

Solving Toeplitz least squares problems by means of Newton's iteration

D.A. Bini
G. Codevico
M. Van Barel

Report TW 333, December 2001



Katholieke Universiteit Leuven
Department of Computer Science

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Solving Toeplitz least squares problems by means of Newton's iteration

D.A. Bini
G. Codevico
*M. Van Barel**

Report TW 333, December 2001

Department of Computer Science, K.U.Leuven

Abstract

We extend the algorithm of [D. Bini and B. Meini, *Approximate displacement rank and applications*, in Structured Matrices in Mathematics, Computer Science, and Engineering II,, V. Olshevsky Editor, Contemporary Mathematics 281, pp.215-232, American Mathematical Society, Rhode Island, 2001 , AMS, 1999], based on Newton's iteration and on the concept of ϵ -displacement rank, to the computation of the generalized inverse A^+ of an $m \times n$ Toeplitz matrix A . We introduce new strategies for the dynamical control of the truncation level ϵ at each step of the iteration. Numerical experiments and an application to a problem of image restoration are shown. An object-oriented implementation in C++ is described.

Submitted for the Proceedings of the International Conference on Numerical Algorithms, October 1-5, 2001, Marrakesh, Morocco (Special volumes of Numerical Algorithms and Advances in Computational Mathematics).

Keywords : Toeplitz matrices, least squares, approximate displacement rank, Newton's iteration, image restoration.

AMS(MOS) Classification : Primary : 65F10, Secondary : 65F20.

*Corresponding author

Solving Toeplitz least squares problems by means of Newton's iteration

D.A. Bini*, G. Codevico† and M. Van Barel†

10th December 2001

Abstract

We extend the algorithm of [4], based on Newton's iteration and on the concept of ϵ -displacement rank, to the computation of the generalized inverse A^+ of an $m \times n$ Toeplitz matrix A . We introduce new strategies for the dynamical control of the truncation level ϵ at each step of the iteration. Numerical experiments and an application to a problem of image restoration are shown. An object-oriented implementation in C++ is described.

Keywords: Toeplitz matrices, least squares, approximate displacement rank, Newton's iteration, image restoration.

AMS Subject Classification: 65F10, 65F20,

1 Introduction

Let $A = (a_{i,j})$ be an $m \times n$ Toeplitz matrix, i.e., such that $a_{i,j} = t_{i-j}$ for $i = 1, \dots, m$, $j = 1, \dots, n$, where t_k , $k = -n + 1, \dots, m - 1$ are real numbers. The problem of solving Toeplitz systems $A\mathbf{x} = \mathbf{b}$ for $m = n$ or Toeplitz least squares problems if $n \neq m$, is encountered in many applications, e.g., in image restoration problems [2].

In this paper, by following [5, 4], we consider Newton's iteration for approximating the generalized inverse A^+ of A . Firstly, we recall the nice convergence and regularizing properties of Newton's iteration, then we extend the concept of displacement rank and ϵ -displacement rank to nonsquare matrices and analyze its properties. This concept is then applied for modifying Newton's iteration in order to maintain its convergence properties and to reduce its computational cost per step. Finally we introduce some heuristic strategies which can reduce the computational cost and present an object oriented implementation of the algorithm in C++. Numerical experiments and an application to a problem of image restoration are shown.

*Dipartimento di Matematica, Università di Pisa

†Department of Computer Science, Catholic University of Leuven

2 Newton's iteration

We recall the main properties of the sequence of $n \times m$ matrices $\{X_i\}_i$ generated by the Newton iteration

$$X_{i+1} = 2X_i - X_iAX_i, \quad i = 0, 1, \dots, \quad (1)$$

starting from an initial $n \times m$ matrix X_0 . The iteration (1) is obtained by formally applying the customary Newton iteration to the scalar equation $a - x^{-1} = 0$ and, under suitable conditions on X_0 it holds $\lim X_i = A^{-1}$ for $m = n$ and $\det A \neq 0$. We refer the reader to [1, 6, 14, 3] for issues related with Newton's iteration.

2.1 Convergence

Let us first consider for simplicity the case where A is symmetric and positive definite with eigenvalues $0 < \lambda_1 \leq \dots \leq \lambda_n$. From (1) we deduce that for the residual $R_i = I - AX_i$ it holds $R_{i+1} = R_i^2$ and $X_i = A^{-1}(I - R_i)$. In this way, if $\rho(R_0) < 1$ then the sequence X_i converges double exponentially to A^{-1} , since $R_i = R_0^{2^i}$ and $\rho(R_i) = \rho(R_0)^{2^i}$.

