Journal of Computational Mathematics, Vol.24, No.5, 2006, 647–656.

MODIFIED NEWTON'S ALGORITHM FOR COMPUTING THE GROUP INVERSES OF SINGULAR TOEPLITZ MATRICES [*1)]

Jian-feng Cai
(*Department of Mathematics, The Chinese University of Hong Kong, Hong Kong*)

Michael K. Ng
(*Department of Mathematics, Hong Kong Baptist University, Hong Kong*)

Yi-min Wei
(*School of Mathematical Sciences, Fudan University, Shanghai 200433, China & Key Laboratory of Mathematics for Nonlinear Sciences (Fudan University), Ministry of Education*)

Abstract

Newton's iteration is modified for the computation of the group inverses of singular Toeplitz matrices. At each iteration, the iteration matrix is approximated by a matrix with a low displacement rank. Because of the displacement structure of the iteration matrix, the matrix-vector multiplication involved in Newton's iteration can be done efficiently. We show that the convergence of the modified Newton iteration is still very fast. Numerical results are presented to demonstrate the fast convergence of the proposed method.

Mathematics subject classification: 15A09, 65F20.
Key words: Newton's iteration, Group inverse, Toeplitz matrix, Displacement rank.

1. Introduction

Let A be an $n \times n$ Toeplitz matrix [9, 12, 13], i.e.,

$$A = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-n} & a_{1-n} \\ a_1 & a_0 & a_{-1} & \ddots & a_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-2} & \ddots & a_1 & a_0 & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \end{bmatrix}.$$

The main aim of this paper is to modify Newton's iteration for the computation of the group inverse of A if A is singular with index 1, i.e.,

$$\operatorname{rank}(A) = \operatorname{rank}(A^2) < n.$$

The group inverse of A is the unique solution of the following three equations [1, 4, 14]

$$A^2X = A \quad XAX = X \quad \text{and} \quad AX = XA$$

and we denote it A_g throughout the paper. The application of the group inverses of matrices can be found in the field of Markov chains [4] and numerical analysis [5, 15, 16, 18, 19]. The

* Received January 12, 2005; final revised March 20, 2006.
1) Research supported in part by the National Natural Science Foundation of China under grant 10471027 and Shanghai Education Committee, RGC 7046/03P, 7035/04P, 7045/05P and HKBU FRGs.

computation of the inverses and the Moore-Penrose inverses of structured matrices [3, 2, 17] is a recent interesting problem in matrix computation, see for instance [6].

In this paper, Newton's iteration is modified for the computation of the group inverses of singular Toeplitz matrices. At each iteration, the iteration matrix is approximated by a matrix with a low displacement rank. Because of the displacement structure of the iteration matrix, the matrix-vector multiplication involved in Newton's iteration can be done efficiently. We show that the convergence of the modified Newton iteration is still very fast.

The outline of this paper is as follows. In Section 2, we review the concept of ϵ-displacement rank [3]. In Section 3, we introduce the modified Newton iteration and show the convergence results. Numerical results in Section 4 are presented to demonstrate the fast convergence of the proposed method. In Section 5, we present some concluding remarks.

2. ϵ-Displacement Rank

2.1 Displacement Rank

The concept of displacement rank was introduced by Kailath and his coauthors (see [10]) for close-to-Toeplitz matrices and was systematically studied in the general case in [9]. There are many definitions for displacement rank. Here we briefly describe two of them that will be used in the following discussion. Let us denote

$$C^+ = \begin{bmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbf{R}^{n\times n}, \qquad C^- = \begin{bmatrix} 0 & & & -1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} \in \mathbf{R}^{n\times n}.$$

The displacements for an n-by-n matrix B can be defined by

$$\Delta^+(B) = C^+B - BC^-, \quad \Delta^-(B) = C^-B - BC^+.$$

The rank of $\Delta^+(B)$ (or $\Delta^-(B)$) is called the $(+)$-displacement (or $(-)$-displacement) rank of B and are denoted by $drk^+(B)$ (or $drk^-(B)$). For a Toeplitz matrix A, we have $drk^+(A) \le 2$ and $drk^-(A) \le 2$, see for instance [9].

