STABILITY OF THE LEVINSON ALGORITHM FOR TOEPLITZ-LIKE SYSTEMS

P. FAVATI†, G. LOTTI‡, AND O. MENCHI§

Abstract. Numerical stability of the Levinson algorithm, generalized for Toeplitz-like systems, is studied. Arguments based on the analytic results of an error analysis for floating point arithmetic produce an upper bound on the norm of the residual vector, which grows exponentially with respect to the size of the problem. The base of such an exponential function can be small for diagonally dominant Toeplitz-like matrices. Numerical experiments show that, for these matrices, Gaussian elimination by row and the Levinson algorithm have residuals of the same order of magnitude. As expected, the empirical results point out that the theoretical bound is too pessimistic.

Key words. Levinson algorithm, Toeplitz-like matrices, stability

AMS subject classifications. 15A06, 15B05

DOI. 10.1137/090753619

1. Introduction. Toeplitz systems arise frequently in linear algebra (see [2] for a list of possible sources), and special fast and superfast algorithms have been devised to solve them. Starting from the original Durbin algorithm to solve the Yule–Walker equations, the Levinson algorithm has been proposed for symmetric positive definite Toeplitz matrices and extended to the case of general Toeplitz matrices [8]. The Levinson algorithm is a fast method, i.e., it has a cost of $O(N^2)$ operations, $N$ being the size of the system. Unfortunately, when a simple operation like multiplication or inversion or low rank modification is applied to a Toeplitz matrix, the Toeplitz structure is lost and more general structures must be considered. The class of Toeplitz-like matrices, which is closed for the most common operations applied in numerical algorithms, seems ideal from this point of view. It is based on the concept of displacement rank introduced in [15] and has been studied by many authors (see, for example, [11, 13, 14]). The displacement operator allows a compact representation of the matrices of this class by means of a set of generators. The standard situation, when dealing with a structured matrix, assumes that its entries are not exactly known but can be computed from the generators when they are needed.

For the Toeplitz-like matrices, fast and superfast algorithms have been devised as well (see the extensive bibliography in [14]), but the question of their stability is still a matter of discussion. The Levinson algorithm, too, has been generalized for this class of matrices, maintaining its computational cost [6, 13].

The numerical stability of the original Levinson algorithm for symmetric positive definite matrices has been proved in [2, 5]. Look-ahead modifications have been proposed in [3, 4] to obviate instability in the general Toeplitz case. We are interested in studying the stability of the Levinson algorithm generalized for Toeplitz-like systems, taking into consideration also the error due to the recovering of the entries of the matrix from the generators.

* Received by the editors March 23, 2009; accepted for publication (in revised form) by N. Mastroardi June 28, 2010; published electronically August 31, 2010. This work was partially supported by MIUR, grant PRIN 20083KLJEZ, Italy.

http://www.siam.org/journals/simax/31-5/75361.html

†Istituto di Informatica e Telematica (IIT), CNR 56124 Pisa, Italy (paola.favati@iit.cnr.it).
‡Department of Mathematics, Università di Parma, 43100 Parma, Italy (grazia.lotti@unipr.it).
§Department of Computer Science, Università di Pisa, 56127 Pisa, Italy (menchi@di.unipi.it).
The generalized version of the Levinson algorithm can be introduced in different ways (see, for example, [7, 16]). The version given here points out its relation with block Gaussian elimination, allowing the use of standard techniques to deal with stability issues. In fact, the generalized Levinson algorithm can be seen as composed of two parts: an outer part which recursively uses a block factorization of the matrix and an inner part which computes such a factorization according to the recursive formulas characteristic of the method.

In section 2 the problem is introduced, and the outer part of the algorithm is described. The properties of Toeplitz-like matrices, recalled in section 3, allow the formulation of the inner part of the algorithm, which is presented in section 4. A theoretical analysis of the stability of the algorithm is given in section 5. To this aim, some stability parameters are defined, and for comparison purposes the well-known error bound of Gaussian elimination is recalled. Finally, in section 6 numerical experiments are performed to investigate the various sources of instability detected by the theoretical results.

2. The problem. Let \( N \geq 2 \) be an integer, \( A_N = [a_{ij}] \) an \( N \times N \) matrix, and \( b_N = [b_i] \) an \( N \)-vector. We assume that \( A_N \) is strongly nonsingular, i.e., that all its leading principal minors \( A_n \) of size \( n \) are nonsingular for \( n = 1, \ldots, N \). Then the solution of the system

\[
A_N x_N = b_N
\]

can be found by recursively computing the solutions of the systems of increasing size

\[
A_n x_n = b_n \quad \text{for} \quad n = 1, \ldots, N,
\]

where \( b_n \) is the upper subvector of size \( n \) of \( b_N \). The starting point of the recursion is

\[
x_1 = \frac{b_1}{a_{1,1}}.
\]

We partition the minor \( A_n \) and the vector \( b_n \) as

\[
A_n = \begin{bmatrix} A_{n-1} & r_{n-1} \\ s_{n-1}^T & a_{n,n} \end{bmatrix}, \quad b_n = \begin{bmatrix} b_{n-1} \\ b_n \end{bmatrix}
\]

and consider the block factorization

\[
A_n = P_n Q_n, \quad P_n = \begin{bmatrix} A_{n-1} & 0_{n-1} \\ s_{n-1}^T & z_n \end{bmatrix}, \quad Q_n = \begin{bmatrix} I_{n-1} & A_{n-1}^{-1} r_{n-1} \\ 0_{n-1}^T & 1 \end{bmatrix},
\]

where \( 0_k \) is the null vector of size \( k \), \( I_k \) is the identity matrix of size \( k \), and

\[
z_n = a_{n,n} - s_{n-1}^T A_{n-1}^{-1} r_{n-1}
\]

is the Schur complement of \( A_{n-1} \) in \( A_n \). The hypothesis of strong nonsingularity of \( A_N \) guarantees that \( z_n \neq 0 \) at any step. In fact, (4) implies that

\[
\det A_n = \det P_n = z_n \det A_{n-1}
\]

and \( z_n \neq 0 \) if both \( A_{n-1} \) and \( A_n \) are nonsingular. If the \( z_n \)'s are too small, then this procedure will be unstable.
The solution of (2) satisfies

$$Q_n x_n = u_n, \quad \text{where} \quad P_n u_n = b_n.$$  \hspace{1cm} (6)

Then

$$u_n = \begin{bmatrix} x_{n-1} \\ w_n/z_n \end{bmatrix}, \quad \text{where} \quad w_n = b_n - s_{n-1}^T x_{n-1},$$  \hspace{1cm} (7)

and

$$x_n = \begin{bmatrix} x_{n-1} \\ 0 \end{bmatrix} + \frac{w_n}{z_n} \begin{bmatrix} y_{n-1} \\ 1 \end{bmatrix},$$  \hspace{1cm} (8)

where

$$A_{n-1} y_{n-1} = -r_{n-1}. \hspace{1cm} (9)$$

To transform these formulas into an effective algorithm, recursive relations for computing $y_n$ and $z_n$ are needed. They are particularly simple in the special case of Toeplitz and Toeplitz-like matrices, as we shall see in section 4.

