussion

We have considered a great variety of Toeplitz systems and observed the performance of the new algorithm, in comparison to other techniques: Levinson's method (implemented in a form that requires $2n^2$ flops [4]), the PCG method with the circulant preconditioner given in [23], and the PCG method with Chan's circulant preconditioner [22] (here renamed PCG2). For each problem, the starting vector was the zero vector, y is a random vector, and the iterations were terminated when the norm of the residual $Tx - y$ fell below a fixed threshold. All the necessary FFTs were computed using the FFTW package [31,32], taking advantage of the real character of the data. The CPU time was measured by averaging over a number of runs.

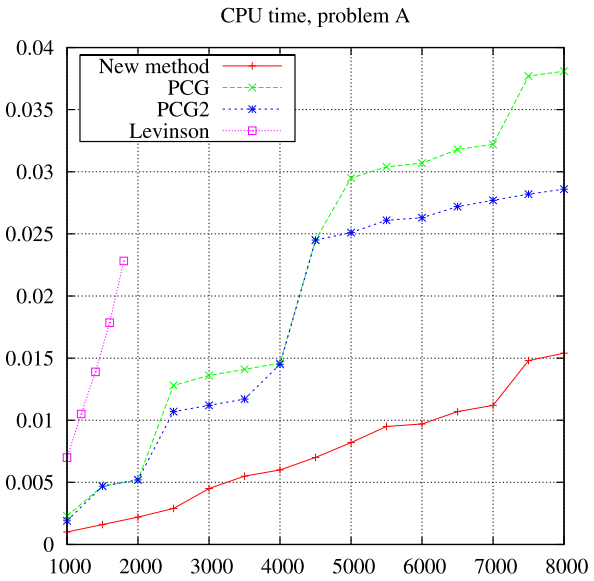

Fig. 2. Problem A: CPU time as a function of the Toeplitz matrix size, for the new algorithm, PCG, PCG2 and Levinson's method.

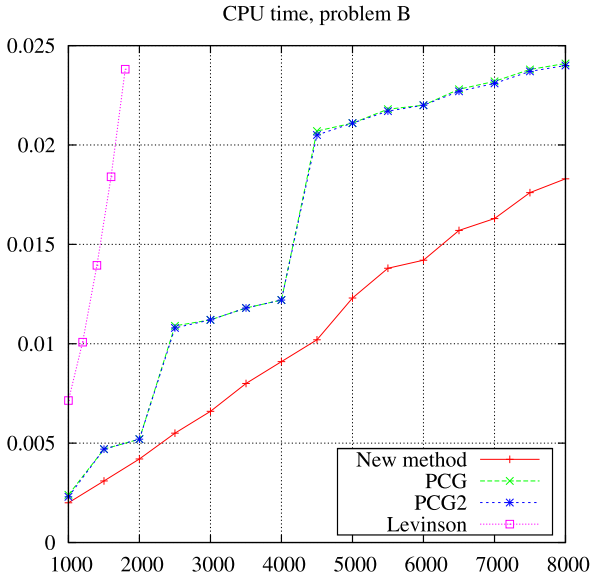

Fig. 3. Problem B: CPU time as a function of the Toeplitz matrix size, for the new algorithm, PCG, PCG2 and Levinson's method.

Problem A. The matrices were randomly generated, real, symmetric, and positive definite. The results are shown in Fig. 2. Clearly the new method outperforms the existing ones, for all matrix sizes.

Problem B. Fig. 3 shows the results of the second experiment, for the matrix elements $t_i = (1+|i|)^{-2}$. This problem appeared in [23] as Problem 6. Once again, the new algorithm outperforms the others, regardless the matrix size.

Problem C. This problem corresponds to Problem 1 in [23], defined through a parameter a as $t_i = a^i$ for $i = 0, 1, 2, 3$, $t_i = 0$ otherwise. Fig. 4 shows the optimum behavior of our algorithm, as in the preceding problems.

Problem D. Corresponds to Problem 2 in [23], defined through a parameter a as $t_i = a^i$ for all i. In this case the results given in Fig. 5 show that the best performance is achieved either by PCG or the new method, depending on the matrix size.

Problem E. The last problem is Problem 4 of [23], defined through two parameters a, b as $t_i = (i+1)a^i + b^i$. Fig. 6 illustrates that PCG, PCG2 and the new method compete for the best performance, but there is not a clear favorite for all matrix sizes.