The operators $\Delta^+(\cdot)$ and $\Delta^-(\cdot)$ are both invertible. If we know the displacement of B, then B can be recovered by the sum of a series of products of circulant matrix and anti-circulant matrices. Let $\mathbf{h} = [h_1, h_2, \cdots, h_n]^T$ and $C^+(\mathbf{h})$ be a circulant matrix with its first column $\mathbf{h}$:

$$C^+(\mathbf{h}) = \begin{bmatrix} h_1 & h_n & \cdots & h_3 & h_2 \\ h_2 & h_1 & h_n & \ddots & h_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{n-1} & \ddots & h_2 & h_1 & h_n \\ h_n & h_{n-1} & \cdots & h_2 & h_1 \end{bmatrix}$$

and $C^-(\mathbf{h})$ be an anti-circulant matrix with its first column $\mathbf{h}$:

$$C^-(\mathbf{h}) = \begin{bmatrix} h_1 & -h_n & \cdots & -h_3 & -h_2 \\ h_2 & h_1 & -h_n & \ddots & -h_3 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{n-1} & \ddots & h_2 & h_1 & -h_n \\ h_n & h_{n-1} & \cdots & h_2 & h_1 \end{bmatrix}.$$

Theorem 1. *[3] We have the following results:*

$$\Delta^+(B) = \sum_{i=1}^{k} \mathbf{g}_i \mathbf{h}_i^T \quad \Longleftrightarrow \quad B = \frac{1}{2} \sum_{i=1}^{k} C^+(\mathbf{g}_i) C^-(J\mathbf{h}_i), \tag{1}$$

$$\Delta^-(B) = \sum_{i=1}^{k} \mathbf{g}_i \mathbf{h}_i^T \quad \Longleftrightarrow \quad B = -\frac{1}{2} \sum_{i=1}^{k} C^-(\mathbf{g}_i) C^+(J\mathbf{h}_i), \tag{2}$$

where J is the permutation matrix having 1 on the anti-diagonal.

It is well-known that both circulant and anti-circulant matrices can be diagonalized by the discrete Fourier transform matrix F [3, 13], more precisely, we have

$$\begin{aligned} &C^+(\mathbf{h}) = F\mathrm{Diag}(\mathbf{g})F^H \\ &C^-(\mathbf{h}) = DF\mathrm{Diag}(\widehat{\mathbf{g}})F^H D \\ &\mathbf{g} = F^H\mathbf{h} \quad \widehat{\mathbf{g}} = F^H D\mathbf{h}. \end{aligned} \tag{3}$$

Here D is a suitable diagonal matrix. From (3), we find that both matrix-vector multiplications $C^+(\mathbf{h})\mathbf{x}$ and $C^-(\mathbf{h})\mathbf{x}$ can be done efficiently by fast Fourier transforms (FFTs), i.e., the number of operations required are $O(kn\log n)$.

2.2 ϵ-Displacement Rank

The ϵ-displacement rank [3] is about the rank of the perturbed displacement. More precisely, it can be defined as follows:

Definition 1. *For a given $\epsilon > 0$ define the ϵ-(+)-displacement rank of a matrix B as*

$$drk_\epsilon^+(B) = \min_{\|E\| \le \epsilon} \mathrm{rank}[\Delta^+(B) + E]$$

and the ϵ-(−)-displacement rank of a matrix B as

$$drk_\epsilon^-(B) = \min_{\|E\| \le \epsilon} \mathrm{rank}[\Delta^-(B) + E].$$

Let

$$\Delta^+(B) = U^+\Sigma^+(V^+)^T = \sum_{i=1}^{drk^+(B)} \sigma_i^+ \mathbf{u}_i^+ (\mathbf{v}_i^+)^T \tag{4}$$

and

$$\Delta^-(B) = U^-\Sigma^-(V^-)^T = \sum_{i=1}^{drk^-(B)} \sigma_i^- \mathbf{u}_i^- (\mathbf{v}_i^-)^T,$$

be the singular value decomposition of $\Delta^+(B)$ and $\Delta^-(B)$ respectively.

