A Look-ahead Algorithm for Solving Nonsymmetric Linear Toeplitz Equations

Thomas Huckle†

Abstract

Fast solvers for linear Toeplitz equations like the Levinson algorithm are known to be stable only for symmetric positive definite matrices. In general they may break down caused by a division with zero in view of a singular submatrix. One way to overcome this breakdown uses the connection between Toeplitz matrices and orthogonal polynomials. Here, we will present another approach for dealing this case. In the case of a breakdown we introduce perturbations of the original matrix in order to make the submatrix regular. Then we can step forward without breakdown, and after a few steps we have to return to the original matrix.

1 Look-ahead Algorithm for Computing Gohberg-Semencul Formulas

The solution of linear equations $T_n x = b$ with $T_n = (t_{i-j})_{i,j=0}^{n}$ a Toeplitz matrix appears in many applications (see e.g. [2]). If we want to solve more than one Toeplitz equation then usually we solve one or two special linear systems in order to determine a Gohberg-Semencul formula that expresses T_n^{-1} as the sum of products of upper and lower Toeplitz matrices. Given a Gohberg-Semencul formula we can solve every equation $T_n x = b$ in $O(n\log(n))$ steps. For the positive definite case the original Gohberg-Semencul formula [1,7] that we get by solving $T_n x = e_0$ and $T_n y = e_n$ exists,

$$T_n^{-1} = (1/x_0)\Big(L_n(x)U_n(y) - L_n(ZJy)U_n(ZJx)\Big)$$

but in the general case it breaks down for $x_0 = 0$, x_0 the first component of the vector x. The second Gohberg-Semencul formula is given by [6,9]

$$T_n^{-1} = \Big(L_n(z)U_n(ZJx) - L_n(x)U_n(-e_0 + ZJz)\Big)$$

with $T_n z = (0, t_{-n}, ..., t_{-1})^T$), $e_0 = (1, 0, ..., 0)^T$, Z is the lower shift matrix, J is the counteridentity $J_{j,k} = \delta_{n+1-j,k}$, $j, k = 0, ..., n$, L_n denotes a lower Toeplitz matrix and U_n an upper Toeplitz matrix. This formula is always well defined. All possible Gohberg-Semencul formulas are described in [10] using the displacement representaion of a Toeplitz matrix [11]. If we have computed one Gohberg-Semencul formula then we can easily get the generating vectors of all other formulas [10].

The aim of the presented algorithm is to compute generating vectors for Gohberg-Semencul formulas for a given regular Toeplitz matrix T_n recursively as fast and stable as possible. The method is based on the recurrence between the generating vectors x, y,

---

† Department of Computer Science, SCCM, Stanford University, and Institut f. Angew. Mathematik, Universität Würzburg, Germany

and z that is described in [9,p.56]. Let us first recall these equations for the case that all submatrices of T_n are regular. Set

$$T_k x^k = e_0 \ , \ T_k z^k = \tau_-^k = (t_{-k-1} \ \ \dots \ \ t_{-1})^T \ , \ T_k^T y^k = e_0 \ , \ T_k^T w^k = \tau_+^k = (t_{k+1} \ \ \dots \ \ t_1)^T \ .$$

Then we get the two different recursions

$$x^{k+1} = \frac{1}{1 - w_0^k z_0^k}\left(\begin{pmatrix} x^k \\ 0 \end{pmatrix} - w_0^k \begin{pmatrix} 0 \\ y^k \end{pmatrix}\right) \tag{1}$$

$$y^{k+1} = \frac{1}{1 - w_0^k z_0^k}\left(\begin{pmatrix} 0 \\ y^k \end{pmatrix} - z_0^k \begin{pmatrix} x^k \\ 0 \end{pmatrix}\right) \tag{2}$$

or

$$x^{k+1} = \begin{pmatrix} x^k \\ 0 \end{pmatrix} - \frac{w_0^k}{\tau_+^k z^k - t_0}\begin{pmatrix} z^k \\ -1 \end{pmatrix} \tag{3}$$

$$z^{k+1} = \begin{pmatrix} 0 \\ z^k \end{pmatrix} - (\tau_-^k z^k - t_{-k-2})x^{k+1} \tag{4}$$

with

$$w_0^k = \tau_+^k x^k \ , \ z_0^k = \tau_-^k x^k \ , \ w_z^k = \tau_+^k z^k = \tau_-^k w^k = z_w^k \ , \ \gamma^k = w_z^k - t_0 \ , \ z_z^k = \tau_-^k z^k \ . \tag{5}$$

