Fraction-Free Inversion of a Toeplitz Matrix

Yuval Bistritz, *Fellow IEEE*, and Yaron Segalov
School of Electrical Engineering, Tel Aviv University
Tel Aviv 69978, Israel

***Abstract*—The paper considers Levinson algorithms for Hermitian and non-Hermitian Toeplitz matrices that for integer matrices remain fraction-free (FF). A recently introduced FF algorithm is extended from Hermitian to non-symmetric Toeplitz matrices. An alternative proof for the integer-preservation property is obtained by linking the elements of the solution vectors to minors of the Toeplitz matrix. These links are also used to prove that the length of integers grows at a very restrained rate, a property that implies that the algorithms are very efficient integer algorithms.**

I. Introduction

The Levinson Algorithm is a fast algorithm that solves in $O(n^2)$ of arithmetic operations a set of equations with a Toeplitz matrix,

$$\mathbf{T}_n = \begin{bmatrix} r_0 & r_1 & \cdots & r_n \\ r_{-1} & r_0 & \cdots & r_{n-1} \\ \vdots & & \ddots & \\ r_{-n} & r_{-(n-1)} & \cdots & r_0 \end{bmatrix} \tag{1}$$

where $r_i \in \mathbb{C}$ (the field of complex numbers). It is widely used in a variety of applications including linear prediction, modeling the vocal tract in speech processing, modeling wave propagation in layered media and more. Often there $\mathbf{T}_n$ is positive definite real symmetric $r_{-i} = r_i$ or Hermitian $r_{-i} = r_i^\star$ matrix because its entries are the (auto-) correlations of the processed signal. The algorithm for the Hermitian case (brought below as Algorithm 1) solves successively the following subsets of normal equations (^{T} denotes transpose)

$$\mathbf{T}_m[a_{m,0}, \cdots, a_{m,m-1}, 1]^T = [0, \cdots, 0, D_m]^T , \; m = 0, \ldots, n \tag{2}$$

In the non-Hermitian case, the algorithm (brought below as Algorithm 3) involves side by side successive solution for the next two sets of normal equations,

$$\mathbf{T}_m[a_{m,0}, \cdots, a_{m,m-1}, 1]^T = [0, \cdots, 0, D_m]^T, \; m = 0, ..., n \tag{3a}$$

$$[b_{m,0}, \cdots, b_{m,m-1}, 1]\mathbf{T}_m = [0, \cdots, 0, D_m] , \; m = 0, \ldots, n \tag{3b}$$

In order that all these sets of equations have solution, the principal minors of the matrix,

$$\Delta_m := \det(\mathbf{T}_m) , \; m = 0, 1, \ldots, n \tag{4}$$

must all be nonzero. This condition, usually called *strong regularity*, will be assumed in this paper. Namely, the paper will consider Levinson algorithms for only strongly regular Hermitian or non-Hermitian matrices.

The algorithm for a symmetric case is due to Levinson [1]. The algorithm for a nonsymmetric Toeplitz set of equations was devised by Trench [2] and afterwards was considered by several other researchers, see [3] and references there in. Our templates for citing these algorithms follows the study in [4] that treated these algorithm in a somewhat more general framework called quasi-Toeplitz matrices.

In [5] we proposed a new form for the Levinson algorithm that acquires it with an *integer preserving* (IP) property. The new property means that when $\mathbf{T}_n$ is a matrix of integers the algorithm (and an implied triangular factorization of the inverse matrix) is carried out over integers. The algorithm is not devoid of divisions. Actually the reason it excels as an integer algorithm is because it uses divisions to get rid from superfluous integer common factors that otherwise would blow up the size of the integers as the algorithm proceeds. The term *fraction-free* (FF) is meant to emphasize the fact that the algorithm remains IP even though it involves divisions.

This paper studies further the properties of the new algorithm and brings its extension to non-Hermitian Toeplitz matrices. It will show that the leading coefficients of the polynomials that the new algorithm produces are the principal minors of the Toeplitz matrix and that the remaining coefficients are too linked to determinants of a certain submatrices of the matrix. This implies an alternative proof for the integer preserving (IP) property of the new algorithm. It will also be used to prove the restrained growth of the length of the integers in the algorithm.