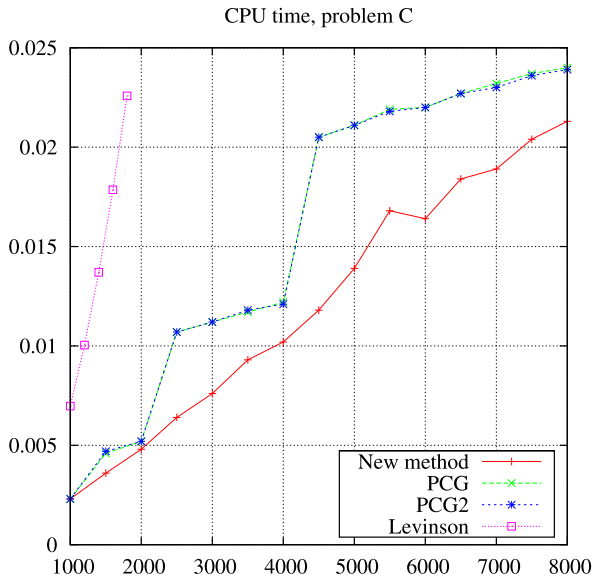

Fig. 4. Problem C: CPU time as a function of the Toeplitz matrix size, for the new algorithm, PCG, PCG2 and Levinson's method.

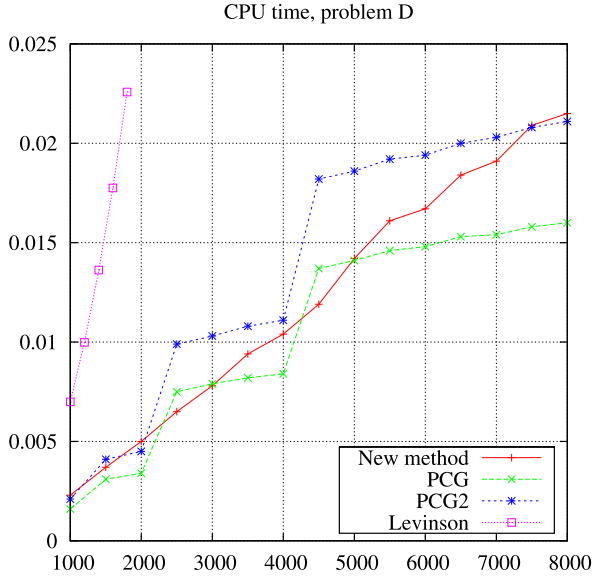

Fig. 5. Problem D: CPU time as a function of the Toeplitz matrix size, for the new algorithm, PCG, PCG2 and Levinson's method.

In summary, we deduce that, for a given matrix size, one can find examples where a particular method (PCG, PCG2 or the proposed one) outperforms the others. But there is not a unique method with optimal behavior independently of the matrix size. Nevertheless, the algorithm here proposed outperforms the other techniques for a great variety of problems and for a wide range of matrix sizes.

When comparing the results, one should keep in mind that the proposed technique does not converge for all symmetric, positive definite Toeplitz matrices. In particular, the usefulness of the new method is limited when the matrix is ill-conditioned. However, the theoretical results given in this paper help in identifying the class of matrices for which the method is useful. Note that the method can be applied in the nonsymmetric case, though.

No attempt was made to improve the convergence rate of the method (say, using α_{best}) or to tune the free parameters of the circulant in any other way. Nevertheless, for circulants of size $m = 2n$, using α_{best} would lead to better performance.

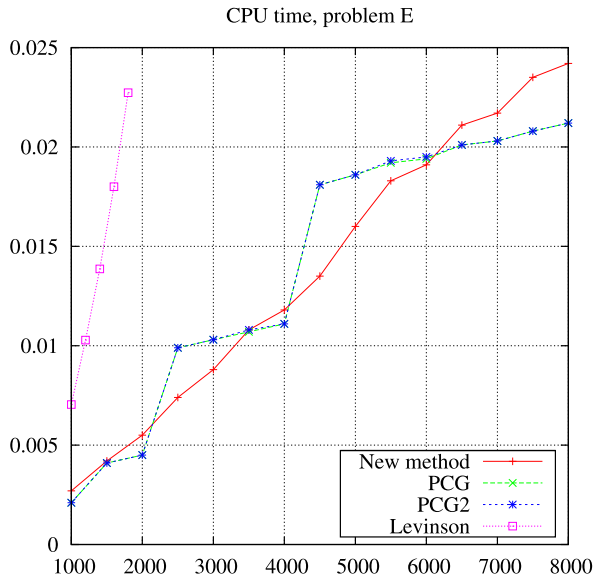

Fig. 6. Problem E: CPU time as a function of the Toeplitz matrix size, for the new algorithm, PCG, PCG2 and Levinson's method.