Theorem 2. *(i) Let $\sigma_1^+ \ge \cdots \ge \sigma_k^+ > 0$ be the nonzero singular values of $\Delta^+(B)$ and ϵ be such that $\epsilon < \sigma_1^+$. Then $drk_\epsilon^+ = r$ if and only if $\sigma_r^+ > \epsilon \ge \sigma_{r+1}^+$.*
(ii) Let $\sigma_1^- \ge \cdots \ge \sigma_k^- > 0$ be the nonzero singular values of $\Delta^-(B)$ and ϵ be such that $\epsilon < \sigma_1^-$. Then $drk_\epsilon^- = r$ if and only if $\sigma_r^- > \epsilon \ge \sigma_{r+1}^-$.

The proof of this theorem is similar to Theorem 2.1 in [3], so we omit it.

2.3 Matrix Approximation

With the help of the ϵ-displacement rank, a matrix B can be approximated by a matrix with small displacement rank. Let us take (+)-displacement as an example to illustrate it. According to Theorem 2, we know that the ϵ-displacement rank of B is r if and only if $\sigma_r^+ > \epsilon \geq \sigma_{r+1}^+$. It is well-known that the best rank r approximation of $\Delta^+(B)$ is given by

$$\sum_{i=1}^{r} \sigma_i^+ \mathbf{u}_i^+ (\mathbf{v}_i^+)^T. \tag{5}$$

Using (5),we approximate B by B_ϵ^+

$$B_\epsilon^+ = \frac{1}{2} \sum_{i=1}^{r} \sigma_i^+ C^+(\mathbf{u}_i^+) C^-(J\mathbf{v}_i^+).$$

It is clear that the (+)-displacement rank of B_ϵ^+ is smaller than that of B. Similarly, we use

$$B_\epsilon^- = -\frac{1}{2} \sum_{i=1}^{r} \sigma_i^- C^-(\mathbf{u}_i^-) C^+(J\mathbf{v}_i^-),$$

where $\sigma_r^- > \epsilon \geq \sigma_{r+1}^-$, to approximate B.

The following theorem gives the distances between B and B_ϵ^+, and B and B_ϵ^-. We use these results to show the convergence theory of the modified Newton iteration.

Theorem 3. *(i) Let $r = drk_\epsilon^+(B) \leq drk^+(B) = k$, and let $\sigma_1^+ \geq \cdots \geq \sigma_k^+ > 0$ be the nonzero singular values of $\Delta^+(B)$. Then it holds*

$$\|B - B_\epsilon^+\| \leq \frac{1}{2} n \sum_{i=r+1}^{k} \sigma_i^+ \leq \frac{1}{2} n(k-r)\epsilon.$$

(ii) Let $r = drk_\epsilon^-(B) \leq drk^-(B) = k$, and let $\sigma_1^- \geq \cdots \geq \sigma_k^- > 0$ be the nonzero singular values of $\Delta^-(B)$. Then it holds

$$\|B - B_\epsilon^-\| \leq \frac{1}{2} n \sum_{i=r+1}^{k} \sigma_i^- \leq \frac{1}{2} n(k-r)\epsilon.$$

Proof. For any $\mathbf{h}$ such that $\|\mathbf{h}\| = 1$, it holds that

$$\|C^+(\mathbf{h})\| = \|\sum_{i=1}^{n} h_i (C^+)^{i-1}\| \leq \sum_{i=1}^{n} |h_i| \ \|(C^+)^{i-1}\| = \sum_{i=1}^{n} |h_i| \leq \sqrt{n}\|\mathbf{h}\| = \sqrt{n}$$

and

$$\|C^-(\mathbf{h})\| = \|\sum_{i=1}^{n} h_i (C^-)^{i-1}\| \leq \sum_{i=1}^{n} |h_i| \ \|(C^-)^{i-1}\| = \sum_{i=1}^{n} |h_i| \leq \sqrt{n}\|\mathbf{h}\| = \sqrt{n}.$$

From (4) and (5), we have

$$\|B - B_\epsilon^+\| = \frac{1}{2} \sum_{i=r+1}^{k} \sigma_i^+ \|C^+(\mathbf{u}_i^+)\| \ \|C^-(J\mathbf{v}_i^+)\| \leq \frac{1}{2} n \sum_{i=r+1}^{k} \sigma_i^+ \leq \frac{1}{2} n(k-r)\epsilon.$$

Thus we obtain the first assertion. The second assertion can be proved analagously.