II. The Hermitian Toeplitz case

A. *The ordinary algorithm*

The familiar Levinson algorithm for solving the equations (2) is brought below using polynomial notation with $a_m(z) = \sum_{i=0}^m a_{m,i}z^i, \quad m = 0, \ldots, n$ and $a_m^\#(z) := \sum_{i=0}^m a_{m,m-i}^\star z^i$.

Algorithm 1 [The ordinary Levinson algorithm for a Hermitian Toeplitz matrix]. Set $a_0(z) = 1$ and $D_0 = r_0$. [1]

For $m = 1, \ldots, n$ do:

$$k_m = \frac{[a_{m-1,0}, \cdots, a_{m-1,m-1}][r_1, \cdots, r_m]^T}{D_{m-1}} \tag{5a}$$

$$a_m(z) = za_{m-1}(z) - k_m a_{m-1}^\sharp(z) \tag{5b}$$

$$D_m = D_{m-1}\left(1 - |k_m|^2\right) \tag{5c}$$

The coefficient vector of $a_m(z)$ and D_m solve (2).

B. *The IP algorithm*

We recently presented in [5] the next Levinson algorithm and showed that it is IP. The algorithm is again brought in polynomial notation with $f_m(z) = \sum_{i=0}^m = f_{m,i}z^i$, $m = 0, \ldots, n$.

Algorithm 2 [Fraction-Free Levinson algorithm for a Hermitian Toeplitz matrix]. Consider $\mathbf{T}_n$ in (1) with $r_i \in \mathbb{C}$ and $r_{-i} = r_i^\star$. Set $\epsilon_{-1} = 1, f_0(z) = 1$ and $\epsilon_0 = r_0$.

For $m = 1, \ldots, n$ do:

$$\delta_m = [f_{m-1,0}, \ldots, f_{m-1,m-1}][r_1, \ldots, r_m]^T \tag{6a}$$

$$f_m(z) = \frac{\epsilon_{m-1}zf_{m-1}(z) - \delta_m f_{m-1}^\sharp(z)}{\epsilon_{m-2}} \tag{6b}$$

$$\epsilon_m = \frac{\epsilon_{m-1}^2 - |\delta_m|^2}{\epsilon_{m-2}} \tag{6c}$$

[1] One usually finds $D_0 = 1$ in the literature. This happens because it is customary to assume that the matrix is normalized to $r_0 = 1$. It can be shown that taking $D_0 = r_0$ is the only correction needed to set the customary setting (e.g. [4, eq. 3.46]) free from this normalization that is not appropriate for integer Toeplitz matrices.

Proc. of the 2010 IEEE Int. Symposium on Circuits and Systems, ISCAS-2010, May 2010, Paris, France.

By comparing the leading coefficient in the above algorithm, step after step, it follows that

$$f_{m+1,m+1} = \epsilon_m \ , \ m = 0, \dots, n-1 \tag{7}$$

It was shown in [5] that the coefficients of the polynomial $f_m(z)$ and ϵ_m produced by Algorithm 2 solve the set of equations

$$\mathbf{T}_m [f_{m,0}, ..., f_{m,m-1}, f_{m,m}]^T = [0, \dots, 0, \epsilon_m]^T, m = 1, \dots, n \tag{8}$$

Reading the last row of (8) reveals that ϵ_m can be computed from the coefficient vector of vector $f_m(z)$ by the inner product

$$\epsilon_m = \sum_{i=0}^{m} f_{m,i} r^*_{m-i} \tag{9}$$

The simpler update formula (6c) for the computation of ϵ_m can be obtained from (9) by induction. Check (6c) for ϵ_1. Assume that (6c) holds till ϵ_m, then ϵ_{m+1}

$$\begin{aligned} &= \sum_{i=0}^{m+1} f_{m+1,i} r^*_{m+1-i} = \sum_i \frac{\epsilon_m f_{m,i-1} - \delta_{m+1} f^*_{m,m-i}}{f_{m,m}} r^*_{m+1-i} \\ &= \frac{\epsilon_m \sum_j f_{m,j} r^*_{m-j} - \delta_{m+1} \sum_k f^*_{m,k} r^*_{k+1}}{f_{m,m}} \\ &= \frac{\epsilon_m^2 - \delta_{m+1}\delta^*_{m+1}}{f_{m,m}} = \frac{\epsilon_m^2 - |\delta_{m+1}|^2}{\epsilon_{m-1}} \end{aligned}$$

It is also apparent from (6c) that the ϵ_m's are real (also for a complex matrix).

