Pergamon

Computers Math. Applic. Vol. 30, No. 8, pp. 57–63, 1995

0898-1221/95 $9.50 + 0.00

0898-1221(95)00137-9

A Fast, Preconditioned Conjugate Gradient Toeplitz and Toeplitz-Like Solvers

V. PAN*
Mathematics and Computer Science Department
Lehman College, CUNY, Bronx, NY 10460, U.S.A.
and
Computer Science Department
The Graduate School and University Center, CUNY
33 West 42nd Street, New York, NY 10036, U.S.A.
vpan@lcvax.cuny.edu

A. L. ZHENG*, O. DIAS* AND X. H. HUANG*
Department of Mathematics
The Graduate School and University Center, CUNY
33 West 42nd Street, New York, NY 10036, U.S.A.

(Received January 1995; accepted March 1995)

Abstract—For a Toeplitz or Toeplitz-like matrix T, we define a preconditioning applied to the symmetrized matrix T^HT, which decreases the condition number compared to the one of T^HT and even the one of T. This enables us to accelerate the conjugate gradient algorithm for solving Toepiltz and Toeplitz-like linear systems, thus extending the previous results of [1], restricted to the Hermitian positive definite case. The extension relies on some recent formulae of Gohberg and Olshevsky for the inverses of Toeplitz-like matrices.

Keywords—Toeplitz systems of linear equations, Toeplitz solver, Toeplitz-like systems, Preconditioned conjugate gradient method, Inversion of Toeplitz-like matrices.

1. INTRODUCTION

We present a new approach to preconditioning of an unsymmetric Toeplitz matrix T, which substantially improves the solution of unsymmetric Toeplitz linear systems of n equations, by means of the conjugate gradient method. The approach also works for the more general class of Toeplitz-like linear systems too.

In contrast to the direct Toeplitz solvers using order of the n^2 or $n \log^2 n$ arithmetic operations [2–8], the conjugate gradient method requires $O(kn \log n)$ operations, where $k = k(T)$ is the condition number of T. Therefore, the method is particularly effective for well-conditioned Toeplitz linear systems, which motivates the search for good preconditioners that would decrease the condition number and preserve the Toeplitz structure.

In [1], such effective preconditioning was proposed for Hermitian (or real symmetric) positive definite (hereafter, h.p.d.) Toeplitz systems, based on factorization of T into the product

$$T = (T + \mu I) \left(I - \mu (T + \mu I)^{-1}\right)$$

*Supported by NSF Grant CRR 9020690 and PSC CUNY Award 664334 and 665301.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX for a scalar μ. The key idea of [1] is that an appropriate choice of the scalar μ defined by two extreme eigenvalues of T implies a substantial decrease of the condition number of both factors relatively to k and thus substantially accelerates the solution of an associated Toeplitz linear system. This algorithm, however (as well as other competitive iterative preconditioned Toeplitz solvers [9–12]), works neither for the unsymmetric nor for Toeplitz-like cases, which are also highly important in computational practice.

The present paper gives a desired extension of the algorithm of [1] to these cases. The extension relies on the properties of the circulant and skew-circulant displacement operators associated with Toeplitz and Toeplitz-like matrices and, in particular, on the recent explicit formulae expressing the displacement generators of the inverses of such matrices via few vectors associated with the inverses [13]. More specifically, we replace T by its symmetrization T^HT and respectively change the factorization. $T^HT+\mu I$ and $I-\mu\left(T^HT+\mu I\right)^{-1}$ are still Toeplitz-like matrices, which we represent by using their short displacement generators and the explicit formulae from [13]. This still enables fast multiplication of the matrix $I-\mu\left(T^HT+\mu I\right)^{-1}$ by a vector and leads to the desired extension of the algorithm of [1], defining fast Toeplitz-like solvers, in the case of an ill-conditioned input.