3. The Proposed Algorithm

3.1 Newton's Iteration

The classical Newton iteration for computing the inverse or the generalized inverse of A is given by

$$X_{i+1} = 2X_i - X_iAX_i, \quad i = 0, 1, 2, \cdots \tag{6}$$

If A is singular, the sequence $\{X_i\}$ may converge to different generalized inverses, and the limit depends on the initial guess X_0. For example, $\{X_i\}$ converges to the Moore-Penrose inverse if $X_0 = \alpha A^T$ (see [1]), or the group inverse if $X_0 = \alpha A$ (see [5]).

Note that A is a Toeplitz matrix. The initial guess X_0 has the low displacement rank. The limit X_∞, the inverse or the generalized inverse of A, has also the low displacement rank, see [8, 9, 19]. However, the iteration matrices X_i in the Newton iteration may not have low displacement rank. In the worst case, the displacement rank of X_i can increase exponentially. It follows that the displacement rank of X_i must be controlled in order to develop an efficient Newton's iteration.

By using ϵ-displacement rank, Bini and his coauthors in [3, 2] modified the classical Newton iteration to compute the inverses of nonsingular Toeplitz matrices, and the Moore-Penrose inverses of rectangular full-column rank Toeplitz matrices. For a given Toeplitz matrix A, they set

$$X_0 = \alpha A^T$$

and construct the sequence $\{X_i\}$ as follows:

$$Y_i = 2X_i - X_iAX_i, \quad \text{and} \quad X_{i+1} = (Y_i)_{\epsilon_i}, \tag{7}$$

where $(Y_i)_{\epsilon_i}$ is a low displacement rank approximation of Y_i. We note that the displacement rank of X_i is controlled via ϵ_i.

If we directly apply their method to compute the generalized inverses of rank-deficient Toeplitz matrices, the sequence $\{X_i\}$ may converge to other generalized inverses. For example, the group inverse, whose entries are corrected to 4 places after the decimal, of a Toeplitz matrix ($n = 12$) in Section 4 is given by

$$\begin{bmatrix}
0.2707 & -0.0045 & -0.0051 & -0.0058 & -0.0068 & -0.0083 & -0.0105 & -0.0143 & -0.0220 & -0.0430 & -0.2554 & 0.2707 \\
-0.2554 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.2554 \\
-0.0430 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0430 \\
-0.0220 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0220 \\
-0.0143 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0143 \\
-0.0105 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0105 \\
-0.0083 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0083 \\
-0.0068 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0068 \\
-0.0058 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0058 \\
-0.0051 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0051 \\
-0.0045 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0045 \\
0.2707 & -0.0045 & -0.0051 & -0.0058 & -0.0068 & -0.0083 & -0.0105 & -0.0143 & -0.0220 & -0.0430 & -0.2554 & 0.2707
\end{bmatrix}.$$

By using (7), the computed group inverse, whose entries are also corrected to 4 places after the decimal, is given by

$$\begin{bmatrix}
-34.9772 & -0.0108 & -0.0066 & -0.0046 & -0.0024 & -0.0108 & -0.0109 & -0.0130 & -0.0249 & -0.0447 & -0.2503 & 35.5217 \\
-0.2532 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.2577 \\
-0.0495 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0366 \\
-0.0163 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0277 \\
-0.0149 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0138 \\
-0.0101 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0109 \\
-0.0073 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & -0.0137 & -0.0093 \\
-0.0156 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0116 & 0.0019 \\
0.0058 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & -0.0101 & -0.0174 \\
-0.0115 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0090 & 0.0013 \\
-0.0042 & -0.0101 & -0.0116 & -0.0137 & -0.0166 & -0.0211 & -0.0287 & -0.0440 & -0.0861 & -0.5109 & 1.0828 & -0.0048 \\
35.5193 & 0.0018 & -0.0035 & -0.0071 & -0.0113 & -0.0058 & -0.0102 & -0.0157 & -0.0190 & -0.0414 & -0.2606 & -34.9810
\end{bmatrix}.$$

It is clear that there is a big difference between the two matrices. For instance, the (1,1)-th entry of the group inverse is 0.2707, while the result of (7) is -34.9772. In [17], we have studied how

to modify the Newton algorithm to deal with rank-deficient Toeplitz matrices and to compute their Moore-Penrose inverses.