The (x, y)-based formulas (1-2) have the advantage that they need only two inner products for w_0^k and z_0^k but the disadvantage that a breakdown occurs if T_{k-1} or T_{k+1} is singular. The recurrence (3-4) takes three inner products for w_0^k, w_z^k, and z_z^k but breaks down only if T_{k+1} is singular. Thus we are interested in mainly using (1-2), but we have to switch between both recurrences sometimes in order to avoid unnecessary breakdowns. To describe these switches we can use the formulas (see [10] and [9])

$$z^k = \frac{1}{x_0^k}(z_0^k x^k - ZJy^k) \ \text{and} \ y^{k+1} = \frac{1}{w_z^k - t_0} J \begin{pmatrix} z^k \\ -1 \end{pmatrix} . \tag{6}$$

Furthermore, we see from [10]

$$-e_0 + ZJz^k = (w_z^k - t_0)y^k - z_0^k w^k \ ,$$

that

$$-(w_z^k - t_0)x_0^k = 1 - z_0^k w_0^k \tag{7}$$

Therefore, (1-2) breaks down if $x_0^k = 0$ or if (3-4) breaks down. Especially for $w_z^k - t_0 = 0$ there holds $1 = z_0^k w_0^k$ and hence $z_0^k \neq 0$. Thus, if in (3-4) a breakdown occurs, that means $w_z^k - t_0 = 0$, then z_0^k is not equal 0 and we can force $\tilde{w}_z^k - t_0 \neq 0$ by changing t_{k+1} to $\tilde{t}_{k+1}$. In the same way, if (1-2) breaks down with $x_0^k \neq 0$ then also $z_0 \neq 0$ and in view of (7) we can force $1 - w_0^k z_0^k$ to be not equal to zero. Hence, we get regular extensions $\tilde{T}_{k+1}, \dots, \tilde{T}_{k+r}$ of T_k.

In every step we now have to check whether the original matrix T_{k+r} is regular, and if this is true we have to switch from the generated vectors $\tilde{x}^{k+r}$ and $\tilde{y}^{k+r}$ to the original vectors x^{k+r} and z^{k+r}. Note, that we can not compute y^{k+r} because T_{k+r-1} will be illconditioned. From (x^{k+r}, z^{k+r}) we can compute (x^{k+r+1}, y^{k+r+1}), or in the breakdown case $(\tilde{x}^{k+r+1}, \tilde{y}^{k+r+1})$. Thus, the main recurrence is of the form (1-2), and only after a breakdown we use (3) and the second equation in (6).

To get (x^{k+r}, z^{k+r}) we apply the Sherman-Morrison-Woodbury formula

$$(8) \qquad T_{k+r}^{-1} = (\tilde{T}_{k+r} + \tilde{\Delta})^{-1} = \tilde{T}_{k+r}^{-1} - \tilde{T}_{k+r}^{-1} \begin{pmatrix} 0 \\ I_r \end{pmatrix} G_{k+r}^{-1} \begin{pmatrix} \Delta_r & 0 \end{pmatrix} \tilde{T}_{k+r}^{-1}$$

with

$$(9) \qquad G_{k+r} = I_r + \Delta_r \begin{pmatrix} I_r & 0 \end{pmatrix} \tilde{T}_{k+r}^{-1} \begin{pmatrix} 0 \\ I_r \end{pmatrix} .$$

and

$$\Delta_r = \begin{pmatrix} t_{k+1} - \tilde{t}_{k+1} & 0 & \cdots & 0 \\ t_{k+2} - \tilde{t}_{k+2} & t_{k+1} - \tilde{t}_{k+1} & & \vdots \\ \vdots & & & \vdots \\ t_{k+r} - \tilde{t}_{k+r} & \cdots & \cdots & t_{k+1} - \tilde{t}_{k+1} \end{pmatrix} .$$