For $X_0 = \theta I$, the eigenvalues of R_0 are $\mu_j = 1 - \theta\lambda_j$, $j = 1, \dots, n$ so that if $\theta \leq 1/\lambda_n$ then $\rho(R_0) = 1 - \lambda_1\theta < 1$ and convergence occurs. Since for any operator norm $\|\cdot\|$ it holds $\rho(A) \leq \|A\|$ we find that choosing $\theta = 1/\|A\|$ guarantees convergence.

If $m \neq n$ and A has full rank, we can provide an explicit representation of the generalized inverse A^+ of A as [9]

$$\begin{aligned} A^+ &= (A^T A)^{-1} A^T \quad \text{for } m > n \\ A^+ &= A^T (A A^T)^{-1} \quad \text{for } m < n. \end{aligned} \quad (2)$$

Consider the case $m > n$, denote $S = A^T A$, apply Newton's iteration to S and obtain the sequence

$$\begin{aligned} W_{i+1} &= 2W_i - W_i S W_i, \\ W_0 &= \theta I \end{aligned}$$

which converges to S^{-1} if $\theta \leq 1/\|A^T A\|$. Multiplying the latter equation to the right by A^T and denoting $X_i = W_i A^T$ we obtain that (1) holds where $X_0 = \theta A^T$. Since for $\theta \leq 1/\|A^T A\|$ the sequence W_i converges to S^{-1} , from (2) we deduce that X_i converges to A^+ . Setting $R_i = I - AX_i$, we have $R_i = R_{i-1}^2$, $X_i = A^+(I - R_i)$, and $\rho(R_i) = \rho(R_0)^{2^i} = (1 - \theta\lambda_1)^{2^i}$, where λ_1 is the smallest eigenvalue of $A^T A$.

Similarly we may proceed in the case $m < n$ by multiplying (1) on the left by A^T and setting $X_i = A^T W_i$. In this way for the residual $R_i = I - X_i A$, we have $R_i = R_{i-1}^2$, $X_i = (I - R_i)A^+$, $\rho(R_i) = \rho(R_0)^{2^i} = (1 - \theta\lambda_1)^{2^i}$, where λ_1 is the smallest eigenvalue of $A A^T$.

Since $\mathbf{x}^* = A^+ \mathbf{b}$ provides the least squares solution of the system $A\mathbf{x} = \mathbf{b}$ (of minimal Euclidean norm), then $X_i \mathbf{b}$ provides a sequence of vectors converging quadratically to this solution \mathbf{x}^* .

Remark 1. *Newton's versus Landweber's.* Newton's iteration provides an approximation of the least squares solution $\mathbf{x}^* = A^+ \mathbf{b}$ by means of the polynomial

representation $\mathbf{x}_k = X_k \mathbf{b} = A^T \theta \sum_{i=0}^{2^k-1} (I - \theta A A^T)^i \mathbf{b}$ while the Landweber iteration $\mathbf{y}_{i+1} = \mathbf{y}_i + \theta(\mathbf{b} - A \mathbf{y}_i)$ [2] provides the similar approximation $\mathbf{y}_k = A^T \theta \sum_{i=0}^{k-1} (I - \theta A A^T)^i \mathbf{b}$. Observe that since $\mathbf{x}_k = \mathbf{y}_{2^k}$, Newton's iteration provides a quadratic acceleration of Landweber's method.

2.2 Regularization

Newton's iteration, as well as Landweber's, has nice regularization features. Assume that the vector \mathbf{b} is affected by an unknown noise $\boldsymbol{\eta}$ so that we would like to recover the solution \mathbf{x} such that $A \mathbf{x} = \mathbf{b} - \boldsymbol{\eta}$ but we are compelled to solve the system $A \mathbf{w} = \mathbf{b}$. Let $A = U \Sigma V^T = \sum \mathbf{u}_i \sigma_i \mathbf{v}_i^T$ be the SVD of A where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$, $p = \min(m, n)$, are the singular values of A and \mathbf{u}_i , \mathbf{v}_i the left and right singular vectors of A , respectively. In this way $\mathbf{w} - \mathbf{x} = A^+ \boldsymbol{\eta} = \sum_i \mathbf{v}_i \frac{\mathbf{u}_i^T \boldsymbol{\eta}}{\sigma_i}$ provides the expression of the noise component in the least squares solution \mathbf{w} .