3. **Toeplitz-like matrices.** Toeplitz-like matrices can be represented by means of a set of generators defined through a displacement operator. The idea of representing a Toeplitz-like matrix by using the generators has its origin in the Gohberg–Semencul formula for the inverse of a Toeplitz matrix. Given an $N \times N$ matrix $A_N$, we consider the displacement operator $\nabla$,

$$\nabla(A_N) = A_N - Z_N A_N Z_N^T = A_N - \begin{bmatrix} 0 & 0 \\ 0 & A_N \end{bmatrix},$$  \hspace{1cm} (10)

where $Z_N$ is the down-shift matrix of order $N$,

$$Z_N = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$  

The matrix $A_N$ is said to have displacement rank $m$ if $\nabla(A_N)$ has rank $m$, i.e., if $\nabla(A_N)$ can be expressed in the form

$$\nabla(A_N) = \sum_{i=1}^{m} c_N^{(i)} d_N^{(i)T},$$  \hspace{1cm} (11)

for suitable vectors $c_N^{(i)}$ and $d_N^{(i)}$ (called generators) of length $N$. The matrix $A_N$ is said to be Toeplitz-like if $m$ is small relative to the size $N$. For Toeplitz matrices, which have displacement rank 2, relation (11) becomes

$$\nabla(A_N) = e_1 d_N^{(1)T} + c_N^{(2)} e_1^T,$$

where $d_N^{(1)} = Z_N Z_N^T A_N e_1$, $c_N^{(2)} = A_N e_1$, $e_1$ being the first canonical vector.
The generators enable us to represent a Toeplitz-like matrix as the sum of products of lower and upper triangular factors [14]. The computation can be performed recursively. Let $C_N$ and $D_N$ be the $N \times m$ matrices formed by the columns $c_N^{(i)}$ and $d_N^{(i)}$,
\[
C_N = \begin{bmatrix} c_1^{(1)} & \cdots & c_1^{(m)} \\ \vdots & \ddots & \vdots \\ c_N^{(1)} & \cdots & c_N^{(m)} \end{bmatrix}, \quad D_N = \begin{bmatrix} d_1^{(1)} & \cdots & d_1^{(m)} \\ \vdots & \ddots & \vdots \\ d_N^{(1)} & \cdots & d_N^{(m)} \end{bmatrix}.
\]
Then
\[
A_N = Z_N A_N Z_N^T + C_N D_N^T.
\]
Let $E_n$ be the $n \times N$ matrix formed by the first $n$ rows of $I_N$. Multiplying relation (12) by $E_n$ on the right and by $E_n^T$ on the left shows that any principal minor $A_n = E_n A_N E_n^T$ of size $n$ can be written as
\[
A_n = \begin{bmatrix} 0 & 0 \\ 0_{n-1}^T & A_{n-1} \end{bmatrix} + C_n D_n^T,
\]
where $C_n = E_n C_N$ and $D_n = E_n D_N$ are obtained by taking the first $n$ rows of $C_N$ and $D_N$, respectively. Calling $\mathbf{r}_k^T$ and $\mathbf{d}_k^T$ the $k$th row of $C_N$ and of $D_N$, respectively, and comparing (13) with (3), we obtain the recursions
\[
a_{n,n} = a_{n-1,n-1} + \mathbf{r}_n^T \mathbf{d}_n,
\]
\[
r_{n-1} = \begin{bmatrix} 0 \\ r_{n-2} \end{bmatrix} + C_{n-1} \mathbf{d}_n,
\]
\[
s_{n-1}^T = \begin{bmatrix} 0 \\ s_{n-2}^T \end{bmatrix} + \mathbf{r}_n^T D_{n-1}^T.
\]
For a given matrix $A_N$, the matrices $C_N$ and $D_N$ satisfying (12) are not unique. This means that even if the vectors $c_N^{(i)}$ and $d_N^{(i)}$ are given as data of the problem, they can be replaced if it is necessary to gain stability. As a matter of fact, it is possible that the sum of the outer products in $\nabla(A_N)$, computed in floating point arithmetic, does not have rank $m$ or does not satisfy (10). In particular, we will see in section 5 that generators $C_N$ and $D_N$ having entries with very large absolute values, associated with a matrix $A_N$ with "relatively small" entries, may be a source of instability.

To reduce the entries of the generators, we can replace the given $C_N$ and $D_N$ with new $\hat{C}_N$ and $\hat{D}_N$, associated to the same $A_N$, by applying the following technique as suggested in [9].

Let $C_N = Q_N R_N$ be the $QR$ factorization of $C_N$. Then
\[
\nabla(A_N) = \hat{C}_N \hat{D}_N^T, \quad \text{where} \quad \hat{C}_N = Q_N, \hat{D}_N = D_N R_N^T.
\]
From [9], the orthonormality of the columns of $Q_N$ implies that
\[
\| \hat{C}_N \|_2 \| \hat{D}_N^T \|_2 \leq m \| \hat{C}_N \hat{D}_N^T \|_2.
\]
The relationship between the two-norm and the infinity-norm of a matrix implies that
\[
\| \hat{C}_N \| \| \hat{D}_N^T \|_\infty \leq \sqrt{N} \| \hat{C}_N \|_2 \| \hat{D}_N^T \|_2 \leq Nm \| \hat{C}_N \hat{D}_N^T \|_\infty.
\]
However, if $C_N$ has large entries, then so does $R_N$, so the corresponding small entries of $D_N$ may be corrupted by cancellation errors. Therefore, the computation of the QR factorization and of $D_N$ should be carried out in high precision arithmetic. The cost of this preprocessing phase is of order $O(m^2 N)$. Numerical experiments show that this reduces the largest components of the generators.

4. The generalized Levinson algorithm. Let $A_N$ be an $N \times N$ Toeplitz-like matrix with displacement rank $m$, defined by a set of generators $c_N^{(i)}$ and $d_N^{(i)}$, $i = 1, \ldots, m$. As noted in section 2, we must find recursive relations for computing $y_n$ and $z_n$. The generalized Levinson algorithm (in the following denoted by GLev) will be based on these relations.

As seen in (9), $y_n$ satisfies

$$A_n y_n = -r_n,$$

which is known as the Yule–Walker problem. We consider $m$ auxiliary vectors $t_n^{(i)}$ forming a matrix $F_n$,

$$F_n = \begin{bmatrix} f_1^{(1)} & \cdots & f_1^{(m)} \\ \vdots & \ddots & \vdots \\ f_n^{(1)} & \cdots & f_n^{(m)} \end{bmatrix}$$

such that $A_n F_n = C_n$.

The matrix $F_n$ satisfies a recursive relation similar to the one satisfied by $x_n$, i.e.,

$$F_n = \begin{bmatrix} F_{n-1} \\ 0_m \end{bmatrix} + \frac{1}{z_n} \begin{bmatrix} y_{n-1} \\ 1 \end{bmatrix} t_n^T,$$

where $t_n^T = \tau_n^T - s_{n-1}^T F_{n-1}$.

From (17) and (14),

$$\begin{bmatrix} 0 & 0 \\ 0 & A_{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix} = - \begin{bmatrix} 0 \\ r_{n-1} \end{bmatrix} = -r_n + C_n \overline{d}_{n+1},$$

and using (13) and (17) yields

$$(A_n - C_n D_n^T) \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix} = A_n y_n + C_n \overline{d}_{n+1}.$$ 

Hence

$$A_n y_n = A_n \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix} - C_n \left( D_n^T \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix} + \overline{d}_{n+1} \right),$$

and using (18) yields

$$y_n = \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix} - F_n v_n,$$

where $v_n = D_{n+1}^T \begin{bmatrix} 0 \\ y_{n-1} \end{bmatrix}$.

It remains to find a recursive relation for $z_n$. From (5), (17), and (14),

$$z_n - z_{n-1} = a_{n,n} + s_{n-1}^T y_{n-1} - a_{n-1,n-1} - s_{n-2}^T y_{n-2}$$

$$= \tau_n^T \overline{d}_n + s_{n-1}^T y_{n-1} - \begin{bmatrix} 0 \\ s_{n-2}^T \end{bmatrix} \begin{bmatrix} 0 \\ y_{n-2} \end{bmatrix}$$

$$= s_{n-1}^T \left( y_{n-1} - \begin{bmatrix} 0 \\ y_{n-2} \end{bmatrix} \right) + \tau_n^T \left( \overline{d}_n + D_{n-1}^T \begin{bmatrix} 0 \\ y_{n-2} \end{bmatrix} \right),$$
so (20) implies that

\[ (21) \quad z_n = z_{n-1} + t_n^T v_{n-1}. \]

Now we can give a detailed implementation of the algorithm GLev. The initial positions are as follows:

\[ z_1 = c_1^T \bar{d}_1, \quad x_1 = \frac{1}{z_1} [b_1], \quad F_1 = \frac{1}{z_1} c_1^T, \quad v_1 = \bar{d}_2, \quad y_1 = -F_1 v_1, \quad s_1 = [c_2^T \bar{d}_1]. \]