Let $\mathbb{Z}$ and $\mathbb{Z}[\mathbf{i}]$ denote the field of real and complex integers, respectively. (The elements of $\mathbb{Z}[\mathbf{i}]$, also known as *Gaussian* integers, are of the form $\alpha + j\beta$ with $\alpha, \beta \in \mathbb{Z}$, where j denotes the imaginary unit).

Theorem 1. *Algorithm 2 is integer preserving: If r_m, $m = 0, \dots, n$ are in $\mathbb{Z}[\boldsymbol{i}]$ (resp. in $\mathbb{Z}$) then $\{f_{m,i} \ ,: i = 0, \dots, m\}$, δ_m are $\mathbb{Z}[\boldsymbol{i}]$ (resp. in $\mathbb{Z}$) and ϵ_m in $\mathbb{Z}$, for all $m = 0, \dots, n$*

The derivation of Algorithm 2 in [5] began with an algorithm that is IP simply because it avoids divisions. Next, it was shown that the divisions that appear in Algorithm 2 leave the algorithm fraction free because they remove common integer factors in the numerator. The manner of derivation there also provided a constructive proof for Theorem 1.

Theorem 2. *An upper-bound for the length of the coefficients of $f_m(z)$ created by Algorithm 2 is given by*

$$mB + \frac{1}{2} m \log(m), \ , \ m = 1, \dots, n \tag{10}$$

where B represents a bound on the length of the largest entry of the integer matrix [2].

We now proceed to establish some interesting relations between the coefficients of the polynomials produced by Algorithm 2 and certain sub-determinants of $\mathbf{T}_n$. They will imply an alternative proof for Theorem 1 and will also be used to prove Theorem 2 (that was stated in [5] without proof).

Let $\mathrm{M}^{(m)}_{i,j}$ denote the (i, j)-th minor of the matrix $\mathbf{T}_m$. That is, $\mathrm{M}^{(m)}_{i,j}$ is the determinant of the m-th size submatrix obtained from the $(m+1)$-th matrix $\mathbf{R}_m$ by removing from it row i and column j. Let $C^{(m)}_{i,j}$ denote the corresponding(i,j) cofactor. Namely,

$$C^{(m)}_{i,j} \equiv (-1)^{i+j} \mathrm{M}^{(m)}_{i,j}$$

Theorem 3. *The ϵ_m's produced by Algorithm 2 are equal to the principal minors* (4) *of the Hermitian Toeplitz matrix $\boldsymbol{T}_n$,*

$$\epsilon_m = \Delta_m \ , \ m = 0, 1, \dots, n \tag{11}$$

Proof: The coefficient vector of $f_m(z)$ solves the set of equations (8), therefore

$$\begin{bmatrix} f_{m,0} \\ f_{m,1} \\ \vdots \\ f_{m,m} \end{bmatrix} = \mathbf{T}_m^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \epsilon_m \end{bmatrix}$$

Express $\mathbf{T}_m^{-1}$ as $\frac{\mathrm{adj}(\mathbf{T}_\mathrm{m})}{\Delta_m}$ where

$$\mathrm{adj}(\mathbf{T}_\mathrm{m}) = \begin{bmatrix} C^{(m)}_{0,0} & C^{(m)}_{1,0} & \dots & C^{(m)}_{m-1,0} & C^{(m)}_{m,0} \\ C^{(m)}_{0,1} & C^{(m)}_{1,1} & \dots & C^{(m)}_{m-1,1} & C^{(m)}_{m,1} \\ \vdots & \vdots & & \vdots & \vdots \\ C^{(m)}_{0,m-1} & C^{(m)}_{1,m-1} & \dots & C^{(m)}_{m-1,m-1} & C^{(m)}_{m,m-1} \\ C^{(m)}_{0,m} & C^{(m)}_{1,m} & \dots & C^{(m)}_{m-1,m} & C^{(m)}_{m,m} \end{bmatrix} \tag{12}$$