3.2 The Modified Method

If we set the initial guess as the form

$$X_0 = \alpha ABA$$

where α is a scalar and B is a matrix, then the Newton iteration (6) will converge to the group inverse A_g of A. We note that each iteration matrix X_i is of the form $X_i = AY_iA$. In particular, the iteration (6) can be written as follows:

$$Y_{i+1} = 2Y_i - Y_iA^3Y_i, \quad X_i = AY_iA, \quad i = 0, 1, 2, \cdots.$$

We modify the Newton iteration by approximating Y_{i+1} in the above iterative process with a low displacement rank matrix. We set

$$X_0 = \alpha ABA, \quad \text{i.e.,} \quad Y_0 = \alpha B,$$

and $\{X_i\}$ is defined by

$$Z_i = 2Y_i - Y_iA^3Y_i, \quad Y_{i+1} = (Z_i)^-_{\epsilon_i}, \quad \text{and} \quad X_{i+1} = AY_{i+1}A, \tag{8}$$

where α is a scalar such that the spectral radius $\rho(AA_g - AX_0) < 1$ and $\rho(A_gA - X_0A) < 1$. There are many choices of B to achieve such above objective. For instance, we can choose $B = (A^3)^T$, and therefore $AX_0 = A^2(A^3)^TA$. If the parameter α is equal to $1/\rho(A^3(A^T)^3)$, then $\rho(AA_g - AX_0) < 1$. We note that the spectral radius of $A^3(A^T)^3$ can be approximated by the power method [11] which in fact requires a few steps of matrix-vector multiplication with a low cost. In the next section, this stragey will be employed to conduct in our experiments.

In our modified Newton's iteration we compute and store the SVD of $\Delta^-(Y_i)$ instead of Y_i. Suppose that the SVD of $\Delta^-(Y_i)$ is $U_{Y_i}\Sigma_{Y_i}V_{Y_i}^T$. Noting that

$$\Delta^-(Z_i) = 2\Delta^-(Y_i) - \Delta^-(Y_i)A^3Y_i - Y_i\Delta^+(A^3)Y_i - Y_iA^3\Delta^-(Y_i), \tag{9}$$

we have $drk^-(Z_i) \le k + 2drk^-(Y_i)$ where $k = drk^+(A^3) \le 6$. Suppose that the SVD of $\Delta^+(A^3)$ is $U_A\Sigma_AV_A^T$. Rewriting (9) into matrix form, we obtain

$$\Delta^-(Z_i) = U_{Z_i}\Sigma_{Z_i}V_{Z_i}^T = [\,U_{Y_i} \quad Y_iA^3U_{Y_i} \quad Y_iU_A\,] \begin{bmatrix} 2\Sigma_{Y_i} & 0 & -\Sigma_{Y_i} \\ -\Sigma_{Y_i} & 0 & 0 \\ 0 & -\Sigma_A & 0 \end{bmatrix} \begin{bmatrix} V_{Y_i}^T \\ V_A^TY_i \\ V_{Y_i}^TA^3Y_i \end{bmatrix}. \tag{10}$$

The SVD of of $\Delta^-(Y_{i+1}) = \Delta^-((Z_i)^-_{\epsilon_i})$ can be computed by the following algorithm:

Algorithm 1. Compute the SVD of $\Delta^-(Y_{i+1})$

Input: The SVD of $\Delta^-(Y_i)$: $U_{Y_i}\Sigma_{Y_i}V_{Y_i}^T$, and the truncation value ϵ_i.

Output: The SVD of $\Delta^-(Y_{i+1})$: $U_{Y_{i+1}}\Sigma_{Y_{i+1}}V_{Y_{i+1}}^T$.