For \( n = 2, \ldots, N - 1 \), compute the following:

1. \( t_n^T = c_n^T - s_{n-1}^T F_{n-1} \)
2. \( z_n = z_{n-1} + t_n^T v_{n-1} \)
3. \( w_n = b_n - s_{n-1}^T x_{n-1} \)
4. \( x_n = \begin{bmatrix} x_{n-1} \ 0 \end{bmatrix} + \frac{w_n}{z_n} \begin{bmatrix} y_{n-1} \ 1 \end{bmatrix} \)
5. \( F_n = \begin{bmatrix} F_{n-1} \ 0_m \end{bmatrix} + \frac{1}{z_n} \begin{bmatrix} y_{n-1} \ 1 \end{bmatrix} t_n^T \)
6. \( v_n = D_n^{T} \begin{bmatrix} 0 \ y_{n-1} \ 1 \end{bmatrix} \)
7. \( y_n = \begin{bmatrix} 0 \ y_{n-1} \end{bmatrix} - F_n v_n \)
8. \( s_n^T = \begin{bmatrix} 0 \ s_{n-1}^T \end{bmatrix} + c_n^T D_n^T \)

Finally compute \( t_N, z_N, w_N, \) and \( x_N \). The execution of the algorithm should be interrupted if \( |z_n| \) is smaller than a preassigned small tolerance, indicating a possible singular minor.

The cost of the \( n \)th step of the algorithm can be expressed in terms of the number of multiplicative operations:

- \( x_n, s_{n-1}, v_n, \) and \( F_n \) require \((3m + 1)(n - 1) + m \) ops,
- \( y_n, w_n/z_n, \) and \( t_n/z_n \) require \((2m + 1)n \) ops,
- \( z_n \) requires \( m \) ops.

Summing for \( n = 1, \ldots, N \) yields the multiplicative cost \((5m/2 + 1)N^2 + O(N)\).

5. **Error analysis.** We assume that the computations are carried out in a floating point arithmetic with unit roundoff \( \epsilon \). The possible application of the technique (15) described in section 3 may require a higher precision. We use the following notations:

- The computed value of a variable (scalar, vector, or matrix) \( v \) is denoted by \( \bar{v} \) or by the “\( fl \)” notation. Following the standard model of floating point arithmetic, we assume that for any pair \( x \) and \( y \) of floating point numbers and for any “\( op \)” arithmetic operation,

\[ fl(x \ op \ y) = (x \ op \ y)(1 + \delta_{op}) \quad \text{with} \quad |\delta_{op}| \leq \epsilon. \]

- Given two \( n \times n \) matrices \( A \) and \( B \), \( |A| \leq |B| \) means that \( |a_{i,j}| \leq |b_{i,j}| \) for \( i, j = 1, \ldots, n \) and analogously for vectors.

- From now on, the infinity-norm is used for both matrices and vectors. For convenience, we drop the \( \infty \) subscript, i.e., \( |A| = ||A||_{\infty} \) and \( ||v|| = ||v||_{\infty} \). The one-norm \( ||v||_{1} \) is also used (always with the subscript).

- For simplicity, the term “\( + O(\epsilon^2) \)” is omitted in the proofs. Consequently, since \( x = O(\epsilon) \) and \( \bar{y} - y = O(\epsilon) \) imply \( x\bar{y} = xy + O(\epsilon^2) \), in the proofs the expression \( x\bar{y} \) is replaced by \( xy \).
We use the following bounds, which are standard in error analyses. (The proofs are straightforward; see, for example, [12], Chap. 3.) Let $p$, $q$ be two floating point scalars and $u$, $v$ two floating point $k$-vectors.

- If $t = u^T v$, then
  \[ t = t + \theta, \text{ where } |\theta| \leq k \epsilon |u|^T |v| + O(\epsilon^2). \]  

- If $t = p + u^T v$, then
  \[ t = t + \theta, \text{ where } |\theta| \leq \epsilon (|t| + k |u|^T |v|) + O(\epsilon^2). \]  

- If $t = (p - u^T v)/q$, then
  \[ (q + \eta) t + (u + \theta)^T v = p, \]
  where $|\eta| \leq (k + 1) \epsilon |q| + O(\epsilon^2)$, $|\theta| \leq (k + 1) \epsilon |u| + O(\epsilon^2)$.

- If $w = u + p v$, then
  \[ (I_k + \Delta I_k) \bar{w} = u + p (v + \theta), \]
  where $||\Delta I_k|| \leq \epsilon I_k + O(\epsilon^2)$, $|\theta| \leq \epsilon |v| + O(\epsilon^2)$.

Our aim is to analyze the effect of the rounding errors when GLev is used to solve (1). In the case of structured problems, the elements of $A_N$ are not given explicitly but are computed recursively from the generators during the application of the algorithm. Hence, for any $n$, we have to consider two matrices: $A_n$, the coefficient matrix which would be generated in exact arithmetic by $C_n$ and $D_n$, as expressed in (13), and $\tilde{A}_n$, the matrix effectively reconstructed from $C_n$ and $D_n$ using the recursion in finite arithmetic, i.e.,

\[ \tilde{A}_n = fl \left( \begin{bmatrix} 0 & 0_{n-1}^T \\ 0_{n-1} & \bar{A}_{n-1} \end{bmatrix} + C_n D_n^T \right). \]

Their difference $\Gamma_n = \tilde{A}_n - A_n$ can be regarded as a representation error. In the stability analysis of several algorithms for structured matrices, the propagation of such an error is also taken into consideration (see, for example, [1, 9, 10, 17]). Following this line, we analyze first the representation error and then the error produced by applying GLev to $A_N$. Conclusions on the behavior of the overall error will follow from the combination of the two analyses.

When we will analyze the error produced by applying GLev to $\tilde{A}_N$, a double notation will be required: the sign $\circ$ will be associated to variables denoting quantities computed in infinite arithmetic using the elements of $\tilde{A}_n$, while the sign $\bar{\circ}$ will be used to denote the corresponding quantities computed in finite arithmetic. For example,

\[ \begin{align*}
\circ^T t_n &= \bar{c}_n^T - \bar{s}_{n-1}^T F_{n-1}, \quad \bar{t}_n = fl \left( \bar{c}_n^T - \bar{s}_{n-1}^T \bar{F}_{n-1} \right), \\
\circ z_n &= \bar{z}_{n-1} + \circ^T v_{n-1}, \quad \bar{z}_n = fl \left( \bar{z}_{n-1} + \bar{t}_n^T \bar{v}_{n-1} \right)
\end{align*} \]

and analogously for $\bar{F}_n$, $\bar{v}_n$, and $\bar{y}_n$.

GLev can be regarded as composed of two parts: an outer part, which forms the framework of the algorithm and implements recursion (8) given $\bar{y}_{n-1}$ and $\bar{z}_n$, and an inner part, which computes $\bar{y}_{n-1}$ and $\bar{z}_n$ by using recursions (20) and (21). Following this approach, the error analysis of GLev will deal first with the computation of (8), assuming that the errors of the computed $\bar{y}_{n-1}$ and $\bar{z}_n$ are sufficiently bounded from above. Then we will analyze the conditions upon which the computation of (20) and (21) meets these stability requirements.
5.1. Stability parameters. We introduce three useful stability parameters.

- The parameter
  \[ \beta_n = \frac{\|\begin{bmatrix} C_n & D_n \end{bmatrix}^T \|}{\|A_n\|} \]
  measures the conditioning of the problem of representing matrix \( A_n \) by means of the generators.
  We saw in section 3 that a reduction of \( \beta_n \) can be obtained by performing an appropriate replacement of the generators in a preliminary step. (16) implies that \( \beta_n \leq 2nm \).