It follows that

$$\begin{bmatrix} f_{m,0} \\ f_{m,1} \\ \vdots \\ f_{m,m} \end{bmatrix} = \frac{\epsilon_m}{\Delta_m} \begin{bmatrix} C^{(m)}_{m,0} \\ C^{(m)}_{m,1} \\ \vdots \\ C^{(m)}_{m,m} \end{bmatrix} \tag{13}$$

From the last row of this equality,

$$f_{m,m} = \frac{\epsilon_m C^{(m)}_{m,m}}{\Delta_m}$$

But $C^{(m)}_{m,m} = \mathrm{M}^{(m)}_{m,m} = \Delta_{m-1}$ and since by (7) $f_{m,m} = \epsilon_{m-1}$ we can write

$$\epsilon_{m-1} = \frac{\epsilon_m \Delta_{m-1}}{\Delta_m}$$

The last equation implies (11) by a trivial induction: $\epsilon_0 = r_0 = \Delta_0$ is true. It then implies $\epsilon_1 = \Delta_1$ that in turn implies $\epsilon_2 = \Delta_2$ and so forth. ■

Theorem 4. *Algorithm 2 produces a sequence of polynomials whose leading coefficients are the principal minors of the Hermitian Toeplitz matrix $\boldsymbol{T}_n$ as follows*

$$f_{m,m} = \Delta_{m-1} \ , \ m = 1, \dots, n \tag{14}$$

Proof: An obvious combination of Theorem 3 and (7) ■

The feature stated in Theorem 4 is an interesting characteristic of the new algorithm that, in difference from the IP property, is observable irrespective of whether the matrix is over integers or not.

Theorem 5. *All the polynomials that Algorithm 1 produces can be expressed in terms of the cofactors of the matrix in the set of equations that their coefficient vector solve as follows.*

$$f_m(z) = C^{(m)}_{m,0} + C^{(m)}_{m,1} z + \dots + C^{(m)}_{m,m} z^m \ , \ m = 1, \dots, n \tag{15}$$

Proof: Follows from (13) once (11) has been proved. ■

Note that Theorem 5 includes Theorem 4 as a special case because $f_{m,m} = C^{(m)}_{m,m} = \Delta_{m-1}$ and that (15) is a noticeable property

[2]The length of an integer r may be presented by $\log |r|$ or by some similarly behaving measure. 'Length' may be the number of digits required to present r (corresponding to $\log_{10} |r|$) or the number of bits required to present it (corresponding to $\log_2 |r|$). The latter is the more widely used choice (and it will be assumed, for concreteness, in the proof of this theorem below)

whether or not the Hermitian Toeplitz matrix $\mathbf{T}_n$ is over integers. However, for an integer matrix Theorem 15 provides an alternative proof for Theorem 1.

Proof for Theorem 1: When the entries of $\mathbf{T}_n$ are (Gaussian) integers then all its sub-determinants are (Gaussian) integers. Therefore all the cofactors of $\mathbf{T}_m$ are (Gaussian) integers. Thus the coefficients of $f_m(z)$ according to (15) are (Gaussian) integers for all $m = 1, \ldots, n$. It is then apparent that ϵ_m, δ_m are also (Gaussian) integers. ∎

Theorem 5 also provides means to prove the bound on the length of the coefficient growth of the FF algorithm stated in Theorem 2.