Computation:

1. Compute the SVD of $\Delta^-(Z_i)$ by the following steps.
   (a) Compute the matrices $U_{Z_i}, V_{Z_i}, \Sigma_{Z_i}$ according to (10).
   (b) Compute the QR decompositions $U_{Z_i} = Q_1R_1$ and $V_{Z_i} = Q_2R_2$.

(c) Compute the SVD of $R_1\Sigma_{Z_i}R_2^T = U\Sigma V$, where $\Sigma = diag(\sigma_1, \cdots, \sigma_k)$ and determine r such that $\sigma_{r+1} \le \epsilon_i < \sigma_r$.

2. Compute the SVD of $\Delta^-(Y_{i+1})$ by the following step.

(a) Set $\Sigma_{Y_{i+1}} = \mathrm{diag}(\sigma_1, \cdots, \sigma_r)$, $U_{Y_{i+1}} = Q_1U(:, 1:r)$ and $V_{Y_{i+1}} = Q_2V(:, 1:r)$.

The above algorithm costs about $O(h(k+h))$ FFTs, $O((2h+k)^2n)$ and $O((2h+k)^3)$ extra flops, where $h = drk^-(Y_i)$ and $k = drk^+(A^3) \le 6$. It is clear that the computational cost per iteration step is not expensive if h is small. Moreover, if h is independent of n, the algorithm is of $O(n \log n)$ operations.

Now we can complete our modified Newton's iteration as follows:

Algorithm 2. Modified Newton's iteration to approximate A_g.

Input: The first row and column vectors of the Toeplitz matrix A.

Output: An approximation X of A_g given in terms of AYA where Y is given by its $(-)$−displacement $U_Y\Sigma_Y V_Y^T$ in SVD form.

Computation:

1. Compute the SVD of $\Delta^+(A^3)$.
2. Choose α and B such that $\|AA_g - \alpha A^2BA\| < 1$ and compute the SVD of $\Delta^-(\alpha X)$.
3. Determine an ϵ_i and compute the SVD of $\Delta^-(Y_i)$ by means of Algorithm 1.
4. Let $X_i = AY_iA$. If the residual is small enough, then goto Step 3, otherwise output the result.

Theorem 4. *Let X_0 be such that $\|AA_g - AX_0\| \le 1 - \theta$. Let $R_i = AA_g - AX_i$ and $\overline{R_i} = A_gA - X_iA$ be the residual sequences. If $\epsilon_i = \frac{\min(\|R_i\|^2, \|\overline{R_i}\|^2)\theta}{n(2h_i+k)\|A\|^3}$, where $h_i = drk^-(Y_i)$ and $k = drk^+(A^3)$. Then it holds $\|R_i\|^2 \le (1 - \frac{\theta}{2})^{2^i}$ and $\|\overline{R_i}\|^2 \le (1 - \frac{\theta}{2})^{2^i}$.*

Proof. We denote $E_i = Z_i - Y_{i+1}$, thus,

$$\|E_i\| \le \frac{1}{2}n(2h_i + k)\epsilon_i = \frac{\min(\|R_i\|^2, \|\overline{R_i}\|^2)\theta}{2\|A\|^3}$$

by Theorem 3. Noting (8), we obtain

$$\begin{aligned}
R_{i+1} &= AA_g - AX_{i+1} = AA_g - AAY_{i+1}A \\
&= AA_g - AAZ_iA + AAE_iA \\
&= AA_g - AA(2Y_i - Y_iAAAY_i)A + AAE_iA \\
&= AA_g - AAY_iA - AAY_iA + AAY_iAAAY_iA + AAE_iA \\
&= AA_gAA_g - AA_gAAY_iA - AAY_iAAA_g + AAY_iAAAY_iA + AAE_iA \\
&= (AA_g - AAY_iA)^2 + AAE_iA \\
&= R_i^2 + AAE_iA.
\end{aligned}$$

Therefore,

$$\|R_{i+1}\| \le \|R_i\|^2 + \|A\|^3\|E_i\| \le \|R_i\|^2\left(1 + \frac{\theta}{2}\right).$$