- The computation of \( y_n \), performed recursively by means of (20), might introduce cancelation errors induced by a large error in the computation of \( F_nv_n \). To control this form of instability, we introduce the parameter
  \[ \varphi_n = \frac{\|F_n\|}{1 + \|y_n\|} \]
  (28)
  (18), (20), (27), and (28) imply that
  \[ \varphi_n \leq \beta_n + 1 \frac{\|A_n^{-1}\|}{1 + \|y_n\|}, \]
  Under the assumption \( \|A_n\| \sim \|A_{n+1}\| \),
  \[ \varphi_n \leq \beta_{n+1} \kappa(A_n) \frac{1 + \|y_{n-1}\|}{1 + \|y_n\|}, \]
  where \( \kappa(A_n) \) is the condition number of \( A_n \). Hence, if \( A_n \) is ill-conditioned or the fraction in (29) is large (i.e., the norms of two consecutive \( y_n \) are very different), the parameter \( \varphi_n \) can be large also if \( \beta_{n+1} \) is small.

- The quantity
  \[ \pi_n = \prod_{k=1}^n (1 + \|y_k\|) \quad \text{with} \quad \pi_0 = 1 \]
  (30)
  plays an important role in the stability analysis. In fact, we will see that instability can be expected if \( \pi_{N-1} \) is large. The same assumption \( \|A_n\| \sim \|A_{n+1}\| \) yields \( \|y_n\| = \|A_n^{-1}r_n\| \leq \kappa(A_n) \). Hence large conditioning of intermediate leading principal minors of \( A_N \) may lead to instability. It is worth pointing out that the results obtained in the next subsections hold under the implicit assumption that the quantities \( \kappa(A_n) \) are not so large to invalidate a first order error analysis.

5.2. Error analysis of the computation of \( A_N \). The elements of \( A_N \) can be computed recursively by applying (14). Actually, only the last row of \( A_n \), \( n = 1, \ldots, N \), is explicitly computed by GLev.

**Theorem 1.** For \( n = 1, \ldots, N \),

\[ \overline{a}_n = A_n + \Gamma_n, \quad \text{where} \quad \|\Gamma_n\| \leq n \overline{\beta}_n \epsilon \|A_n\| + O(\epsilon^2) \]

with \( \overline{\beta}_n = 1 + m \max\{\beta_1, \ldots, \beta_n\} \).

**Proof.** The proof is by induction on \( n \). (14) implies that

\[ \Gamma_i = [\overline{a}_{i,1}] - [a_{i,1}] = ft(\overline{c}_i^2 \overline{d}_i) - \overline{c}_i^2 \overline{d}_i, \]
so (22) implies that
\[ \| F_n \| \leq m \epsilon |c| |D| \leq m \beta_1 \epsilon |\bar{c}^T \bar{d}| \leq \beta_1 \epsilon |a_{1,1}|. \]

For \( n \geq 2 \) from (26) and (23),
\[ \tilde{A}_n = \begin{bmatrix} 0 & 0^T_{n-1} \\ 0_{n-1} & \tilde{A}_{n-1} \end{bmatrix} + C_n D_n^T + \Theta_n = A_n + \begin{bmatrix} 0 & 0^T_{n-1} \\ 0_{n-1} & \Gamma_{n-1} \end{bmatrix} + \Theta_n, \]
where
\[ |\Theta_n| \leq \epsilon (|A_n| + m |C_n| |D_n|^T). \]

Hence
\[ \| F_n \| \leq \| F_{n-1} \| + \epsilon \left( \| A_n \| + m \| |C_n| |D_n|^T \right), \]
so, from (27),
\[ \| F_n \| \leq \| F_{n-1} \| + (1 + m \beta_n) \epsilon \| A_n \|. \]

Therefore the assertion follows from the induction hypothesis. \( \square \)

5.3. Stability of the outer part. The recursive relation (8), which forms the framework of the algorithm, is based on the block factorization (4) of the matrix \( A_n \).

Here we consider the factorization applied to \( \tilde{A}_n \). The computed approximations of the two factors of (4) are
\[ \tilde{P}_n = \begin{bmatrix} \tilde{A}_{n-1} & 0_{n-1} \\ 0^T_{n-1} & \tilde{s}_{n-1} \end{bmatrix} \quad \text{and} \quad \tilde{Q}_n = \begin{bmatrix} I_{n-1} & -\tilde{y}_{n-1} \\ 0^T_{n-1} & 1 \end{bmatrix}. \]

The error analysis of the outer part is similar to the one given for the point and block LU factorizations. The error of the factorization of \( \tilde{A}_n \) depends on the errors of \( \tilde{z}_n \) and \( \tilde{y}_{n-1} \). In fact,
\[ \tilde{P}_n \tilde{Q}_n = \tilde{A}_n + G_n, \]
where
\[ G_n = \begin{bmatrix} O_{n-1} & -\tilde{A}_{n-1} \Delta y_{n-1} \\ 0^T_{n-1} & \Delta z_n - \tilde{s}_{n-1}^T \Delta y_{n-1} \end{bmatrix}, \]
where \( \Delta y_{n-1} = \tilde{y}_{n-1} - \frac{\tilde{y}}{\tilde{y}_{n-1} - \tilde{s}_{n-1}} \) and \( \Delta z_n = \tilde{z}_n - \frac{\tilde{z}}{\tilde{z}_{n-1}} \).

**Theorem 2.** If for \( n = 1, \ldots, N \) a constant \( \gamma_n \) exists such that
\[ \| G_n \| \leq \gamma_n \pi_{n-1} \epsilon \| A_n \| + O(\epsilon^2), \]
then the computed solution \( \tilde{x}_n \) of the system with coefficient matrix \( \tilde{A}_n \) and right-hand side \( b_n \) satisfies
\[ (\tilde{A}_n + \Delta A_n) \tilde{x}_n = b_n \]
with \( |\Delta A_n| \leq H_n \) and \( \| H_n \| \leq \nu_n \pi_{n-1} \epsilon \| A_n \| + O(\epsilon^2), \]
where \( H_n \) depends on \( \epsilon \) and on \( \tilde{A}_n \) but not on \( b_n \) and where \( \nu_n \) depends on only \( n \) and \( \gamma_n \).
Proof. The proof is by induction on \( n \). The assertion is true for \( n = 1 \) with \( \nu_1 = 1 \).

In fact,

\[
(\tilde{a}_{1,1} + \tilde{a}_{1,1} \epsilon_1) \tilde{x}_1 = b_1,
\]

where \( \epsilon_1 \) is due to the computation of \( b_1/\tilde{a}_{1,1} \). Hence

\[
|\Delta A_1| = |\tilde{a}_{1,1} \epsilon_1| \leq \epsilon |a_{1,1}| = H_1.
\]

For \( n > 1 \) we proceed by analyzing the two systems (6) solved at each step of the algorithm. The first system to be solved is the one with coefficient matrix \( \tilde{P}_n \) and right-hand side \( b_n \). According to (7), its computed solution \( \tilde{u}_n \) has the subvector of the first \( n - 1 \) components equal to \( \tilde{x}_{n-1} \) and a last component \( \tilde{u}_n = f l(\tilde{w}_n/\tilde{z}_n) \). By the induction hypothesis,

\[
(\tilde{A}_{n-1} + \Delta A_{n-1}) \tilde{x}_{n-1} = b_{n-1} \quad \text{with} \quad \|\Delta A_{n-1}\| \leq \nu_{n-1} \pi_{n-2} \epsilon \|A_{n-1}\|,
\]

and, by (24),

\[
(\tilde{s}_{n-1} + \sigma_{n-1})^T \tilde{x}_{n-1} + (\tilde{z}_n + \zeta_n) \tilde{u}_n = b_n
\]

with

\[
|\sigma_{n-1}| \leq n \epsilon |s_{n-1}| \quad \text{and} \quad |\zeta_n| \leq n \epsilon |z_n|.
\]

Hence

\[
(\tilde{P}_n + \Delta P_n) \tilde{u}_n = b_n, \quad \text{where} \quad \Delta P_n = \begin{bmatrix} \Delta A_{n-1} & 0_{n-1} \\ \sigma_{n-1}^T & \zeta_n \end{bmatrix}.
\]