For the approximation $\mathbf{x}_k = X_k \mathbf{b}$ of the least squares solution of the system $A \mathbf{w} = \mathbf{b}$ obtained with Newton's iteration we have $\mathbf{x}_k = \mathbf{x} + \boldsymbol{\epsilon}_k + \boldsymbol{\delta}_k$ where $\boldsymbol{\epsilon}_k = \sum_j \mathbf{v}_j \frac{(1 - \theta \sigma_j^2)^{2^k}}{\sigma_j} (\mathbf{u}_j^T \mathbf{f})$ and $\boldsymbol{\delta}_k = \sum_j \mathbf{v}_j \frac{1}{\sigma_j} (\mathbf{u}_j^T \boldsymbol{\eta}) \phi_{j,k}$ are the approximation error and the noise component, respectively, where $\phi_{j,k} = 1 - (1 - \theta \sigma_j^2)^{2^k}$.

Observe that as k tends to infinity $\boldsymbol{\epsilon}_k$ tends to zero while $\boldsymbol{\delta}_k$ tends to the noise part $A^+ \boldsymbol{\eta} = \sum_j \mathbf{v}_j \frac{1}{\sigma_j} (\mathbf{u}_j^T \boldsymbol{\eta})$. Moreover, the j -th noise component $\mathbf{v}_j (\mathbf{u}_j^T \boldsymbol{\eta})$ in the right hand side of the system is amplified in the solution by the factor $\frac{1}{\sigma_j}$ which is large for smaller σ_j . However, in the expression of $\boldsymbol{\delta}_k$ the components which are more amplified by the lowest singular values, are dumped by the "filters" $\phi_{j,k}$ which are close to zero in correspondence with the smallest singular values σ_j .

3 Displacement rank

In this section we generalize the concept of ϵ -displacement rank of [4] to non-square matrices. For this purpose define $Z_n = (z_{i,j})$ the $n \times n$ lower shift matrix such that $z_{i,j} = 1$ for $i = j + 1$ and $z_{i,j} = 0$ elsewhere, and introduce the displacement operator $\Delta_{n,m}(A) = Z_m A - A Z_n$ for an $m \times n$ matrix A . For the sake of notational simplicity we write $\Delta(A)$ in place of $\Delta_{n,m}(A)$ since the sizes of the matrix A are clear from the context. It is easy to verify that if A is Toeplitz then $\Delta(A)$ has rank at most 2. The following result generalizes Theorem 11.3 of [6] to the case $m \neq n$ and can be easily proven by using the same argument as in [6].

Proposition 1. *Let $A \in \mathbf{R}^{m \times n}$ and $\mathbf{u}_i \in \mathbf{R}^m$, $\mathbf{v}_i \in \mathbf{R}^n$ for $i = 1, \dots, k$ be such that $\Delta(A) = \sum_{i=1}^k \mathbf{u}_i \mathbf{v}_i^T$. Then we have $A = L(\mathbf{a}) + \sum_{i=1}^k L(\mathbf{u}_i) U(Z_n \mathbf{v}_i)$ where \mathbf{a} denotes the first column of A and where $L(\mathbf{x})$, $U(\mathbf{y})$ denote lower triangular and upper triangular Toeplitz matrices, defined by their first column \mathbf{x} and the first row \mathbf{y}^T respectively. The matrices $L(\mathbf{x})$, $U(\mathbf{y})$ are respectively, of dimension $m \times n$ and $n \times n$ if $m \geq n$ or of dimension $m \times m$ and $m \times n$ if $m < n$.*

We call $\text{rank}(\Delta(A))$ the *displacement rank* of A and we denote it with $\text{drk}(A)$; we call *displacement representation* of A the expression $A = L(\mathbf{a}) +$

$\sum_{i=1}^k L(\mathbf{u}_i)U(Z_n \mathbf{v}_i)$. The following properties are immediate formal consequences of the definition of $\Delta(A)$.

$$\begin{aligned} \Delta(AB) &= \Delta(A)B + A\Delta(B) & \Delta(A^{-1}) &= -A^{-1}\Delta(A)A^{-1}, \quad m = n, \\ \Delta(A^T) &= -\Delta(A)^T & A\Delta(A^+) + \Delta(A)A^+ &= 0, \quad m \leq n, \\ & & A^+\Delta(A) + \Delta(A^+)A &= 0, \quad m \geq n, \end{aligned}$$