Proof for Theorem 2: Assume $\mathbf{T}_n$ is an integer real matrix and that its largest element, say r_{max}, has 'length' B. Say that (see footnote 2) this means that all $|r_i|$ are bounded by 2^B. The first in the following sequence of inequalities uses Theorem 5, i.e. (15), in conjunction with Hadamard's bound on the size of a determinant.

$$\mathrm{M}_{0,k}^{(m)} \leq \sqrt{\prod_{i=1}^{m}\left(\sum_{j=1}^{m} r_{max}^2\right)} \leq \sqrt{\prod_{i=1}^{m}\sum_{j=1}^{m}(2^B)^2}$$
$$\leq \sqrt{\prod_{i=1}^{m} m(2^B)^2} \leq (m(2^B)^2)^{m/2}$$

Take base two logarithm from both sides yields (10). The above argument needs adjustments when the matrix is of Gaussian integers. To account for the norm of a complex number whose real and imaginary parts are bounded each by 2^B, we have to replace $r_{max}^2 = (2^B)^2$ by $|r_{max}|^2 = 2(2^B)^2$. Consequently B in (10) should be replaced by $B + 1/2$ that does not change the bound asymptotically. (Alternatively, it is possible from the outset to devise a definition for the bound "B" on the length of the integers in a manner that makes (10) equally applicable for both the real and the complex cases.) ∎

III. Non-Hermitian case

A. The ordinary algorithm

Let $a_m(z) = \sum_{i=0}^{m} a_{m,i}z^i$ and $b_m(z) = \sum_{i=0}^{m} b_{m,i}z^i$ and let $\overset{\leftrightarrow}{b}$ denote reversion (without conjugate of complex values!) of a vector b, i.e. $\overset{\leftrightarrow}{b}_m(z) = \sum_{i=0}^{m} b_{m-i,i}z^i$. Here is the extension of the Algorithm 1 to a non-Hermitian Toeplitz matrix.

Algorithm 3 [The ordinary algorithm for a non-Hermitian Toeplitz matrix]. Consider a (strongly regular) non-Hermitian Toeplitz matrix $\mathbf{T}_n$ (1) with complex entries. Set $a_0(z) = 1$, $b_0(z) = 1$, $D_0 = r_0$.

For $m = 1, \ldots, n$ do:

$$k_m = \frac{[a_{m-1,0}, \cdots, a_{m-1,m-1}][r_1, \cdots, r_m]^T}{D_{m-1}} \quad (16a)$$

$$\xi_m = \frac{[b_{m-1,0}, \cdots, b_{m-1,m-1}][r_{-1}, \cdots, r_{-m}]^T}{D_{m-1}} \quad (16b)$$

$$\begin{bmatrix} a_m(z) \\ \overset{\leftrightarrow}{b}_m(z) \end{bmatrix} = \begin{bmatrix} 1 & -k_m \\ -\xi_m & 1 \end{bmatrix} \begin{bmatrix} za_{m-1}(z) \\ \overset{\leftrightarrow}{b}_{m-1}(z) \end{bmatrix} \quad (16c)$$

$$D_m = (1 - \xi_m k_m)D_{m-1} \quad (16d)$$

Then the coefficients of the polynomials $a_m(z)$, $b_m(z)$ and the scalars D_m, $m = 0, 1, \ldots, n$ solve the sets of equations (3).

The algorithm, first obtained in [2], is somewhat tricky and puzzled researchers [3] because it follows a recursion form typical to the Hermitian case without conforming to this case. It was put in a proper perspective only in [4] after it was shown there that the treatment of not symmetric Toeplitz matrix should truly involves two pairs of recursions; one that handles (3a) and the other (3b). The two pair of recursions are unavoidable when a somewhat more general case (of non-symmetric quasi-Toeplitz matrices) is considered. The are still desirable also for a non-Hermitian Toeplitz matrix in order to realize the algorithm in terms of causal flow of signals in lattices. However, from the strict point of view of computation, the information needed to solve (3a,b) for a Toeplitz matrix becomes available in either of the two pairs of recursions. Thus the above single pair of recursions suffices. The above presentation repeats [4, eqs. 3.26-7] except to removing the restriction to $r_0 = 1$ (see footnote 1 here). For more details we refer the interested reader to section 3.5 in [4].

B. The IP algorithm

The fraction-free counterpart of Algorithm 3 is as follows.