The second system to be solved is the one with coefficient matrix \( \tilde{Q}_n \) and right-hand side \( \tilde{u}_n \). By (25) its computed solution \( \tilde{x}_n \) satisfies

\[
(\tilde{Q}_n + \Delta Q_n) \tilde{x}_n = \tilde{u}_n, \quad \text{where} \quad \Delta Q_n = \begin{bmatrix} \Delta I_{n-1} & \theta_{n-1} \\ 0_{n-1}^T & 0 \end{bmatrix}
\]

with

\[
|\Delta I_{n-1}| \leq \epsilon I_{n-1} \quad \text{and} \quad |\theta_{n-1}| \leq \epsilon |y_{n-1}|.
\]

Substituting (39) into (38) yields

\[
(\tilde{P}_n + \Delta P_n) (\tilde{Q}_n + \Delta Q_n) \tilde{x}_n = b_n,
\]

and, from (34),

\[
(\tilde{A}_n + \Delta A_n) \tilde{x}_n = b_n, \quad \text{where} \quad \Delta A_n = G_n + \Delta P_n \tilde{Q}_n + \tilde{P}_n \Delta Q_n.
\]

Hence

\[
|\Delta A_n| \leq |G_n| + |\Delta P_n| |\tilde{Q}_n| + |\tilde{P}_n| |\Delta Q_n|.
\]
where (38) implies that
\[
|\Delta P_n| \leq M_n = \begin{bmatrix} H_{n-1} & 0_{n-1} \\ |\sigma_{n-1}|^T & |\zeta_n| \end{bmatrix}.
\]
The matrix
\[
H_n = |G_n| + M_n |\tilde{Q}_n| + |\tilde{P}_n| |\Delta Q_n|,
\]
which does not depend on $b_n$, satisfies $|\Delta A_n| \leq H_n$. Moreover, since (5) and (9) imply that
\[
|z_n| \leq |a_{n,n}| + \|s_{n-1}\|_1 \|y_n-1\|,
\]
using (37), (40), and the induction hypothesis yields
\[
\|M_n |\tilde{Q}_n|\| \leq \max \left\{ |H_{n-1}| (1 + |y_{n-1}|), n \epsilon \left( |s_{n-1}|_1 (1 + |y_{n-1}|) + |z_n| \right) \right\}
\leq \max \left\{ |H_{n-1}|, 2n \epsilon \left( |s_{n-1}|_1 + |a_{n,n}| \right) (1 + |y_{n-1}|) \right\}
\leq \nu_{n-1} \pi_{n-1} \epsilon \max \left\{ \|A_{n-1}\|, \left( |s_{n-1}|_1 + |a_{n,n}| \right) \right\} \leq \nu_{n-1} \pi_{n-1} \epsilon \|A_n\|.
\]
We have assumed that $\nu_{n-1} \geq 2n$, which is certainly true already for small values of $n$, as we will see at the end of the next subsection. Analogously, (39) implies that
\[
\| |\tilde{P}_n| |\Delta Q_n|\| \leq \epsilon \max \left\{ \|A_{n-1}\|, |s_{n-1}|_1 \right\} (1 + |y_{n-1}|) \leq \epsilon \|A_n\| (1 + |y_{n-1}|).
\]
Finally, from (35),
\[
\|H_n\| \leq (\gamma_n + \nu_{n-1} + 1) \pi_{n-1} \epsilon \|A_n\|.
\]
Then the assertion follows with $\nu_n = \gamma_n + \nu_{n-1} + 1$.

Remark. In the case of a positive definite symmetric Toeplitz matrix, the first component of the vector $y_n$ coincides with the last component $K_n$ of the vector $a_n$ defined in eq. (2.11) of [5]. The stability analysis carried out in [5] shows that, in this particular case, the computed solution of the Yule–Walker problem is the exact solution of a system with a right-hand side perturbed by a vector bounded in norm by the quantity $K = \prod_{i=1}^N (1 + |K_i|)$ (see Thm. 4.1 in [5]). Hence the quantity $\pi_{N-1}$ on which bound (36) depends can be regarded as a generalization of $K$.

5.4. Stability of the inner part. The error analysis of the inner part of GLev concerns the computation of $F_n$, $\tilde{y}_n$, and $\tilde{z}_n$ and has the aim of showing that bound (35), assumed in Theorem 2, can be met by GLev.

Theorem 3. Let $\tilde{F}_n$, $\tilde{y}_n$, and $\tilde{z}_n$ be computed according to (19), (20), and (21), respectively, and let $g_n$ be the last column of the matrix $G_n$ defined in (34). Then
\[
\|g_n\| \leq \gamma_n \pi_{n-1} \epsilon \|A_n\| + O(\epsilon^2),
\]
where $\gamma_n$ depends on $m$, $n$, $\nu_{n-1}$, and the stability parameters.

Proof. The proof is by induction on $n$. The assertion is obviously true for $n = 1$ with $\gamma_1 = 0$. For $n > 1$ we assume that relation (41) holds for index $n$ and proceed with index $n + 1$.

According to (19) the columns $\tilde{y}_n^{(i)}$ of $\tilde{F}_n$ are computed from the columns $c_n^{(i)}$ of $C_n$ in the same way as the vector $\tilde{x}_n$ is computed from $b_n$. By the induction
hypothesis, the assumption (35) of Theorem 2 at size \( n \) holds when GLev is applied. Hence the solution of each system having \( \tilde{A}_n \) as the coefficient matrix and \( c_n^{(i)} \) as the right-hand side satisfies (36) with a different perturbation \( \Phi_n^{(i)} \) replacing \( \Delta A_n \), i.e., for \( i = 1, \ldots, m \), we have

\[
(\tilde{A}_n + \Phi_n^{(i)})\tilde{F}_n^{(i)} = c_n^{(i)}, \quad \text{where} \quad |\Phi_n^{(i)}| \leq H_n
\]

with

\[
\|H_n\| \leq \nu_n \pi_n - 1 \epsilon \|A_n\|.
\]

In the following we need a bound of the norm of the vector \( (C_n - \tilde{A}_n \tilde{F}_n)u \), where \( u \) is any vector of \( m \) components. Therefore

\[
(C_n - \tilde{A}_n \tilde{F}_n)u = \sum_{j=1}^{m} \Phi_n^{(j)} \tilde{F}_n^{(j)} u_j.
\]

Since the bound on \( |\Phi_n^{(i)}| \) does not depend on \( i \),

\[
\|(C_n - \tilde{A}_n \tilde{F}_n)u\| \leq \nu_n \pi_n - 1 \epsilon \|A_n\| \|\tilde{F}_n\| \|u\|.
\]

Let us now go back to our primary aim, i.e., showing that (41) holds, which requires a relation between the two vectors

\[
\tilde{y}_n = fl \left[ \begin{array}{c} 0 \\ -\tilde{A}_{n-1} \Delta y_{n-1} \\ \Delta z_n - \bar{s}_{n-1}^T \Delta y_{n-1} \end{array} \right] - \tilde{F}_n \tilde{v}_n = \left[ \begin{array}{c} 0 \\ 0 \\ \Delta z_{n+1} - \bar{s}_n^T \Delta y_n \end{array} \right] + \eta_n
\]

with

\[
|\eta_n| \leq \epsilon (|y_n| + m |F_n| |v_n|).
\]

Hence

\[
\Delta y_n = \left[ \begin{array}{c} 0 \\ \Delta y_{n-1} \end{array} \right] - \tilde{F}_n \tilde{v}_n - \bar{o} \bar{s}_n - \bar{o} \bar{s}_{n-1} + \eta_n.
\]

(32), applied to the matrix obtained by removing the last column from \( \tilde{A}_{n+1} \), implies that

\[
\begin{bmatrix} \tilde{A}_n \\ \bar{s}_n^T \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \tilde{A}_{n-1} \\ \bar{s}_{n-1} & 0 \end{bmatrix} + C_{n+1} D_n^T + \bar{\Theta}_{n+1},
\]

where \( \bar{\Theta}_{n+1} \) is the matrix obtained by removing the last column from \( \Theta_{n+1} \). (45) implies that

\[
(46) \quad - \begin{bmatrix} \tilde{A}_n \\ \bar{s}_n^T \end{bmatrix} \Delta y_n = - \begin{bmatrix} 0 \\ -\tilde{A}_{n-1} \Delta y_{n-1} \\ \bar{s}_{n-1} \Delta y_{n-1} \end{bmatrix} + \omega,
\]
where
\[
\omega = - (\Omega_{n+1} + C_{n+1}D^T_n) \begin{bmatrix} 0 \\ \Delta y_{n-1} \end{bmatrix} + \begin{bmatrix} \tilde{A}_n \\ \tilde{s}_n \end{bmatrix} (\tilde{F}_n \tilde{v}_n - \tilde{F}_n \tilde{v}_n - \eta_n).
\]