Moreover, from the above expressions and (2) we deduce that

$$\begin{aligned} \Delta(A^+) &= ((I - A^+A)\Delta(A^T) - A^+\Delta(A)A^T)(AA^T)^{-1} \text{ for } m \leq n \\ \Delta(A^+) &= (A^T A)^{-1}(\Delta(A^T)(I - AA^+) - A^T \Delta(A)A^+) \text{ for } n \leq m \end{aligned} \quad (3)$$

Whence we get the following bounds on the displacement rank

$$\begin{aligned} \text{drk}(A) &= \text{drk}(A^T) & \text{drk}(A^{-1}) &= \text{drk}(A), \quad m = n, \\ \text{drk}(AB) &\leq \text{drk}(A) + \text{drk}(B) & \text{drk}(A^+) &\leq \text{drk}(A) + \min\{\text{drk}(A), |m - n|\}. \end{aligned}$$

In particular, for a Toeplitz matrix A the displacement rank of A^+ is at most 4. Moreover, for the matrices X_i generated by Newton's iteration we have

$$\Delta(X_{i+1}) = \Delta(X_i)(2I - AX_i) - X_i\Delta(A)X_i - X_iA\Delta(X_i), \quad (4)$$

whence $\text{drk}(X_i) \leq 2\text{drk}(X_i) + \text{drk}(A)$. That is the displacement rank of X_i may grow exponentially up to saturation. A nice observation is that for a Toeplitz matrix A , $\lim X_i = A^+$ has displacement rank at most 4. Numerically speaking, this means that the matrices X_i , for i large enough, are well approximated by matrices with displacement rank at most 4. This property can be formalized by the concept of ϵ -displacement rank.

Definition 1. For any $m \times n$ matrix A and for any $\epsilon > 0$ we define the ϵ -displacement rank $\text{drk}_\epsilon(A) = \min_E \text{rank}(\Delta(A) + E)$, where the minimum is taken over all the matrices E such that $\|E\|_2 < \epsilon\|A\|_2$, for the Euclidean norm $\|\cdot\|_2$.

Let $\Delta(A) = U\Sigma V^T = \sum_{i=1}^k \sigma_i u_i v_i^T$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$ be the singular value decomposition (SVD) of $\Delta(A)$. We have the following characterization of the ϵ -displacement rank [4].

Proposition 2. Let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$ be the singular values of $\Delta(A)$. Then $\text{drk}_\epsilon(A) = r$ if and only if $\sigma_r > \epsilon\|A\| \geq \sigma_{r+1}$.

The quadruple (Ae_1, U, σ, V) , for $\sigma = (\sigma_1, \dots, \sigma_k)$, is called *orthogonal displacement generator* (ODG) of A while $(Ae_1, \tilde{U}, \tilde{\sigma}, \tilde{V})$, for $\tilde{\sigma} = (\sigma_1, \dots, \sigma_r)$, and \tilde{U}, \tilde{V} denoting the submatrices of U and V made up by the first r columns, respectively, is called *orthogonal ϵ -displacement generator*. In the light of Proposition 1 they provide the orthogonal displacement representation and the following orthogonal ϵ -displacement representation of A : $A \approx A_\epsilon = L(Ae_1) + \sum_{i=1}^r \sigma_i L(u_i) L^T(Zv_i)$.

So it is straightforward that multiplying an $m \times n$ matrix, represented in terms of its orthogonal displacement representation of length k and a vector can be performed by means of $2k + 2$ FFTs and $2k + 1$ convolutions of dimension $2m$ and $2k + 1$ FFTs and k convolutions of dimension $2n$.

The following properties generalize the analogous properties of [4] which hold in the case of square matrices

Proposition 3. *Let $r = \text{drk}_\epsilon(A) \leq \text{drk}(A) = k$, let A_ϵ be an ϵ -displacement representation of A , and let $\sigma_1, \sigma_2, \dots, \sigma_k$ be the singular values of $\Delta(A)$. Then it holds that*

$$\frac{\|A_\epsilon - A\|}{\|A\|} \leq \sqrt{mn} \sum_{i=r+1}^k \sigma_i \leq \sqrt{mn}(k-r)\epsilon. \quad (5)$$

Conversely, we have If B is an approximation of A such that $\|A - B\| \leq \epsilon\|A\|$ then $\text{drk}_{2\epsilon}(B) \leq k$