In our presentation, we try to follow the line of [1]. In the next section, we recall some relevant results on displacement representation of Toeplitz-like matrices. In Section 3, we show a general outline of the method. In Section 4, we specify various policies of choosing the parameter μ and their influence on the number of arithmetic operations required for the solution of Toeplitz and Toeplitz-like linear systems. In Section 5, we specify the more effective solver in the Toeplitz case.

2. SOME PROPERTIES OF TOEPLITZ-LIKE MATRICES

DEFINITION 2.1. *(Compare [14, Definition 2.11.1].) Let $F : F_{m,n} \to F_{m,n}$ be an operator, let $A \in F_{m\times n}$, and let $G \in F_{m\times l}$, $H \in F_{n\times l}$ denote two matrices such that $F(A) = GH^\top$. Then $l = \operatorname{rank}(F(A))$, the rank of the matrix $F(A)$, is called the F-rank of A, and the pair of the matrices G and H is called an F-generator of A of length l.*

Given a scalar $\phi \neq 0$, an $m \times m$ matrix X, and an $n\times n$ matrix Y, define the operator $F_{(X,Y)}(A) = A - XAY$ and specify a displacement operator of Toeplitz-type as follows:

$$F(A) = F_{(Z_\phi, Z^\top_{1/\phi})}(A) = A - Z_\phi A Z^\top_{1/\phi},$$
$$Z_\phi = \begin{bmatrix} 0 & . & . & . & \phi \\ 1 & 0 & . & . & . \\ 0 & 1 & . & . & . \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{1}$$

DEFINITION 2.2. *An $m\times n$ matrix is called a Toeplitz-like matrix if it has F-rank bounded from above by a constant independent of m and n, where F is the operator defined in (1).*

Hereafter, let $\phi = 1$, $Z = Z_1$. We have the following basic lemmas.

LEMMA 2.1. *[14] Let $A \in F_{n\times n}$, $B \in F_{m\times m}$ be two Toeplitz-like matrices given with their F-generators of lengths l_A and l_B, respectively. Then AB is a Toepiltz-like matrix having an F-generator of length $l_{AB} \le l_A + l_B$.*

PROOF. It follows from the observation that $F(AB) = F(A)B + ZAZ^\top F(B)$.

LEMMA 2.2. *(Compare [13–15].) Let A be a nonsingular Toeplitz-like matrix with an F-generator $F(A) = G_1H_1^\top$ of length l_A. Then A^{-1} is a Toeplitz-like matrix with an F-generator equal to* $GH^\top$, where $G = -A^{-1}G_1$, $H^\top = H_1^\top Z A^{-1} Z^\top$.

PROOF. Immediate.

From these results, we have the following corollary.

COROLLARY 2.1. *Let T be an $n \times n$ Toeplitz-like matrix with an F-generator of length l_T. Then $B = T^H T + \mu I$, $C = I - \mu B^{-1}$ are Toeplitz-like matrices with $l_B \le 2l_T$ and $l_C \le 2l_T$, provided that $-\mu$ is not an eigenvalue of $T^H T$.*

DEFINITION 2.3. *[14] An $m \times n$ matrix $\mathrm{Circ}_\phi(r) = \mathrm{Circ}_{(\phi,m,n)}(r) = [z_{ij}]$, for a vector $\mathbf{r} = [r_0, \dots, r_{m-1}]^\top$ and for a scalar $\phi \neq 0$, is called a ϕ-circulant matrix if $z_{i,j} = r_{i-j \bmod m}$ for $i \ge j$; $z_{i,j} = \phi r_{i-j \bmod m}$ for $i < j$.*

Hereafter, l will stand for l_T.