Let \(g_n\) be the vector obtained by removing the last element from \(g_n\). By the induction hypothesis, \(\|g_n\| = O(\epsilon)\). As noted at the end of subsection 5.1, we are implicitly assuming that all the matrices \(\tilde{A}_n\) are not so ill-conditioned to prevent \(\|\Delta y_{n-1}\| = \|\tilde{A}^{-1}_{n-1} g_n\|\) to be first order in \(\epsilon\). Hence \(\|\Theta_{n+1} \begin{bmatrix} 0 \\ \Delta y_{n-1} \end{bmatrix}\) is second order in \(\epsilon\) and thus negligible. Then
\[
\omega = - C_{n+1}D^T_n \begin{bmatrix} 0 \\ \Delta y_{n-1} \end{bmatrix} + \begin{bmatrix} \tilde{A}_n \\ \tilde{s}_n \end{bmatrix} (\tilde{F}_n \tilde{v}_n - \tilde{F}_n \tilde{v}_n - \eta_n).
\]

Using (23), (21) implies that
\[
\tilde{z}_{n+1} = fl (\tilde{z}_n + t_{n+1}^T \tilde{v}_n) = \tilde{z}_n + t_{n+1}^T \tilde{v}_n + \xi_n,
\]
where
\[
|\xi_n| \leq \epsilon (|z_{n+1}| + m |t_{n+1}|^T |v_n|),
\]
and then
\[
\Delta z_{n+1} = \Delta z_n + t_{n+1}^T \tilde{v}_n - t_{n+1}^T v_n + \xi_n.
\]

Combining (46) and (49) yields
\[
g_{n+1} = \begin{bmatrix} 0 \\ g_n \end{bmatrix} + \omega_{n+1}, \text{ where } \omega_{n+1} = \omega + \begin{bmatrix} \tilde{A}_n \\ \tilde{s}_n \end{bmatrix} (\tilde{F}_n \tilde{v}_n - \tilde{F}_n \tilde{v}_n - \eta_n).
\]

Now we examine separately the terms of \(\omega_{n+1}\).

1. Using (22), (20) implies that
\[
\tilde{v}_n = fl \left( D^T_{n+1} \begin{bmatrix} 0 \\ \tilde{y}_{n-1} \end{bmatrix} \right) = D^T_{n+1} \begin{bmatrix} 0 \\ \tilde{y}_{n-1} \end{bmatrix} + \theta_n,
\]
where
\[
|\theta_n| \leq n \epsilon |D_{n+1}|^T \begin{bmatrix} 0 \\ |y_{n-1}| \\ 1 \end{bmatrix}.
\]

Hence
\[
\Delta v_n = \tilde{v}_n - v_n = D^T_n \begin{bmatrix} 0 \\ \Delta y_{n-1} \end{bmatrix} + \theta_n,
\]
so
\[
C_{n+1}D^T_n \begin{bmatrix} 0 \\ \Delta y_{n-1} \end{bmatrix} = C_{n+1} \Delta v_n - C_{n+1} \theta_n.
\]
2. (18) implies that
\[
\tilde{A}_n \left( \tilde{F}_n \tilde{v}_n - \phi_n \phi_n \right) = C_n \Delta v_n - \Psi_n,
\]
where, for (43),
\[
\Psi_n = \left( C_n - \tilde{A}_n \tilde{F}_n \right) \tilde{v}_n \quad \text{with} \quad \| \Psi_n \| \leq \nu_n \pi_{n-1} \epsilon \| F_n \| \| v_n \|.
\]

3. Using (23), (19) implies that
\[
\tilde{t}_n^T = f \left( (c_n^T - \tilde{s}_n^T \tilde{F}_n) \tilde{v}_n \right) = c_n^T \tilde{v}_n - \tilde{s}_n^T \tilde{F}_n + \tau_{n+1}^T,
\]
where
\[
|\tau_{n+1}^T| \leq \epsilon \left( |t_{n+1}^T| + n |s_n^T | | F_n | \right).
\]
Hence
\[
\tilde{s}_n^T (\tilde{F}_n \tilde{v}_n - \phi_n \phi_n) + \tilde{t}_n^T \tilde{v}_n - t_{n+1}^T \tilde{v}_n = \tilde{c}_n^T \Delta v_n + \tau_{n+1}^T \tilde{v}_n.
\]
From (50), (53), (54), and (57),
\[
\omega_{n+1} = \left[ \begin{array}{c} -\Psi_n \\ \tau_{n+1}^T \tilde{v}_n + \xi_n \\ \tilde{A}_n \\ \tilde{s}_n^T \end{array} \right] \eta_n + C_{n+1} \theta_n.
\]

At this point we have all the elements to bound \( \| \omega_{n+1} \| \). To simplify the final expression, we use the following large bound:
\[
\| y_{n-1} \| , 1 + \| y_{n-1} \| , \| y_n \| , 1 + \| y_n \| , \| F_n v_n \| \leq \pi_n.
\]
From (44) and (28),
\[
\left\| \begin{array}{c} \tilde{A}_n \\ \tilde{s}_n^T \end{array} \right\| \eta_n \leq (m \varphi_n + 1) \pi_n \epsilon \| A_{n+1} \|.
\]
From (27) and (51),
\[
\| C_{n+1} \theta_n \| \leq n \pi_n \beta_{n+1} \epsilon \| A_{n+1} \|.
\]
From (55),
\[
\| \Psi_n \| \leq \varphi_n \nu_n \pi_n \epsilon \| A_n \|.
\]
From (56), (48), and (19),
\[
| \tau_{n+1}^T \tilde{v}_n | + | \xi_n | \leq \epsilon \left( | z_{n+1} | + (m + n + 1) | s_n^T | | F_n | | v_n | + (m + 1) | c_{n+1}^T | | v_n | \right).
\]
The bounds
\[
| z_{n+1} | \leq \pi_n \| A_{n+1} \|, \quad | s_n^T | | F_n | | v_n | \leq \varphi_n \pi_n \| A_{n+1} \|,
\]
and
\[
| c_{n+1}^T | | v_n | \leq \| C_{n+1} \| \| D_{n+1} \| (1 + \| y_{n-1} \|) \leq \beta_{n+1} \pi_n \| A_{n+1} \|
\]
imply that
\[
\left\| \begin{bmatrix}
-\Psi_n \\
\tau_{n+1}^T \widetilde{v}_n + \xi_n
\end{bmatrix} \right\| \leq \pi_n \epsilon \|A_{n+1}\| \max \{\varphi_n \nu_n, 1 + (m + n + 1) \varphi_n + (m + 1) \beta_{n+1}\}.
\]
Assuming that \( m + 1 < n \), (58) implies that
\[
\|\omega_{n+1}\| \leq (\varphi_n \nu_n + \delta_n) \pi_n \epsilon \|A_{n+1}\|, \quad \text{where} \quad \delta_n = 2n (\varphi_n + \beta_{n+1}).
\]
Finally, from (41), \( \gamma_{n+1} = \gamma_n + \varphi_n \nu_n + \delta_n \).

We have thus obtained two recursive relations for the coefficient of the perturbation bound: from Theorem 2,
\[
\nu_n = \gamma_n + \nu_{n-1} + 1,
\]
and from Theorem 3,
\[
\gamma_{n+1} = \gamma_n + \varphi_n \nu_n + \delta_n.
\]
Combining these two relations and setting \( \delta_0 = 1 \), we see that \( \nu_n \) can be upper bounded by an expression of the form \( \sum_{k=1}^{n-1} c_k \delta_k \), where \( c_k \) is, up to a constant, the product of \( 1 + \varphi_h \) with \( h > k \). Hence \( \nu_n \) can grow as the product of all the quantities \( 1 + \varphi_k \) with \( 1 \leq k \leq n \). If these quantities are small, we expect an exponential growth with a base slightly larger than 1.