Algorithm 4 [FF Levinson algorithm for a non-Hermitian Toeplitz matrix]. Denote $f_m(z) = \sum_{i=0}^{m} f_{m,i}z^i$ and $g_m(z) = \sum_{i=0}^{m} g_{m,i}z^i$ and consider a (strongly regular) non-Hermitian Toeplitz matrix $\mathbf{T}_n$ (1) with complex entries. Set $\epsilon_{-1} = 1$, $f_0(z) = 1$, $g_0(z) = 1$ and $\epsilon_0 = r_0$.

For $m = 1, \ldots, n$, do:

$$\delta_m = [f_{m-1,0}, \cdots, f_{m-1,m-1}][r_1, \cdots, r_m]^T \quad (17a)$$

$$\zeta_m = [g_{m-1,0}, \cdots, g_{m-1,m-1}][r_{-1}, \cdots, r_{-m}]^T \quad (17b)$$

$$\begin{bmatrix} f_m(z) \\ \overset{\leftrightarrow}{g}_m(z) \end{bmatrix} = \frac{1}{\epsilon_{m-2}} \begin{bmatrix} \epsilon_{m-1} & -\delta_m \\ -\zeta_m & \epsilon_{m-1} \end{bmatrix} \begin{bmatrix} zf_{m-1}(z) \\ \overset{\leftrightarrow}{g}_{m-1}(z) \end{bmatrix} \quad (17c)$$

$$\epsilon_m = \frac{\epsilon_{m-1}^2 - \zeta_m\delta_m}{\epsilon_{m-2}} \quad (17d)$$

The coefficients of the polynomials solve the next two sets of equations

$$\mathbf{T}_m[f_{m,0}, \cdots, f_{m,m-1}, f_{m,m}]^T = [0, \cdots, 0, \epsilon_m]^T, \; m = 0, \ldots, n \quad (18a)$$

$$[g_{m,0}, \cdots, g_{m,m-1}, g_{m,m}]\mathbf{T}_m = [0, \cdots, 0, \epsilon_m], \; m = 0, \cdots, n \quad (18b)$$

Comparison of leading coefficients, step after step, in the two polynomial recursions reveal that

$$f_{m,m} = g_{m,m} = \epsilon_{m-1} \quad (19)$$

One way to establish Algorithm 3 and its IP property is to begin with a division-free (hence IP) form the algorithm and then show that it is the result of diving out recursively common factors that it contains [7]. As an alternative way, it is to take Algorithm 3 as a given postulation and show next that it creates solutions to (18) because it produces polynomials proportional to respective ones in Algorithm 2. It then remains to show that in spite of the divisions the algorithm remains IP. This can be done as follows. A repetition of the proof brought for Theorem 3 shows

Theorem 6. *The ϵ_m's produced by Algorithm 4 are equal the principal minors* (4) *of the non-Hermitian $\boldsymbol{T}_n$,*

$$\epsilon_m = \Delta_m \; , \; m = 0, 1, \ldots, n \quad (20)$$

Combining the above with (19) shows

Theorem 7. *The leading coefficients of the each of the two sequences that Algorithm 4 produces are equal to the principal minors of the non-Hermitian Toeplitz matrix $\boldsymbol{T}_n$ as follows*

$$f_{m,m} = g_{m,m} = \Delta_{m-1} \; , \; m = 1, \ldots, n \quad (21)$$

From here it can be shown that actually all the coefficients of all the polynomials are related to certain minors of $\mathbf{T}_n$ as follows.

Theorem 8. *The polynomials that Algorithm 4 produces can be expressed in terms of the cofactors of the matrix in the set of equations that its coefficient vector solves as follows.*

$$f_m(z) = C^{(m)}_{m,0} + C^{(m)}_{m,1} z + \ldots + C^{(m)}_{m,m} z^m \ , \ m = 1, \ldots, n \tag{22}$$

$$g_m(z) = C^{(m)}_{0,m} + C^{(m)}_{1,m} z + \ldots + C^{(m)}_{m,m} z^m \ , \ m = 1, \ldots, n \tag{23}$$

Proof: The coefficients of $f_m(z)$ solve here (18a), a same equation that it solves also in the Hermitian case (8). Therefore, the proof for Theorem 5, via (12) and (13), can be repeated to show (22). The proof for (23) follows after using instead (18b) to write the coefficient vector of $g_m(z)$ as a scaled first row of $\mathbf{T}_n^{-1}$ and then expressing the later by the cofactor matrix (12). ∎

Theorems 7 and 8 are the non-Hermitian Toeplitz parallel of Theorems 4 and 5. Clearly, these properties hold also for non-integer matrices. However, Theorem 8 implies that Algorithm 4 is FF over integers.