The following inequalities are an immediate consequence of the definition: $\text{drk}_\delta(A+B) \leq \text{drk}_\epsilon(A) + \text{drk}_\epsilon(B)$, $\delta = \epsilon \frac{\|A\| + \|B\|}{\|A+B\|}$; $\text{drk}_\delta(AB) \leq \text{drk}_\epsilon(A) + \text{drk}_\epsilon(B)$, $\delta = \epsilon \frac{\|A\| \cdot \|B\|}{\|AB\|}$; $\text{drk}_\delta(A^{-1}) \leq \text{drk}_\epsilon(A)$, $\delta = \epsilon\mu(A)$, for $m = n$ where $\mu(A) = \|A\| \cdot \|A^+\|$ is the spectral condition number of $\|A\|$. A similar expression can be found for $\text{drk}_\epsilon(A^+)$ by means of (3).

Now we are ready to introduce the operator $\text{trunc}_\epsilon(\cdot)$ defined on the sets of orthogonal displacement generators in the following way:

$$\begin{aligned} \text{trunc}_\epsilon((\mathbf{a}, U, \sigma, V)) &= (\mathbf{a}, \widehat{U}, \widehat{\sigma}, \widehat{V}), \\ \widehat{\sigma} &= (\sigma_1, \dots, \sigma_{\widehat{k}}), \widehat{U} = [u_1, \dots, u_{\widehat{k}}], \widehat{V} = [v_1, \dots, v_{\widehat{k}}] \in \mathbb{R}^{n \times \widehat{k}}, \\ \text{and } \widehat{k} \text{ is such that } &\sigma_{\widehat{k}} > \sigma_1 \epsilon \geq \sigma_{\widehat{k}+1}, \end{aligned} \quad (6)$$

and extend $\text{trunc}_\epsilon(\cdot)$ to the set of matrices $\widetilde{A}_\epsilon = \text{trunc}_\epsilon(A)$ where $\widetilde{A}_\epsilon = L(A\mathbf{e}_1) + \sum_{i=1}^{\widehat{k}} L(u_i)L^T(Zv_i)$. Clearly from proposition 3 it holds $\|A - \widetilde{A}_\epsilon\| \leq \sqrt{mn}(k - \widehat{k})\sigma_1\epsilon$.

The equation (6) provides a means of computing an ϵ -displacement representation of A whenever an ODG is available. The concept of ϵ -displacement generator has been implicitly used in [14] for implementing the Newton iteration for inverting Toeplitz matrices. We discuss this topic in the next section.

4 Approximated Newton's Iteration.

With the tools introduced in the previous sections, we are able to define our first version of the Newton iteration method adjusted to cope with the case of nonsquare Toeplitz-like matrices, similar as in [5, 4]. Local convergence analysis of a similar modification of Newton's iteration method has been performed in [14] for square matrices.

Let us introduce the following approximated Newton iteration:

$$\begin{aligned} Y_i &= 2X_i - X_iAX_i \\ X_{i+1} &= \text{trunc}_{\epsilon_i}(Y_i) \end{aligned} \quad (7)$$

where $\text{trunc}_\epsilon(\cdot)$ is defined in (6) and the value ϵ_i is adaptively chosen at each step according to suitable strategies which we will discuss later on. These strategies are designed in order to keep $\text{drk}(X_i)$ as small as possible by still guaranteeing convergence of the iteration. All the matrices generated by (7) are represented in terms of their ODGs. We recall the following convergence result of [5, 4] which we extend to the case of arbitrary m and n :

Proposition 4. Let $X_0 \in \mathbb{R}^{m \times n}$ and β such that $\|X_0\| \leq 1 - \beta$, and let $\epsilon_i = \beta / (4\sqrt{mn}(2h_i + k)\sigma_1^{(i)}\|A\|)$, $k = \text{drk}(A)$, $h_i = \text{drk}(X_i)$, $\sigma_1^{(i)} = \rho(\Delta(X_i))$. Then for the residual $R_i = I - AX_i$ of the sequence $(X_i)_i$ generated from the approximate Newton iteration we have $\|R_i\| \leq (1 - \beta/2)^{2^i}$.

A different convergence result which has more adaptive features is

Proposition 5. With the notation of (7) let $R_i = I - AX_i$. If ϵ_i is chosen so that $\|A(X_{i+1} - Y_i)\| \leq \|R_i\|^2(1 - \|R_i\|^2)/2$ then the sequence X_i quadratically converges.