If we change \( C_N \) and \( D_N \) by applying technique (15), the matrix \( A_N \) does not change, while the coefficient \( \nu_n \) in bound (36) changes because of the variation of \( \beta_{n+1} \) and \( \varphi_n \). The experiments presented in section 6 will show the effects on the stability of different representations of the same matrix \( A_N \).

5.5. Stability of GLev. In Theorem 3 we have shown that the hypothesis (35) of Theorem 2 is satisfied by GLev. Combining the estimates of the errors found in these theorems, we can conclude about the stability of GLev.

**Theorem 4.** The solution \( \bar{x}_n \) of system (2) for \( n = 1, \ldots, N \), computed by using GLev, satisfies
\[
(A_n + I_n + \Delta A_n) \bar{x}_n = b_n
\]
with
\[
\|I_n + \Delta A_n\| \leq (n \beta_n + \nu_n \pi_{n-1}) \epsilon \|A_n\| + O(\epsilon^2),
\]
where \( \beta_n \) is defined in Theorem 1, \( \pi_n \) is defined in (30), and \( \nu_n \) satisfies (60) and (61).

The first component of the error is negligible with respect to the second one, pointing out that the error arising in the computation of the elements of the matrix \( A_N \) affects only marginally the stability of GLev.

5.6. Comparison with Gaussian elimination. The outer part of GLev is based on the factorization of \( A_n \) into a block lower triangular factor by a block upper triangular factor. We now show that this factorization is related to the standard factorization of a matrix into the product of two point triangular factors.

**Theorem 5.** Consider the sequences of matrices of increasing size defined by
\[
Q_1 = [1], \quad Q_{n+1} = \begin{bmatrix} Q_n & 0_n \\ 0_n^T & 1 \end{bmatrix} Q_{n+1} \quad \text{for} \quad n = 1, \ldots, N-1,
\]
(62) \[ P_n = A_n Q_n^{-1} \quad \text{for} \quad n = 1, \ldots, N. \]

Then \( Q_n \) is a unit upper triangular matrix, and \( P_n \) is a lower triangular matrix with 
\( z_1, \ldots, z_n \) on the diagonal.

**Proof.** The proof is by induction on \( n \). The assertion is obviously true for \( n = 1 \).

For \( n > 1 \), (4) implies that

\[
P_n = A_n Q_n^{-1} = P_n \begin{bmatrix} Q_{n-1}^{-1} & 0_{n-1} \\ 0_{n-1}^T & 1 \end{bmatrix} = \begin{bmatrix} A_{n-1} & 0_{n-1} \\ s_{n-1}^T & z_n \end{bmatrix} \begin{bmatrix} Q_{n-1}^{-1} & 0_{n-1} \\ 0_{n-1}^T & 1 \end{bmatrix}
\]

Matrix \( P_n \) is lower triangular because of the induction hypothesis. \( \square \)

(62) implies that

\[ A_n = P_n Q_n. \]

Hence \( P_n Q_n \) is the point LU factorization of \( A_n \) into the product of a lower triangular 
factor by a unit upper triangular factor. The corresponding method is Gaussian 
elimination by rows (in the following denoted by GEr) applied without pivoting, i.e., 
with \( z_1, \ldots, z_n \) acting as pivots.

The solution \( c \tilde{x}_n \) of the system having \( A_n \) as a coefficient matrix and \( b_n \) as 
a right-hand side, computed by GEr, satisfies

\[ (A_n + \Delta A_n) c \tilde{x}_n = b_n, \quad \text{where} \quad \|\Delta A_n\| \leq \mu_n \epsilon \|P_n\| \|Q_n\| + O(\epsilon^2), \]

where \( \mu_n \) is a linear function of \( n \).

To show that (64) holds, we note that applying Gaussian elimination by rows to 
the matrix \( A_n \) (and getting \( A_n = P_n Q_n \)) is equivalent to applying standard Gaussian 
elimation by columns to the matrix \( A_n^T \) (and getting \( A_n^T = L_n U_n \)). From the 
uniqueness of the LU decomposition, it follows that \( P_n = U_n^T \) and \( Q_n = L_n^T \). Referring to the 
roundoff analysis of Gaussian elimination in [8] yields that the computed triangular 
factors \( L_n \) and \( U_n \) satisfy

\[ \tilde{L}_n \tilde{U}_n = A_n^T + H_n, \quad \text{where} \quad |H_n| \leq \mu'_n \epsilon (|A_n^T| + \|L_n\| \|U_n\|) + O(\epsilon^2), \]

\( \mu'_n \) being a linear function of \( n \). Since \( \tilde{P}_n \tilde{Q}_n = A_n + H_n^T \), relation (64) follows from the 
classical backward error analysis (see, for example, [8], sect. 3.3) under the reasonable 
assumption that \( \|\tilde{P}_n\| \|\tilde{Q}_n\| \sim \|P_n\| \|Q_n\| \).

Hence the stability of GEr depends on how much the elements grow during the 
computation of the factorization (63). Since GEr is applied with no pivoting, \( \|P_n\| \) 
and \( \|Q_n\| \) can become arbitrarily large with \( \|P_n\| \) bounded by \( \|A_n\| \) multiplied by 
the growth factor and, in the worst case,

\[ \|Q_n\| \leq \|Q_{n-1}\| \|Q_n\| \leq \|Q_{n-1}\| \|Q_{n-1}\| (1 + \|y_{n-1}\|) \leq \pi_{n-1}. \]

Not much more can be said about stability if \( A_N \) does not enjoy special properties 
that can be exploited to simplify the bounds we have found. An important class of 
matrices, for which GEr without pivoting is stable, is the class of diagonally 
dominant matrices (see [12], sect. 9.4). The following theorem, stated for general matrices, 
applies also to computed matrices.
Theorem 6. If $A_N$ has diagonal dominance, then

(a) $\|y_n\| \leq 1$,  
(b) $\|Q_n\| \leq 2$,  
(c) $\|Q_n^{-1}\| \leq n$,  
(d) $\|P_n\| \leq n \|A_n\|

for any $n$.

Proof. $A_n^T$ has diagonal dominance by columns for any $n$. For bound (a), see Prob. 12.5 of [12], applied to $A_n^T$ partitioned as in (3). For bound (b), see sect. 12.3.1 of [12] since (63) implies that

$$A_n^T = Q_n^T P_n^T$$

i.e., $Q_n^T$ is the unit lower triangular factor of the LU factorization of $A_n^T$. Bound (c) follows from

$Q_n^{-1} = \begin{pmatrix} 1 & y_{1,1} & \cdots & y_{n-1,1} \\ & 1 & \vdots & \\ & & \ddots & y_{n-1,n-1} \\ & & & 1 \end{pmatrix}$

and bound (d) follows from $\|P_n\| \leq \|A_n\| \|Q_n^{-1}\|$. \qed

An a priori control of the diagonal dominance of $A_N$ would be too expensive. Hence it is natural to turn our attention to quantities which can be monitored simply during the computation, that is, the sequences of the vectors $\tilde{y}_n$ and of the Schur complements $\tilde{z}_n$.

In our case, the backward error analysis which leads to (64) should be carried out on $A_N$, and (64) should be replaced by

$$\begin{pmatrix} \tilde{A}_n + \Delta A_n \end{pmatrix} \tilde{\mathbf{x}}_n = \tilde{b}_n, \quad \text{where} \quad \|\Delta A_n\| \leq \mu_n \epsilon \|P_n\| \|Q_n\| + O(\epsilon^2).$$

The stability results (65) for GEr and (36) for GLev, even if pessimistic, suggest that the stability of the two methods may, in certain cases, be comparable. We investigate this aspect in the numerical experiments.