Table 1: Results for the modified Newton's iteration

n	$Nstep$	$Mdrk$	$Sdrk$	Time (seconds)	Time (seconds) by the standard Newton
32	20	10	112	6.2	0.2
64	22	11	132	9.4	0.4
128	23	13	146	13.0	2.3
256	24	12	153	18.0	7.0
512	25	13	162	19.0	35.0
1024	26	14	174	53.0	190.0
2048	27	14	184	96.0	1500.0
4096	28	15	197	360.0	>10000.0
8192	29	15	205	780.0	Out of Memory
16384	29	15	205	1.6e+3	Out of Memory

Whence we obtain

$$\|R_i\| \le ((1-\theta)(1+\theta/2))^{2^i} \le \left(1-\frac{\theta}{2}\right)^{2^i}.$$

The assertion for the sequence $\{\overline{R_i}\}$ can be proved analagously.

According to above theorem, it reveals the quadratic convergence of our modified Newton iteration for the certain ϵ_i. However, in practice, a larger ϵ_i is chosen in order to reduce the computational cost of each step.

In the next section, the numerical results shows that the method still has fast convergence. On the other hand, it is expensive to compute the residual sequence since A_g is not known in advance. A cheap method is to compute the $res(X_i)$ defined by

$$res(X_i) = \max\{\|(A - A^2X_i)e_1\|, \|(X_i - X_iAX_i)e_1\|, \|(AX_i - X_iA)e_1\|\}. \tag{11}$$

Each term can be computed by a few FFTs. We also use this strategy in our numerical experiments.

4. Numerical Results

In this section, we test the proposed algorithm for computing the group inverse of a Toeplitz matrix, whose its first column is $(1, 1/2, \cdots, 1/(n-1), 1)$ and the first row is chosen such that the last column is same as the first column. We can check that this Toeplitz matrix has index 1. The group inverse of this Toeplitz matrix is given by the following formula:

$$A_g = \begin{bmatrix} I \\ e_1^T \end{bmatrix} (I + e_1e_1^T)^{-1}C^{-1}(I + e_1e_1^T)^{-1} \begin{bmatrix} I & e_1 \end{bmatrix}, \tag{12}$$

where C is a circulant matrix whose the first column is given by $(1, 1/2, \cdots, 1/(n-1))^T$, I is the $(n-1)$-by-$(n-1)$ identity matrix, and e_1 is the first column of the identity matrix I.

The truncation value ϵ_i is set to be $res(X_i)/\|A\|^4$ throughout the tests. When $res(X_i) < 10^{-6}$, the iteration is stopped. In Table 1, we report the number of iterations $Nstep$ required for the iteration which indicates the convergence, the sum drk^- of Y_i ($Sdrk$) which indicates the computational costs for the whole iteration, and the maximum drk^- of Y_i ($Mdrk$) which indicate the maximum computational costs per step. The CPU time used by the modified Newton method and by the original Newton method are also reported in Table 1.

From the table, we see that both $Mdrk$ and $Nstep$ increase very slowly as the matrix size increases. Therefore, the computational cost per step of our method is very low, and the convergence is very fast. The latter fact can be also seen from the curve of convergence history plotted in Figure 1.