Proof. Let $E = Y_i - X_{i+1}$, $R_{i+1} = I - AX_{i+1}$. Then from $R_{i+1} = I - A(Y_i - E) = R_i^2 + AE$ we obtain $\|R_{i+1}\| \leq \|R_i\|^2 + \|AE\| \leq \phi(\|R_i\|)$, where $\phi(x) = x^2(3 - x^2)/2$. The proposition holds since 0 is an attractive fixed point of $\phi(x)$ and $\phi'(0) = 0$.

If the approximate Newton iteration is applied in order to exploit its regularizing properties derived by the exact iteration, then we do not need to perform many steps of it and asymptotic issues like convergence are not relevant anymore.

In order to implement the iteration in terms of ODGs we have to determine the formulae which relate $\text{ODG}(X_i)$ to $\text{ODG}(X_{i+1})$. Observe that from (4) we get $\Delta_{i+1} = 2\Delta_i - \Delta_i AX_i - X_i \Delta(A) X_i - X_i A \Delta_i$ with $\Delta_i = \Delta(X_i)$ which provides a representation of Δ_{i+1} of rank at most $2\text{rank}\Delta_i + \text{rank}\Delta(A)$. Computing the SVD of Δ_{i+1} allows us to truncate X_{i+1} to the desired value ϵ_{i+1} . We leave to the reader the details of this computation. We just observe that the computational cost of this part is $O(r_i^2 n \log n)$ where $r_i = \text{rank}\Delta_i$.

Remark 2. A rather difficult computational issue is the computation of the norm of matrices represented in terms of their ODG which is needed to implement Proposition 5 and to keep under control the convergence. If $(\mathbf{x}, U, \boldsymbol{\sigma}, V)$ is an ODG of a matrix X then the expression $X = L(\mathbf{x}) + \sum_{i=1}^k \sigma_i L(\mathbf{u}_i) U(Z_n \mathbf{v}_i)$ provides the useful upper bound to $\|X\|$ $\|X\| \leq \|\mathbf{x}\| + \sqrt{mn} \sum \sigma_i$.

5 Adaptive Strategies

Performing numerical experiments with the approximate Newton's iteration shows that in most cases the ϵ -displacement rank of X_i takes values larger than needed for convergence. In fact, for many square Toeplitz matrices, approximating Y_i with X_i such that $\text{drk}(X_i) = k = 2$, i.e, with *cutting level 2*, is enough to guarantee convergence. In this section we discuss some adaptive strategies which generally produce convergent sequences $\{X_i\}$, keeping $\text{drk}(X_i)$ as small as possible. We will use pseudo-code to explain the strategies analyzed. These strategies have been implemented in an *Object-Oriented* way in *C++* which we describe next.

5.1 Cutting-level strategies and optimizations

The first computational tool we need is an efficient test for checking the condition $\rho(R) < 1$ given $\text{ODG}(R)$. Remark 2 provides in part this tool since we can compute at a low cost an upper bound to $\rho(R)$. This is not enough since we

need a lower bound too. We know that for any $\mathbf{x} \in \mathbf{R}^n$ such that $\|\mathbf{x}\| = 1$ the quantity $\|R\mathbf{x}\|$ is a lower bound of the spectral radius of R . Given an ODG of R , the computation of $R\mathbf{x}$ can be performed with few FFT's. This bound can be turned into an accurate estimate by means of $\|R\mathbf{x}^*\|$ where $\mathbf{x}^* = \mathbf{x}_k/\|\mathbf{x}_k\|$ is the normalized vector obtained by applying say, k steps of the power method [9] $\mathbf{x}_{i+1} = R\mathbf{x}_i$, with $\mathbf{x}_0 = \mathbf{x}$ and k large enough. In fact, this method generates a sequence of vectors which converges to the dominant eigenvector of R . Let us denote by $\rho^*(R)$ the value obtained in this way.

A first strategy for cutting the displacement rank of X_i to the value k , which depends on how accurate is the estimate of $\rho(R_i)$ provided by $\rho^*(R_i)$, relies in increasing by a given amount h the value of the cutting level k in order to capture more information of Y_{i+1} if $\rho^*(R_{i+1}) > \rho^*(R_i)$. The strategy is synthesized below where initially we set $k = \text{drk}(A^+)$.

```

Given ODG( $X_i$ ) compute ODG( $Y_{i+1}$ ); set cond=true;
while (cond) { set  $X_{i+1} = \text{trunc}_k(Y_i)$ ,  $R_{i+1} = I - AX_{i+1}$ ,
               set cond=( $\rho^*(R_{i+1}) > \rho^*(R_i)$ ),
               if (cond) then  $k=k+h$  }