6. Numerical experiments. The experiments have been conducted on an Intel Core Duo at 3 GHz, 2GB RAM, using double precision arithmetic. Two sets of numerical experiments are performed, for different values of the size $N$, in order to investigate two different sources of instability: (i) the magnitude of $y_n$, related to the smallness of $z_n$ and to the ill-conditioning of $A_n$, and (ii) the magnitude of $\beta_n$ and $\varphi_n$. In the first set, the entries of $C_N$ and $D_N$ belong to the interval $[-1, 1]$, while in the second set, a few entries of large magnitude are allowed. The left-hand side vector $\tilde{b}_N$ is computed from an exact solution, randomly generated.

We must note that the stability analysis of the outer part assumes that $\tilde{y}_{n-1}$ and $\tilde{z}_n$ are computed with an error suitably bounded. Comparing the errors produced by GEr and GLev does not allow splitting the errors due to the inner part from the errors due to the outer part. For this reason, in addition to GEr, we consider an ideal hybrid method, called $\mathcal{PQ}$, which employs the outer scheme of GLev but uses exact values of $y_{n-1}$ and $z_n$ (i.e., computed with high precision).

(i) The matrices for the first set of experiments have been generated for different values of the displacement rank. The first $m - 1$ columns of both $C_N$ and $D_N$ are randomly generated with uniform distribution between $-1$ and 1. The column $d_N^{(m)}$
is the first canonical vector, and the column $c_{(m)}^N$ is the first canonical vector multiplied by a value $\lambda \geq 1$ which influences the diagonal dominance of $A_N$. Finally, $C_N$ is multiplied by $1/\lambda$. In this way matrices $A_N$ with different dominance properties and different conditioning histories $H = \{\kappa(A_n), n = 1, \ldots, N\}$ are obtained. Consequently, problems with different values of $\|y_n\|$ for $n = 1, \ldots, N - 1$ are dealt with. For each problem the geometrical mean $Y = \pi^N_{N-1}$ is considered, where $\pi_{N-1}$ is defined in (30). In the experiments $Y$ varies from 1 to 10. As already noted, if matrix $A_N$ is diagonally dominant, then $Y \leq 2$. The theoretical results of section 5 suggest that $Y$ is a relevant tool to analyze the stability behavior of GLev. Hence $Y$ is taken as the independent variable in the following figures.

Denoting by

$$R_n = \frac{\|b_n - A_n\tilde{x}_n\|}{\|	ilde{x}_n\|}$$

the relative residual at size $n$, each problem is solved by applying the three methods, and for each method, the last relative residual $R = R_N$ is found. Varying $b_N$, the largest $R$ can be taken as an experimental estimate of the perturbation $\|\Delta A_N\|$ such that $(A_N + \Delta A_N)\tilde{x}_N = b_N$. Of course, $R$ is expected to have a better behavior than the bound appearing in Theorem 4. Figure 1, which refers to a set of 400 test problems with $m = 4$ and $N = 2^{10}$, shows for each method the log-log plot of $R$ as a function of $Y$. (No substantial difference appears in the plots for other values of $m$ and $N$.) The gray level of the points characterizes the method.

![Fig. 1. Relative residuals $R$ as functions of $Y$: the black points refer to the GLev results, the middle gray points to the GEr results, and the light gray points to the PQr results.](image)

We see that for small values of $Y$, the three methods share the same good behavior. The performance of the methods worsens for increasing $Y$. The loss of precision for GLev is greater than for GEr and PQr. The fact that the residuals of GEr and PQr are almost indistinguishable suggests that the reduction of the GLev performance is
mainly due to the inner part. In any case, we can say that, on average, GLev appears less stable than GEr, especially when $Y$ is large, i.e., when the matrix $A_N$ is not diagonally dominant.

Limiting our analysis to the GLev residuals, we see that the points gather in a cone in the log-log scale, suggesting a polynomial growth of the residuals with respect to $Y$.

To investigate how the GLev relative residuals $R_n$ depend on $\|y_n\|$ varying $n$, we examine in detail one of the previously considered problems. Figure 2 shows the tight relation between the behavior of $R_n$ (black points), the behavior of $\|y_n\|$ (light gray points), and the conditioning history $\mathcal{H}$ (middle gray points) for a typical test problem with $n$ increasing from 1 to $2^{10}$. To put the graphs in the same figure, the norms $\|y_n\|$ have been divided by $10^9$, and the condition numbers $\kappa(A_n)$ have been divided by $10^{10}$. Comparing the three graphs, we see that after an initial phase where $R_n$, $\|y_n\|$, and $\kappa(A_n)$ increase in a regular slow way, at the 291st step there is a sudden increase of $R_n$ due to an equally sudden increase of $\|y_n\|$ from 2.7 to 64.4 and of $\kappa(A_n)$ from 3896 to 87014. The same phenomenon occurs again on and off in subsequent steps.

(ii) For the second set of experiments, we generate matrices $C_N$ and $D_N$ as in the previous case except for the fact that different couples of generators corresponding to the same matrix $A_N$ are obtained by allowing the columns of $C_N$ and $D_N$ to depend on a new parameter. In this way, very different values of the stability parameters $\beta_n$ and $\varphi_n$ occur. First we generate matrices $C_N$ and $D_N$ with small $\beta_n$ and different $\varphi_n$ magnitudes. Figure 3 shows the relative residuals obtained by applying GLev to a set of test problems with $m=5$ and $N=2^{10}$. Only the matrices with $\varphi < 10^2$ or $\varphi > 10^4$, where $\varphi = \max\{\varphi_1, \ldots, \varphi_{n-1}\}$, are selected. The light grey points refer to $\varphi < 10^2$ and the black points to $\varphi > 10^4$. The dependence on $\varphi$ appears clearly, confirming the theoretical bound of Theorem 4.

---

**Fig. 2.** Log plot of the relative residuals $R_n$ (black points), $\|y_n\|/10^9$ (light gray points), and $\kappa(A_n)/10^{10}$ (middle gray points) as functions of $n$. 
Then we generate examples with large values of $\beta_n$. Figure 4 refers to a set of 400 test problems with $m = 5$ and $N = 2^{10}$, each one represented by three different couples of $C_N$ and $D_N$, for which the quantity $\delta = \max\{\delta_1, \ldots, \delta_{N-1}\}$, where $\delta_n$
is defined in (59), has values of order $10^2$, $10^5$, and $10^8$. The corresponding relative residuals are indicated by light gray points in the first case, by middle gray points in the second case, and by black points in the third case. It is evident that the relative residuals grow with the quantity $\delta$, in accord with the bound given in Theorem 4.

To test the effectiveness of the technique (15) described at the end of section 3, the couples of $C_N$ and $D_N$, related to quantities $\delta$ of order $10^5$ and $10^8$ in the previous experiment, have been modified by applying (15), replacing large generators with smaller ones. (The computation has been carried out in high precision arithmetic.) Then GLev has been applied to solve the corresponding test problems, and the relative residuals (black points in Figure 5) have been compared with those obtained in the case with $\delta \sim 10^2$ (light gray points in Figures 4 and 5). It is evident that the technique is effective in reducing the residuals. In fact, the quantities $\delta$, initially of order $10^5$ and $10^8$, have been reduced to order $10^1$.

7. Conclusions. The backward error analysis of the generalized Levinson algorithm has produced an upper bound on the perturbation matrix and hence on the norm of the relative residuals. Numerical experiments have confirmed that $R_n$ and $\|y_n\|$, regarded as functions of $n$, are strictly related. They have also shown that bound (36) is too pessimistic, especially for large values of $\|y_n\|$. In fact, the residuals in actually computed problems are significantly smaller than the quantity indicated by the bound. The empirical comparison with Gaussian elimination by rows has shown that the residuals are of the same order of magnitude when the norms $\|y_n\|$ are “sufficiently” small for any $n$. Diagonally dominant matrices satisfy this condition. Moreover, leaving aside the magnitude of $\|y_n\|$, the major source of numerical instability for the generalized Levinson algorithm appears to lie in the inner part, which is highly conditioned by the magnitude of the entries of $C_N$ and $D_N$. The technique suggested in [9] to overcome this kind of problems has been considered and successfully tested.
REFERENCES