3. A CONDITION-IMPROVING MATRIX FACTORIZATION

LEMMA 3.1. *[1] Let A be an $n \times n$ matrix, $B = A + \mu I$, $C = I - \mu B^{-1}$. Then $A = BC = CB$. If $-\mu$ is not an eigenvalue of A, then both B and C have inverses, and $A^{-1} = C^{-1}B^{-1} = B^{-1}C^{-1}$.*

Let the eigenvalues of A, B and C be given by

$$\begin{aligned}\alpha_n &\le \alpha_{n-1} \le \cdots \le \alpha_1 = \lambda(A),\\ \beta_n &\le \beta_{n-1} \le \cdots \le \beta_1 = \lambda(B),\\ \gamma_n &\le \gamma_{n-1} \le \cdots \le \gamma_1 = \lambda(C).\end{aligned}$$

By the definition of B and C, we have

$$\beta_j = \alpha_j + \mu, \qquad \gamma_j = 1 - \mu\beta_j^{-1}.$$

LEMMA 3.2. *[1] Let A, B and C be as above and let $\mu > 0$. Then the condition numbers of B and C are given by*

$$k(B) = \frac{\alpha_1 + \mu}{\alpha_n + \mu} \qquad \text{and} \tag{2}$$

$$k(C) = \frac{\alpha_1}{\alpha_n}\left(\frac{\alpha_n + \mu}{\alpha_1 + \mu}\right), \tag{3}$$

so that for all $\mu > 0$, we have

$$k(A) = k(B)k(C). \tag{4}$$

LEMMA 3.3. *[1] Let $\mu = \sqrt{\alpha_1\alpha_n}$. Then $k(B) = k(C) = \sqrt{k(A)}$.*

4. A FAST TOEPLITZ-LIKE SOLVER

Consider the linear system

$$Tx = b, \tag{5}$$

where T is an $n \times n$ nonsingular Toeplitz-like matrix, given with its F-generator of length l. Apply the matrix factorization of the previous section to the linear system,

$$T^H T x = T^H b. \tag{6}$$

Let $A = T^H T$, then A is an $n \times n$ h.p.d. Toeplitz-like matrix, $l_A \le 2l$. Define $B = A + \mu I$, $C = I - \mu B^{-1}$. Suppose that $-\mu$ is not an eigenvalue of A. Then, by the results of the previous section, B and C are nonsingular Toeplitz-like matrices with $l_B \le 2l$ and $l_C \le 2l$. By the results of [13], B^{-1} is completely defined by its last row and its F-generator:

$$B^{-1} = \mathrm{Circ}_{l_r} + \frac{1}{1-\phi} \sum_{m=1}^{2\alpha} \mathrm{Circ}_{\phi}\left(r_m\right) \mathrm{Circ}_1\left(s_m^{\top}\right), \tag{7}$$

where ϕ is arbitrary, $\phi \neq 1$, Circ_{l_r} is the 1-circulant matrix with the last row equal to $y^{\top}$. Furthermore, r_m, s_m and $y^{\top}$ satisfy the following equations:

$$Br_m = g_m, \tag{8}$$

$$Bt_m = -Z_1^{\top} h_m, \tag{9}$$

$$s_m = Z_1 t_m, \qquad m = 1, 2, \dots, 2l, \tag{10}$$

$$By = e_{n-1}, \qquad e_{n-1} = (0, 0, \dots, 1)^{\top}, \tag{11}$$

where $G = [g_1, \dots, g_{2l}]$, $H = [h_1, \dots, h_{2l}]$ of A. Therefore, we have the following algorithm:

ALGORITHM 1.

Input: *An $n \times n$ nonsingular Toeplitz-like matrix T, a vector b, and a shift value μ.*
Output: $T^{-1}b$.
Stage 1: *Solve the equations (8), (9), (10) and (11).*
Stage 2: *Solve $Bz = T^H b$.*
Stage 3: *Solve $Cx = z$; return x.*

We use conjugate gradient (CG) method [16] to obtain the solution at Stages 1 and 3 in n_B and n_C iteration steps, respectively. Stage 2 amounts to $2\alpha + 1$ multiplications of f-circulant matrices by vectors for $f = 1$ and $f = \phi$ (see the representation (7)). Therefore, by the well-known results (see, e.g., [13]), the arithmetic cost of performing Stage 1, i.e., the arithmetic cost of performing n_B steps of the CG iteration on B, equals

$$\mathrm{cost}(B) = (4l+1)(4l+3)\phi(n)n_B,$$

and similarly at Stage 3, we have

$$\mathrm{cost}(C) = (4l+3)\phi(n)n_C,$$

for n_C iterations of CG, where $\phi(n)$ is the cost of an n-point FFT.