6. Conclusions

It is possible to replace a set of linear equations with a larger but more structured set, the solution of which leads to the solution of the original problem. The question is whether there are computational benefits in doing so. We have tested this idea for Toeplitz equations, since the Toeplitz structure immediately suggests trading-off size by circulant structure. The idea of circulant embeddings, in one way or another, is quite well known. In signal processing, the computation of linear convolutions is routinely accomplished using zero padding and the FFT; see also [19]. However, the idea explored in the present paper uses circulant embeddings in a quite different direction: the Toeplitz set of equations $Tx = y$ (T nonsingular, y known) is mapped into an equivalent, larger set $Ca = b$ with circulant structure, so that both a and b contain unknown quantities. The unknown quantities are determined by a simple iteration (multiplication by C and C^{-1}), which constraints the vectors a and b to certain known subspaces.

This idea leads to a very simple yet efficient algorithm. Since the method does not converge for all nonsingular T, or even all positive definite T, we gave convergence conditions. We have also presented the results of numerical experiments.

The same basic idea might also be tried in the case of submatrices of circulants that are not necessarily Toeplitz. In fact, the idea could in principle be explored for other matrix structures which are "close" to an even more structured class (Hankel matrices and submatrices of retrocirculants, for example). In these cases, however, the potential of the method, at least in its basic form, remains to be seen.

Acknowledgments

This work has been partially supported by the Fundação para a Ciência e Tecnologia (FCT), Portugal, and by Universidad Politécnica de Madrid through the UPM TACA research group, and the Ministerio de Ciencia e Innovación (Spain) under the research project TEC2009-08133.

References

[1] G.H. Golub, C.F. Van Loan, Matrix Computations, second edition, The Johns Hopkins University Press, Baltimore, 1989.
[2] C. Papaodysseus, E. Koukoutsis, G. Carayannis, Numerical behaviour of Toeplitz solutions, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 89, vol. 4, Glasgow, UK, May 1989, pp. 2097–2100.
[3] M.K. Ciliz, H. Krishna, Split Levinson algorithm for Toeplitz matrices with singular sub-matrices, IEEE Trans. Circuits Syst. 36 (6) (June 1989) 922–924.
[4] T.F. Chan, P.C. Hansen, A look-ahead Levinson algorithm for general Toeplitz systems, IEEE Trans. Signal Process. 40 (5) (May 1992) 1079–1090.
[5] W.E. Trench, Solution of systems with Toeplitz matrices generated by rational functions, Linear Algebra Appl. 74 (1986) 191–211.
[6] W.E. Trench, Toeplitz systems associated with the product of a formal Laurent series and a Laurent polynomial, SIAM J. Matrix Anal. Appl. 9 (2) (April 1988) 181–193.
[7] B. Friedlander, M. Morf, T. Kailath, L. Ljung, New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices, Linear Algebra Appl. 27 (1979) 31–60.
[8] N. Kalouptsidis, D. Manolakis, G. Carayannis, Efficient recursive triangularization, inversion and system solution of near-to-Toeplitz matrices and applications in signal processing, Signal Process. 6 (3) (June 1984) 235–259.
[9] I. Gohberg, T. Kailath, I. Koltracht, Efficient solutions of linear systems of equations with recursive structure, Linear Algebra Appl. 80 (1986) 81–113.
[10] N. Kalouptsidis, G. Carayannis, D. Manolakis, Fast algorithms for block Toeplitz matrices with Toeplitz entries, Signal Process. 6 (1) (January 1984) 77–81.