We can compute G_{k+r} by using the Gohberg-Semencul formula for $\tilde{T}_{k+r}^{-1}$ in terms of $\tilde{x}^{k+r} = x$ and $\tilde{y}^{k+r} = y$ in the form

$$(10) \quad (I_r\ 0)\tilde{T}_{k+r}^{-1} \begin{pmatrix} 0 \\ I_r \end{pmatrix} = L_1(1:r, 1:r)U_1(1:r, k+1:k+r) - L_2(1:r, 1:r)U_2(1:r, k+1:k+r)$$

in $O(r^3)$ operations. Then we check the singular values of G_{k+r}. If G_{k+r} is near-singular we choose a new $\tilde{t}_{k+r+1}$ and compute $(\tilde{x}^{k+r+1}, \tilde{y}^{k+r+1})$. We always try to choose the new element equals the original t_{k+r+1}, and if this is not possible we force γ^{k+r+1} or $1 - w_0^{k+r+1} z_0^{k+r+1}$ to be 1.

If the singular values of G_{k+r} are not too small we have to jump back to the regular original matrix T_{k+r}. This matrix has near singular submatrices, and therefore we use x and z instead of x and y in order to characterize T_{k+r}^{-1}. First, we compute $\tilde{z}^{k+r}$ using (6) which needs in addition one inner product. Then we have to compute with $m = k + r$

$$T_m^{-1}(e_0, \tau_-^m) = (\tilde{x}^m, \tilde{z}^m) - \tilde{T}_m^{-1} \begin{pmatrix} 0 \\ I_r \end{pmatrix} G_m^{-1}(\Delta_r\ 0)(\tilde{x}^m, \tilde{z}^m) = (\tilde{x}^m, \tilde{z}^m) - \tilde{T}_m^{-1}(\begin{pmatrix} 0 \\ u \end{pmatrix}, \begin{pmatrix} 0 \\ v \end{pmatrix}).$$

Now, any solution of $\tilde{T}_{k+r} a = b$ can be computed recursively by (see [9,p.60])

$$a^j = \begin{pmatrix} a^{j-1} \\ 0 \end{pmatrix} - J\tilde{y}^j(\tilde{\tau}_+^m a^{j-1} - b_j) .$$

In our case, the vector b has $k+1$ leading zeros and therefore the first iterates are zero. This leads to the recursions

$$(11) \qquad \phi^k = 0_{k+1} \ , \ \phi^{k+j} = \begin{pmatrix} \phi^{k+j-1} \\ 0 \end{pmatrix} - J\tilde{y}^{k+j}(\tilde{\tau}_+^{k+j} \phi^{k+j-1} - u(j))$$

$$(12) \qquad \psi^k = 0_{k+1} \ , \ \psi^{k+j} = \begin{pmatrix} \psi^{k+j-1} \\ 0 \end{pmatrix} - J\tilde{y}^{k+j}(\tilde{\tau}_+^{k+j} \psi^{k+j-1} - v(j))$$

for $j = 1, ..., r$. Then we get the new generating vectors for T_{k+r} by

$$(13) \qquad (x^{k+r}, z^{k+r}) = (\tilde{x}^{k+r}, \tilde{z}^{k+r}) - (\phi^{k+r}, \psi^{k+r}) .$$

Now, we have to compute a new pair (x, y) on level $(k+r+1)$ from (x, z). For x^{k+r+1} we use (3), which also shows whether T_{k+r+1} is singular or not. The vector y^{k+r+1} results from (6) without using additional inner products. Moreover, from (7) we get w_0^{k+r} without using further inner products. Hence, the algorithm takes in every step two inner products plus two SAXPY's of size k for computing the generating vectors, and in the case of breakdown we need in addition 2 inner products plus 2 SAXPY's of size $k+1, k+2, ..., k+r$ for updating ϕ and ψ. If T_{k+r+1} again is near singular then we again have a breakdown, and we compute $\tilde{x}^{k+r+1}$ and $\tilde{y}^{k+r+1}$ using (3), (6), and (7), after perturbing T_{k+r+1}, in order to get a regular matrix $\tilde{T}_{k+r+1}$.