4.1. The Optimal Shift

We will next follow [1] by choosing the optimal μ such that the total work $[(4l+1)(4l+3)n_B + (4l+3)n_C]\phi(n)$ is minimized, where n_B and n_C are the numbers of steps of the CG iteration at Stages 1 and 3, respectively. Let

$$n_B = F\sqrt{k(B)}, \tag{12}$$

$$n_C = F\sqrt{k(C)}, \tag{13}$$

where F is a constant. Then by (4),

$$n_B n_C = F^2 \sqrt{k(A)} = M = \text{constant}.$$

Define

$$f(n_B) = Ln_B + n_C = Ln_B + \frac{M}{n_B},$$

where $L = 4l + 1$. Then $f(n_B)$ is minimized at

$$n_B = \sqrt{\frac{M}{L}}, \qquad n_C = Ln_B. \tag{14}$$

In view of (12)–(14), we choose μ satisfying

$$k(C) = L^2 k(B). \tag{15}$$

Use (2), (3) and let $\mu = m\sqrt{\alpha_1 \alpha_n}$. We have the following equation:

$$m^2 \left(L^2 - k(A)\right) + m\left[2\left(L^2 - 1\right)\sqrt{k(A)}\right] + \left(L^2 k(A) - 1\right) = 0,$$

so

$$m_{\pm} = \frac{-\left(L^2 - 1\right)\sqrt{k(A)} \pm L(k(A) - 1)}{L^2 - k(A)},$$

where $k(A) = \alpha_1/\alpha_n$, $L = 4l + 1$. Since $L \geq 5$, $k(A) \geq 1$, we have $m_- > 0$ only for $k(A) > L^2$.

Lemma 4.1. *[1] Let $\mu = m\sqrt{\alpha_1 \alpha_n}$, where $m = m_-$ (see above). Then*

$$k(B) = L^{-1}\sqrt{k(A)}, \tag{16}$$

$$k(C) = L\sqrt{k(A)}. \tag{17}$$

Now assume (14) and choose $\mu = m_- \sqrt{\alpha_1 \alpha_n}$. Then the total cost is

$$\begin{aligned} (4l+3)[(4l+1)n_B + n_C]\phi(n) &= (4l+3)\left(Ln_B + n_C\right)\phi(n) \\ &= 2(4l+3)F\sqrt{k(C)}\phi(n) \\ &= 2(4l+3)\sqrt{4l+1}\,k^{1/4}(A)F\phi(n). \end{aligned} \tag{18}$$

For comparison, let n_{CG} be the number of iterations required by CG for A. We have

$$\text{Cost}(CG) = (4l+3)n_{CG}\phi(n) = (4l+3)k^{1/2}(A)F\phi(n). \tag{19}$$

Comparing with (18), we can see an improvement for $k(A) > 16(4l+1)^2$.

4.2. Recursive Preconditioning

We may use the factorization $A = T^H T = BC$ recursively. In particular, we may solve equations (8), (9) and (11) at Stage 1 of Algorithm 1 by choosing one optimal shift μ_1, and we may choose another optimal shift μ_2 to solve the system $Cx = z$ for x at Stage 3 of Algorithm 1. Since we have $l_B \leq 2l$, $l_C \leq 2l$ (where l_W denotes the length of an F-generator of W, for $W = B$, $W = C$), it follows from (18), that the total computational cost of performing Stages 1 and 3 is bounded by

$$2(8l+1)(8l+3)\sqrt{8l+1}\,k^{1/4}(B)F\phi(n) \tag{20}$$

and

$$2(8l+3)\sqrt{8l+1}\,k^{1/4}(C)F\phi(n), \tag{21}$$

respectively. Now we choose μ so as to minimize the sum of (20) and (21). Since $k(A) = k(B)k(C)$, we have the solutions $k(B) = \frac{k^{1/2}(A)}{(8l+1)^2}$, $k(C) = (8l+1)^2 k^{1/2}(A)$, and

$$\mu = \frac{\alpha_n k^{1/2}(A)[k^{1/2}(A)(8l+1)^2 - 1]}{k^{1/2}(A) - (8l+1)^2}.$$