[11] 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 (1) (January 1998) 107–139.
[12] T. Kailath, A.H. Sayed, Displacement structure: Theory and applications, SIAM Rev. 37 (3) (1995) 297–386.
[13] 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.
[14] R.P. Brent, F.G. Gustavson, D.Y.Y. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximants, J. Algorithms 1 (1980) 259–295.
[15] G.S. Ammar, W.B. Gragg, Superfast solution of real positive definite Toeplitz systems, SIAM J. Matrix Anal. Appl. 9 (1) (January 1988) 61–76.
[16] G. Codevico, G. Heinig, M. Van Barel, A superfast solver for real symmetric Toeplitz systems using real trigonometric transformations, Numer. Linear Algebra Appl. 12 (8) (2005) 699–713.
[17] D.A. Bini, G. Codevico, M. Van Barel, Solving Toeplitz least squares problems by means of Newton's iteration, Numer. Algorithms 33 (2003) 93–103.
[18] G. Codevico, V.Y. Pan, Marc Van Barel, Newton-like iteration based on a cubic polynomial for structured matrices, Numer. Algorithms 36 (2004) 365–380.
[19] G. Strang, A proposal for Toeplitz matrix calculations, Stud. Appl. Math. 74 (1986) 171–176.
[20] R.H. Chan, G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, SIAM J. Sci. Stat. Comput. 10 (1) (January 1989) 104–119.
[21] R.H. Chan, The spectrum of a family of circulant preconditioned Toeplitz systems, SIAM J. Numer. Anal. 26 (2) (April 1989) 503–506.
[22] R.H. Chan, Circulant preconditioners for Hermitian Toeplitz systems, SIAM J. Matrix Anal. Appl. 10 (4) (October 1989) 542–550.
[23] T.-K. Ku, C.-C. Jay Kuo, Design and analysis of Toeplitz preconditioners, IEEE Trans. Signal Process. 40 (1) (January 1992) 129–141.
[24] R.H. Chan, Michael K. Ng, Conjugate gradient methods for Toeplitz systems, SIAM Rev. 38 (3) (September 1996) 427–482.
[25] M.K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, 2004.
[26] M.E. Dominguez-Jimenez, P.J.S.G. Ferreira, A new preconditioner for Toeplitz matrices, IEEE Signal Process. Lett. 16 (9) (September 2009) 758–761.
[27] P.J.S.G. Ferreira, Localization of the eigenvalues of Toeplitz matrices using additive decomposition, embedding in circulants, and the Fourier transform, in: M. Blanke, T. Söderström (Eds.), Proceedings of SysID'94, 10th IFAC Symposium on System Identification, vol. III, Copenhagen, Denmark, July 1994, pp. 271–276.
[28] P.J.S.G. Ferreira, Noniterative and faster iterative methods for interpolation and extrapolation, IEEE Trans. Signal Process. 42 (11) (November 1994) 3278–3282.
[29] P.J.S.G. Ferreira, Interpolation in the time and frequency domains, IEEE Signal Process. Lett. 3 (6) (June 1996) 176–178.
[30] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1990.
[31] M. Frigo, S.G. Johnson, FFTW: An adaptive software architecture for the FFT, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 98, vol. 3, Seattle, USA, May 1998, pp. 1381–1384.
[32] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2) (February 2005) 216–231.

Paulo J.S.G. Ferreira was born in Torres Novas, Portugal, and is a Professor at the Departamento de Electrónica, Telecomunicações e Informática/IEETA, Universidade de Aveiro, Portugal. He is a member of the editorial board of the *Journal of Applied Functional Analysis* and an Editor-in-Chief of *Sampling Theory in Signal and Image Processing*. He was an Associate Editor of the *IEEE Transactions on Signal Processing*, and co-edited with John J. Benedetto the book *Modern Sampling Theory: Mathematics and Applications*. His research interests are in the area of signal processing and include coding, sampling and signal reconstruction.

María Elena Domínguez received her Bachelor degree in Mathematical Sciences from the Universidad Complutense de Madrid in 1992, and the Ph.D. degree on Electrical Engineering from the Universidad Politécnica de Madrid in 2001. From 1992 she is with the Departamento de Matemática Aplicada, E.T.S.I. Industriales, Universidad Politécnica de Madrid. Her research interests include audio compression, multiresolution signal processing, wavelets and filter bank theory. In 2002 Dr. Domínguez received one of the Extraordinary Awards from the Universidad Politécnica de Madrid for the best doctoral dissertation.