```

In certain cases, due to the approximate estimation of $\rho(R)$ this strategy detects too late the divergence condition so that the increase of the cutting level is not sufficient anymore to repair the damage. For this reason we introduce the more conservative strategy of backtracking, where, in case of presumed divergence, besides increasing the cutting level, we move back a given number s of steps.

```

Given ODG( $X_i$ ) compute ODG( $Y_{i+1}$ ); set cond=true;
while (cond) { set  $X_{i+1} = \text{trunc}_k(Y_i)$ ,  $R_{i+1} = I - AX_{i+1}$ ,
               set cond=( $\rho^*(R_{i+1}) > \rho^*(R_i)$ ),
               if (cond) then  $k = k + h$ ,  $i = i - s$  }

```

With the above strategy the number of failures is substantially reduced, but in certain cases backtracking is applied, even if unneeded, where the previous strategy would still work. Often, this makes the iteration move back to the first step. A more refined strategy is the following. When the condition $\rho^*(R_{i+1}) > \rho^*(R_i)$ is detected, the iteration is continued in a “monitored” stage with the cutting level unchanged and with no backtracking, until either $\rho^*(R_i)$ becomes again a nondecreasing function of i or one of the following conditions is satisfied: the number of monitored iterations reaches a given value, the value $\rho^*(R_i)$ exceeds 1. If one of the latter two cases is encountered, the backtracking strategy with the increase of the cutting level is applied. In this case we move back as many steps as the number of monitored steps.

The last strategy that we introduce is the same as before with the only difference that in case where $\rho^*(R_i) > 1$ or the maximum number of monitored iterations has been reached then the iteration starts again from X_0 with cutting level $k = k + h$.

6 Implementation and numerical experiments.

The code has been written in C++. The algorithm has been designed around four classes which are the models of the main objects of the algorithm: the DRFFT(IRFFT) which represents the DFT (IFT) of a vector or a matrix, the

`Matrix` which models a general matrix of double precision real numbers and the `AODR` which models the Approximated Orthogonal Displacement Representation of a Toeplitz-like matrix. The `Matrix` object has been equipped with methods which are wrappers of the routines for the *QR* factorization and *SVD* decomposition of the LAPACK and with wrappers of the routines for the *product*, *copy* of matrices of the ATLAS Project. The *direct (inverse) FFTs* have been designed around the *double precision* routines of the *Fastest Fourier Transform in the West (FFTW)* [8]. We have chosen to represent the relation between the classes using the *composition* technique to combine the advantages of the O-O programming paradigm and the performance of the *C* language. The class `AODR` is defined by a `Matrix`, a *vector of doubles* and six *containers* for the FFTs used in the multiplication algorithm.

6.1 Efficiency of the implementation

All experiments were performed on a PC having an Intel Pentium III microprocessor running at 866MHz and 512 GBytes of RAM. To measure the efficiency of the implementation, we performed 100 iteration steps each for sizes of matrices $n = 2^{10}, 2^{11}, \dots, 2^{18}$. We observed that the execution time is proportional to the cutting level. The dominant term in the complexity turns out to be $5n \log(2n)$ (the cost of a real FFT of length $2n$). If we divide the execution time per iteration step by this factor (and by the cutting level), we get a number γ which ranges between 0.2 and 0.5 μsec .

6.2 Convergence tests

The algorithm has been implemented with the different strategies. We only report on the last strategy of the previous section. We considered the following classes of $n \times n$ Toeplitz matrices.

Class 1: Take n equidistant points between 1 and $2 \cdot 10^\alpha$. Consider the vector v whose elements are these points ordered from small to large. If we compute the FFT of this vector, we obtain the first column of a Toeplitz (circulant) matrix having condition number $2 \cdot 10^\alpha$. If we take the real part of this matrix, it turns out that the condition number approximates 10^α for large n . From the experiments, it turns out that, in this case, the number of iterations is independent of the size of the matrix and equals approximately $7 + 6\alpha$ for $\alpha = 1, 2, \dots, 5$. When α is taken larger than 5 sometimes convergence does not occur anymore. The cutting level stays equal to 2.

Class 2: The elements of vector v are now taken uniformly random between 0 and $\log_{10}(2) + \alpha$. We take these elements as the exponents of 10. Taking the FFT and the real part of this vector, we obtain a Toeplitz (circulant) matrix whose condition number is also approximating 10^α when n is large. The number of iteration steps is now $20 + 6\alpha$ for $\alpha = 1, 2, \dots, 5$. The cutting level also remains 2.