If $t_0 = 0$, then the algorithm starts with a breakdown. Therefore, in this case we set $x = z = []$, $w_z - t_0 = 0$, and begin with the breakdown case.

BASIC ALGORITHM:

INPUT: $t_{-n}, ..., t_0, ..., t_n$, $ep1$, $ep2$
OUTPUT: (x, y) or (x, z), generating vectors for Gohberg-Semencul formulas
Initialisation:
  If $|t_0| > ep1$: Set $x = y = 1/t_0$, $w_0 = t_1 x$, $z_0 = t_{-1} y$, $m = 1 - w_0 z_0$, $k = 0$
  If $|t_0| \le ep1$: Set $x = y = []$, $m = 0$, $k = -1$
Repeat until ready:
  a) If $|m| > ep1$: (Regular step)
    compute (x, y) according to (1-2) or (x, y) according to (3)(6);
    compute w_0,z_0, $m = 1 - w_0 z_0$; set $k = k+1$; if $k = n-1$: ready;
  b) If $|m| \le ep1$: (Irregular step)
    $\alpha 1$) if $k = -1$: Set $\tilde{t}_0 = \tilde{y} = \tilde{x} = -1$,
    $\alpha 2$) If $k \ge 0$: Set $\tilde{t} = (t_0, ..., t_k)$;
      compute $(\tilde{x}, \tilde{y}_1)$ (1-2) with $\tilde{t}_{k+1} = \frac{m-1}{x_0 z_0} + t_{k+1}$, $w_0 = w_0 + x_0(\tilde{t}_{k+1} - t_{k+1})$;
      or
      compute $(\tilde{x}, \tilde{y}_1)$ (3),(6) with $\tilde{t}_{k+1} = \frac{1-m}{z_0} + t_{k+1}$, $w_0 = w_0 + x_0(\tilde{t}_{k+1} - t_{k+1})$;
    β) Repeat until regular:
      $r = 2$, compute w_0, z_0, $m = 1 - w_0 z_0$;
      1) If $|m| > ep1$: $\tilde{t}_{k+r} = t_{k+r}$, compute $(\tilde{x}, \tilde{y}_r)$ with (1-2);
      2) If $|m| \le ep1$: compute $(\tilde{x}, \tilde{y}_r)$ using (1-2) with $\tilde{t}_{k+r} = \frac{m-1}{x_0 z_0} + t_{k+r}$,
        $m = 1$, $w_0 = w_0 + \tilde{x}_0(\tilde{t}_{k+r} - t_{k+r})$;
    γ) Compute G from $(\tilde{x}, \tilde{y}_r)$ with (10);
    δ) 1) If $min(svd(G)) \le ep2$: r=r+1, singular;
      2) If $min(svd(G)) > ep2$: compute $\tilde{z}$ with (6);
        $(u, v) = G^{-1}\Delta_r((\tilde{x}, \tilde{z})(1:r))$;
        $phi = psi = 0_{k+1}$; compute phi, psi with (11-12) using $\tilde{y}_1, ..., \tilde{y}_r$,
        and (x, z) with (13)
        Set $k = k + r$, if $k = n-1$: ready;
        $m = (t_{k+2}, ..., t_0)\begin{pmatrix} z \\ -1 \end{pmatrix}$, $w_0 = \frac{1+mx_0}{z_0}$; regular.

Note, that m is defined in two different ways, $m = 1 - w_0 z_0$ or $m = w_z - t_0$, dependent on what formula is used for computing (x, y).