Theorem 9. *Algorithm 4 is integer preserving. If r_m , $m = 0, \pm 1, \ldots \pm n$, are in $\mathbb{Z}[\boldsymbol{i}]$ (resp. in $\mathbb{Z}$) then $\{f_{m,i}, g_{m,i}, \ i = 0, \ldots m\}$, δ_m, ζ_m and ϵ_m are in $\mathbb{Z}[\boldsymbol{i}]$ (resp. in $\mathbb{Z}$) for $m = 0, \ldots, n$.*

It also can be used to conclude that the growth of integers in Algorithm 4 is restrained like the growth of integers in Algorithm 2.

Theorem 10. *The length of the coefficients of $f_m(z)$ and $g_m(z)$ that Algorithm 4 creates, are upped-bounded by* (10)*, where B represent a bound on the length of the largest entry of the integer matrix $\boldsymbol{T}_n$.*

Proof: Apply Hadamard bounds to the coefficients of $f_m(z)$ and $g_m(z)$ expressed by (22) and (23) similar to the proof for Theorem 2. ∎

IV. Triangular factorization.

It is well known that the Levinson algorithms 1 and 3 also produce triangular factorization of the inverse of the Toeplitz matrix of the set of equations that they solve. The implied factorizations follow from (2) and (3) that they solve, respectively.

A similar property holds also for the IP Levinson algorithms. Arrange the solution vectors in (7) and (18) that are produced by Algorithm 2 and 4 into matrices as follows,

$$\mathbf{F}_n = \begin{bmatrix} f_{0,0} & f_{1,0} & \cdots & f_{n,0} \\ 0 & f_{1,1} & \cdots & f_{n,1} \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & f_{n,n} \end{bmatrix}, \mathbf{G}_n = \begin{bmatrix} g_{0,0} & g_{1,0} & \cdots & g_{n,0} \\ 0 & g_{1,1} & \cdots & g_{n,1} \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & g_{n,n} \end{bmatrix}$$

and also form the diagonal matrix

$$\mathbf{E}_n = diag\left[\epsilon_{-1}\epsilon_0, \epsilon_0\epsilon_1, \cdots, \epsilon_{n-1}\epsilon_n\right]$$

Then, as was shown in [5], Algorithm 2 produces the following triangular factorization for $\mathbf{T}_n^{-1}$ (^{H} denotes conjugate transpose)

$$\mathbf{T}_n^{-1} = \mathbf{F}_n \mathbf{E}_n^{-1} \mathbf{F}_n^H \tag{24}$$

Similarly, it can be shown that (18) in combination with (19) implies that Algorithm 4 produces the triangular factorization

$$\mathbf{T}_n^{-1} = \mathbf{F}_n \mathbf{E}_n^{-1} \mathbf{G}_n^T \tag{25}$$

The IP properties of Algorithms 2 and 4 imply that the factorizations (24) and (25) are over (Gaussian) integers when $\mathbf{T}_n$ is a (Gaussian) integer matrix. Furthermore, by properties that were established for these algorithms, these factorizations constitute in general (i.e. with the exception of coincidental common integer factors) the integer triangular factors with the lowest possible size of integers.

V. Numerical Example

As a numerical illustration, when Algorithm 4 is run for the following Gaussian-integer Toeplitz matrix

$$\mathbf{T}_3 = \begin{bmatrix} 3 & 2+j & 2j & 1+j \\ 2j & 3 & 2+j & 2j \\ 1+j & 2j & 3 & 2+j \\ 2+j & 1+j & 2j & 3 \end{bmatrix}$$

it solves (18) successively with the following results.