We have $\mu > 0$ for $k(A) > (8l+1)^4$, and the total computational cost of recursive preconditioning is

$$4(8l+1)(8l+3)F\phi(n)k^{1/8}(A). \tag{22}$$

This is less than the cost (18) of nonrecursive preconditioning for

$$k(A) > \frac{2^8(8l+1)^8(8l+3)^8}{(4l+1)^4(4l+3)^8}$$

and is also less than the cost of application of the unpreconditioned (CG) method to $Ax = b$ (see (19)) when $k(A) > \left[\frac{4(8l+1)(8l+3)}{4l+3}\right]^{8/3}$.

For $l = 2, 3$, we compare the estimates (18), (19) and (22) and show the results in the next table.

Cost	$l=2$	$l=3$
CG method	$11k^{1/2}(A)F\phi(n)$	$15k^{1/2}(A)F\phi(n)$
nonrecursive	$66k^{1/4}(A)F\phi(n)$	$30\sqrt{13}k^{1/4}(A)F\phi(n)$
recursive	$1292k^{1/8}(A)F\phi(n)$	$2700k^{1/8}(A)F\phi(n)$

5. PRECONDITIONED CG METHOD FOR A TOEPLITZ MATRIX

In this section, we use the same notation as in the previous section, except that T now denotes a nonsingular Toeplitz matrix (so that $l = 2$). Since $B = T^HT + \mu I$, multiplying the matrix B by a vector costs $8\phi(n) + O(n)$. Thus in Algorithm 1, we have $\text{cost}(B) = 72\phi(n)$ at Stage 1. By [13], $\text{cost}(C) = 11\phi(n)$ at Stage 3, for each iteration. Therefore, the overall work is equal to

$$(72n_B + 11n_C)\,\phi(n) = 11\left(\tilde{L}n_B + n_C\right)\phi(n), \qquad \tilde{L} = \frac{72}{11},$$

where n_B and n_C denote the number of the CG iterations at Stages 1 and 3, respectively. Assume the optimal value of $\mu = m_-\sqrt{\alpha_1\alpha_n}$, where

$$m_\pm = \frac{-\left(\tilde{L}^2-1\right)\sqrt{k(A)} \pm \tilde{L}(k(A)-1)}{\tilde{L}^2 - k(A)}.$$

Then, similarly to (17), we derive the following cost bound for the entire computation:

$$22n_C\phi(n) = 12\sqrt{22}\,k^{1/4}(A)F\phi(n). \tag{23}$$

We may compare the bound of (23) to the cost of the solution via the CG method (without preconditioning), which is estimated similarly to (19) and is bounded by

$$8k^{1/2}(A)F\phi(n). \tag{24}$$

The comparison shows that our preconditioning improves the CG method for

$$k(A) > 2450.25.$$

Now, we use the factorization $A = BC$ recursively. We choose μ_1 so as to minimize the cost of performing Stage 1 of Algorithm 1, which gives us the bound

$$9 \cdot 12 \cdot \sqrt{22}\,k^{1/4}(B)F\phi(n) = 108\sqrt{22}\,k^{1/4}(B)F\phi(n), \tag{25}$$

where the factor 9 comes from the equations at Stage 1. At Stage 3, choose μ_2 so as to decrease the cost to

$$4(8\cdot 4+1)(8\cdot 4+3)F\phi(n)k^{1/8}(C) = 4620F\phi(n)k^{1/8}(C) \tag{26}$$