The numerical examples are taken from [4]. We consider Toeplitz matrices of the form $T = T^0 + \epsilon T^1$ where the entries of T^1 are randomly distributed in $[0, 1]$ and T^0 is given as the coefficients of the Taylor expansion of

$$f(z) = 18 + \frac{1}{1-z} - \frac{3}{1-z^3} + \frac{6}{1-z^6} - \frac{24}{1-z^{24}} + \frac{48}{1-z^{48}} - \frac{96}{1-z^{96}} .$$

The submatrices of dimension $51, ..., 57$ are exactly singular, while all other submatrices are well-conditioned. The second example is given by $T_n(\epsilon)$ with $t_0 = \epsilon$, $t_j = 1/2^j$, $j = 1, ..., n$. Here for $\epsilon = 0$ every third submatrix is exactly singular.

We apply our look-ahead algorithm for different values of ϵ and display the relative error $r = ||x - T^{-1}e_0||/||x||$. $ep1$ and $ep2$ are chosen to be 0.01.

ϵ	1.0e-02	1.0e-04	1.0e-06	1.0e-08	1.0e-10	1.0e-12	1.0e-14	0.0e+00
r	1.2e-13	6.6e-13	4.4e-13	7.5e-13	3.9e-13	5.5e-13	3.6e-13	7.2e-13

Table 1. Relative error for example 1 with different values of ϵ

For $\epsilon = 0.01$ there occured no breakdown in view of $ep1 = 0.01$, but for all other values of ϵ the algorithm made a jump from 51 to 58.

n+1	15	30	60	120	240	480
r	3.11e-15	3.94e-15	1.08e-14	9.87e-14	4.19e-13	1.18e-12

Table 2. Relative error for example 2 with $\epsilon = 10e - 14$

Here the algorithm had a jump of height 2 every third step. If we compare this results with the relative errors for the look-ahead Bareiss algorithm given in [4], we see that our relative errors are only slightly larger, but our computational efford is much smaller.

The algorithm here is given in its fastest version. We can consider different variations of the basic algorithm, e.g. for the symmetric case, which can be found in [10]. For other look-ahead Toeplitz solvers see [3,4,5,8].

References

[1] G. Baxter, I. I. Hirschman, *An explicit inversion formula for finite-section Wiener-Hopf operators*, Bull. AMS, 70 (1974), pp. 820–823.

[2] J. R. Bunch, *Stability of methods for solving Toeplitz systems of equations*, SIAM J. Sci. Stat. Comput., 6 (1985), pp. 349–364.

[3] T. F. Chan, P. C. Hansen, *A look-ahead Levinson algorithm for general Toeplitz Systems*, IEEE Trans. Signal Process., 40 (1992), pp. 1079-1090.

[4] R. W. Freund, *A look-ahead Bareiss algorithm for general Toeplitz matrices*, AT& T Bell Laboratories, (1993).

[5] R. W. Freund, H. Zha, *Formally biorthogonal polynomials and a look-ahead Levinson algorithm for general Toeplitz systems*, Linear Alg. Appl., 188/189 (1993), pp. 255-303.

[6] I. Gohberg, N. Krupnik, *A formula for the inversion of finite-section Toeplitz matrices (Russian)*, Matem. Issled., Kishinev,7,2 (1972), pp. 272-284.

[7] I. Gohberg, A. A. Semencul, *On inversion of finite-section Toeplitz matrices and their continuous analogous (Russian)*, Matem. Issled., Kishinev, 7,2 (1972), pp. 201-224.

[8] M. H. Gutknecht, M. Hochbruck, *Look-ahead Levinson and Schur Algorithms for Non-Hermitian Toeplitz Systems*, IPS Research Report No. 93-11, submitted to Numerische Mathematik.

[9] G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators*, Birkhäuser, Basel, 1984.

[10] T. Huckle, *Computation of Gohberg-Semencul Formulas for a Toeplitz Matrix*, SCCM-Report 93-14, Stanford University, submitted to Linear Alg. Appl..

[11] T. Kailath, S. Kung, M. Morf, *Displacement ranks of Matrices and Linear Equations*, J. Math. An. Appl., 68 (1979), pp. 395-407.