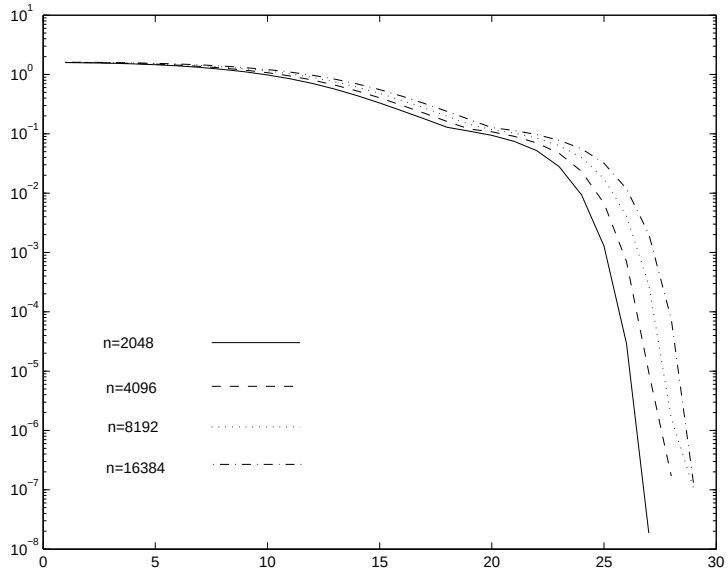

Figure 1: Convergence of the modified Newton iteration

5. Concluding Remarks

We have modified Newton's iteration to compute the group inverses A_g of singular Toeplitz matrices A with index one. In our method, the matrix in the original Newton's iteration is approximated with a low displacement rank. The modification guarantees that the iteration converges to the group inverse, not to other unknown generalized inverses. Moreover, the computation cost of per step has been greatly deduced due to the low displacement rank approximation. With suitable assumptions, quadratic convergence has been established. The numerical experiments has shown that our method is very effective.

It is straightforward to extend our method to compute the Drazin inverse [7, 20] with arbitrary index, since these two kinds of generalized inverse are very similar and the group inverse is the special case of Drazin inverse.

Acknowledgement. The authors would like to thank the referees for their useful suggestions.

References

[1] A. Ben-Israel and T.N.E. Greville, Generalized Inverses: Theory and Applications, John Wiley, New York, 1974; 2nd Edition, Springer-Verlag, New York, 2003.

[2] D. Bini, G. Codevico and M. Van Barel, Solving Toeplitz least squares problems by means of Newton's iteration, *Numer. Algorithms,* **33** (2003), 93-103.

[3] 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.

[4] S.L. Campbell and C.D. Meyer, Generalized Inverses of Linear Transformations, London, Pitman, 1979; Dover, New York, 1991.

[5] X. Chen and R. Hartwig, The hyperpower iteration revisited, *Linear Algebra Appl.,* **233** (1996), 207–229.

[6] G. Codevico, V. Pan, M. Van Barel, X. Wang and A. Zheng, Newton-like Iteration for General and Structured Matrices, *preprint*, 2003.

[7] H. Diao, Y. Wei and S. Qiao, Displacement rank of the Drazin inverse, *J. Comput. Appl. Math.,* **167** (2004), 147–161.

[8] G. Heinig and F. Hellinger, Displacement structure of pseudoinverses, *Linear Algebra Appl.*, **197/198** (1994), 623–649.

[9] G. Heinig and K. Rost, Algebraic Methods for Toeplitz-like Matrices and Operators, Birkhauser Verlag, Berlin, 1984.

[10] T. Kailath, S.Y. Kung and M. Morf, Displacement rank of matrices and linear equations, *J. Math. Anal. Appl.,* **68** (1979), 395–407.

[11] G. Golub and C. Van Loan, Matrix Computations, The John Hopkins University Press, Third Edition, 1996.

[12] X. Jin, Developments and Applications of Block Toeplitz Iterative Solvers, Kluwer Academic Publishers Group, Dordrecht and Science Press, Beijing, 2002.

[13] M.K. Ng, Iterative Methods for Toeplitz Systems, Oxford University Press, New York, 2004.

[14] G. Wang, Y. Wei and S. Qiao, Generalized Inverses: Theory and Computations, Science Press, Beijing, 2004.

[15] Y. Wei, On the peturbation of the group inverse and oblique projection, *Appl. Math. Comput.,* **98** (1999), 29–42.

[16] Y. Wei, Perturbation analysis of singular linear systems with index one, *Int. J. Comput. Math.,* **74** (2000), 483–491.

[17] Y. Wei, J. Cai and M.K. Ng, Computing Moore-Penrose inverses of Toeplitz matrices by Newton's iteration, *Math. Comput. Model.,* **40** (2004), 181–191.

[18] Y. Wei and H. Diao, On group inverse of singular Toeplitz matrices, *Linear Algebra Appl.*, **399** (2005), 109–123.

[19] Y. Wei and M. Ng, Displacement structure of group inverse, *Numer. Linear Algebra Appl.,* **12** (2005), 103–110.

[20] Y. Wei, H. Diao and M. Ng, On Drazin inverse of singular Toeplitz matrix, *Appl. Math. Comput.,* **172** (2006), 809–817.