$\epsilon_0 = 3, \delta_1 = 2+j, \zeta_1 = 2j$
$f_1(z) = (-2-j) + 3z$
$g_1(z) = -2j + 3z$
$\epsilon_1 = 11-4j, \delta_2 = -3+2j, \zeta_2 = 7+3j,$
$f_2(z) = (3-2j) + (-10-3j)z + (11-4j)z^2$
$g_2(z) = (-7-3j) + (1-3j)z + (11-4j)z^2$
$\epsilon_2 = 44-31j, \delta_3 = 29-14j, \zeta_3 = 36-13j$
$f_3(z) = (-29+14j) + (9-4j)z + (-28+5j)z^2 + (44-31j)z^3$
$g_3(z) = (-36+13j) + (-2+5j)z + (-9-8j)z^2 + (44-31j)z^3$
$\epsilon_3 = 63-145j$

From here, it is also possible to express $\mathbf{T}_3^{-1}$ by (25) with

$$\mathbf{F}_3 = \begin{bmatrix} 1 & -2-j & 3-2j & -29+14j \\ 0 & 3 & -10-3j & 9-4j \\ 0 & 0 & 11-4j & -28+5j \\ 0 & 0 & 0 & 44-31j \end{bmatrix}, \mathbf{G}_3 = \begin{bmatrix} 1 & -2j & -7-3j & -36+13j \\ 0 & 3 & 1-3j & -2+5j \\ 0 & 0 & 11-4j & -9-8j \\ 0 & 0 & 0 & 44-31j \end{bmatrix}$$

$$\mathbf{E}_3 = diag\left[3, 33-12j, 360-517j, -1722-8333j\right]$$

VI. Conclusion

The paper considered the inversion of Toeplitz Hermitian and non-Hermitian matrices and the solution of such set of equations by a new Levinson algorithm that runs over integers for (real and complex) integer matrices. These IP algorithms are valuable for symbolic computation and provide powerful means to combat numerical rounding error in the respective ordinary algorithm. It was shown that the polynomial coefficients are equal up to a sign (of the cofactor) to certain minors of the matrix with the leading coefficients equal to the principal minors of the Toeplitz matrix. These relations provide a simple proof for the IP property. They were also used to prove that the algorithms feature nearly linear growth of integer lengths. The latter property is an important indicator for the efficiency of the algorithms as integers algorithms. The measure for the efficiency of an integer algorithm is in terms of its binary complexity, a measure based on estimating its number of arithmetic operations between bits, e.g. [6]). A more comprehensive complexity analysis, that could not be accommodated within the current presentation, shows that these algorithms have very low binary complexity. Thus, the presented fraction-free Levinson algorithms are very efficient integer algorithms.

References

[1] N. Levinson, "The Wiener RMS error criterion in filter design and prediction", *J. Math. Phys.*, vol. 25, pp. 261-278, 1947.

[2] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices", SIAM J. Appl. Math. Vol. 12, pp. 515-522, 1964.

[3] T. Kailath, A. Vieira and M. Morf, "Inverse of Toeplitz operators, innovations and orthogonal polynomials", *SIAM rev.* Vol. 20, pp. 106-119, 1978.

[4] Y. Bistritz and T. Kailath, "Inversion and factorization of non-Hermitian quasi-Toeplitz matrices" in *Linear Algebra and its Applications,* vol. 98, pp. 77-121, January 1988.

[5] Y. Segalov and Y. Bistritz, "Levinson algorithm over integers for strongly regular Hermitian Toeplitz matrices" *Proc. of the 2008 IEEE Int. Conference on Acoustics, Speech, and Signal Processing,* ICASSP-2008, Las Vegas, USA, April 2008.

[6] P. G. Anderson, M.R. Garey and L.E. Heindel, "Computational aspects of deciding if all roots of a polynomial lie within the unit circle", *Computing* vol. 16, pp. 293-304, 1976.

[7] Y. Bistritz and Y. Segalov "Integer Levinson algorithms for quasi-Toeplitz matrices" *Proc. of the 2010 American Control Conference* ACC2010, June 2010, Baltimore, Maryland, USA.