(compare (22)). Now we choose μ so as to minimize the sum of (25) and (26). Then we obtain that

$$k(B) = \left(\frac{1155}{54}\right)^8 \cdot \frac{1}{22^{4/3}} \cdot k^{1/3}(A),$$
$$k(C) = \left(\frac{54}{1155}\right)^{8/3} \cdot (22)^{4/3} \cdot k^{2/3}(A),$$

and the overall cost is bounded by

$$\left[108(22)^{1/6}\left(\frac{1155}{54}\right)^2 + 4620\left(\frac{54}{1155}\right)^{1/3} 22^{1/6}\right] k^{1/12}(A)F\phi(n) = Ek^{1/12}(A)F\phi(n), \tag{27}$$

where

$$E = \left[108\left(\frac{1155}{54}\right)^2 + 4620\left(\frac{54}{1155}\right)^{1/3}\right] 22^{1/6} = 400,993.268\ldots$$

(compare(22)). Therefore, the recursive method is superior to the nonrecursive method only if $k(A)$ is enourmosly large: $k(A) > \left(E/(12\sqrt{22})\right)^6$. We also compare (27) and (24) and conclude that the recursive method improves the unpreconditioned CG method only for extremely large $k(A)$, $k(A) > (E/8)^{12/5}$.

REFERENCES

1. V.Y. Pan and R. Schreiber, A fast, preconditioned conjugate gradient solver, *Computers Math. Applic.* **24** (7), 17–24 (1992).
2. W.F. Trench, A note on a Toeplitz inversion formula, *Linear Algebra Appl.* **29**, 55–61 (1990).
3. G. Cybenko and M. Berry, Hyperbolic householder algorithm for factoring structured matrices, *SIAM J. Matrix Anal.* **11** (4), 499–520 (1990).
4. J. Chan and T. Kailath, Divide and conquer solution of least-squares problems for matrices with displacement structure, *SIAM J. Matrix Anal. Appl.* **12** (1), 128–145 (1991).
5. G. Ammar and W.G. Gragg, Superfast solution of real positive definite Toeplitz systems, *SIAM J. Matrix Anal. Appl.* **9** (1), 61–76 (1988).
6. R.R. Bitmead and B.D.O. Anderson, Asymptotically fast solution of Toeplitz and related system of linear equations, *Linear Algebra Appl.* **41**, 111–130 (1981).
7. R.P. Brent, F.G. Gustavson and D.Y.Y. Yun, Fast solution of Toeplitz systems of equations and computation of Pade approximations, *J. of Algorithms* **1**, 259–295 (1980).
8. F.R. deHoog, On the solution of Toeplitz systems, *Linear Algebra Appl.* **88/89**, 123–138 (1987).
9. R.H. Chan, Circulant preconditioners for Hermitian Toeplitz system, *SIAM J. Matrix Anal. Appl.* **10** (4), 542–550.
10. R.H. Chan, Toeplitz preconditioners for Toeplitz systems with nonnegative generating functions, *IMA J. of Numerical Analysis* **11**, 333–345 (1991).
11. R.H. Chan and G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, *SIAM J. Sci. Stat. Comput.* **10** (1), 104–119 (1989).
12. G. Strang, A proposal for Toeplitz matrix calculations, *Studies in Appl. Math.* **74**, 171–176 (1986).
13. I. Gohberg and V. Olshevsky, Complexity of multiplication with vectors for structured matrices, *Linear Algebra Appl.* **202**, 163–192 (1994).
14. D. Bini and V.Y. Pan, *Polynomial and Matrix Computations*, Volume 1, Fundamental Algorithms, Birkhauser, Boston, (1994).
15. T. Kailath, S.Y. Kung and M. Morf, Displacment ranks of matrices and linear equations, *J. Math. Anal. Appl.* **68** (2), 395–407 (1979).
16. G.H. Golub and C.F. Van Loan, *Matrix Computation*, Johns Hopkins Univ. Press, Baltimore, MD, (1989).