6.3 An application to image restoration

We considered an $n \times n$ image $X = (x_{i,j})$, with $n = 256$ and blurred it by means of the separable Point Spread Function $p_{i,j} = 1/(2q+1)^2$, $i, j = -9, \dots, 9$ where $q = 9$ so that $\sum_{i,j} p_{i,j} = 1$. In this way for the blurred noisy image $Y = (y_{i,j})$ we

have $y_{i,j} = \sum_{r,s=-q}^q p_{r,s} x_{i+r,j+s} + \eta_{i,j}$, $i, j = q+1, \dots, 256-q$, where we chose as $\eta_{i,j}$ the errors generated by rounding the blurred image to its integer part. Since the PSF is separable, the problem is modeled by the equation $TXT^T = Y - E$, where $E = (\eta_{i,j})$ and T is the $m \times n$ banded Toeplitz matrix with $m = 256 - 2q$ and with entries $t_{i-j} = 1/(2q + 1)$ for $-q \leq i - j \leq q$, $t_{i-j} = 0$ otherwise. The least square solution of minimal norm is therefore $T^+(Y - E)(T^+)^T$. We have approximated X with the matrix $T_k Y T_k^T$, where T_k is obtained by means of the approximate Newton iteration where we used the cutting level $k = \text{drk}(A^+) = 3$. The original, noisy and restored images obtained with 8 and 10 iterations are reported below. They correspond to the images obtained after 256 and 1024 steps, respectively, of Landweber's iteration.



References

- [1] A. BEN-ISRAEL, A note on iterative method for generalized inversion of matrices. *Math. Comp.*, 20:439–440, 1966.
- [2] M. BERTERO AND P. BOCCACCI, *Introduction to Inverse Problems in Imaging*, IOP Publishing, Bristol, 1998.

- [3] D. BINI, A. FARUSI, G. FIORENTINO AND B. MEINI, *On the regularized solution of block banded block Toeplitz systems*, in Proceedings of SPIE, Advanced Signal Processing Algorithms, Architectures, and Implementations IX, Vol. 4116, F. T. Luk Editor, pp. 135-146, 2000.
- [4] D. BINI AND B. MEINI, *Approximate displacement rank and applications*, in Structured Matrices in Mathematics, Computer Science, and Engineering II,, V. Olshevsky Editor, Contemporary Mathematics 281, pp.215-232, American Mathematical Society, Rhode Island, 2001 , AMS, 1999.
- [5] D. BINI AND B. MEINI, *Solving block banded block Toeplitz systems with structured blocks: algorithms and applications*, in Structured matrices: recent developments in theory and computation, D. A. Bini, E. Tyrtyshnikov, and P. Yalamov, eds., NOVA Science, New York 2001, pp. 21-42.
- [6] D. BINI AND V. PAN, *Matrix and polynomial computations, vol. 1: Fundamental algorithms*, Birkäuser, Boston, 1994.
- [7] R. CHAN AND M. NG, *Conjugate gradient methods for Toeplitz systems*, SIAM Review, (1996), pp. 427-482.
- [8] M. FRIGO AND S. G. JOHNSON, *FFTW Manual*, Massachusetts Institute of Technology, 1999.
- [9] G. H. GOLUB AND C. F. VAN LOAN, *Matrix computations*, Mathematical Sciences, Johns Hopkins University Press, third ed., 1996.
- [10] M. GU, *Stable and efficient algorithms for structures systems of linear equations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 279-306.
- [11] G. HEINIG AND K. ROST, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin, and Birkhäuser, Basel/Stuttgart, (1984).
- [12] T. KAILATH, S. KUNG, AND M. MORF, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395-407.
- [13] T. KAILATH AND A. SAYED, *Displacement structure: Theory and applications*, SIAM Review, 37 (1995), pp. 297-386.
- [14] V. PAN, S. BRANHAM, R. ROSHOLT, AND A.-L. ZHENG, *Newton's iteration for structured matrices*, in Fast reliable algorithms for matrices with structure, T. Kailath and A. Sayed, eds., SIAM, Philadelphia, 1999, ch. 7, pp. 189-210.
- [15] M. VAN BAREL AND P. KRAVANJA, *A stabilized superfast solver for indefinite Hankel systems*, Linear Algebra and its Applications, (1998), pp. 335-355.