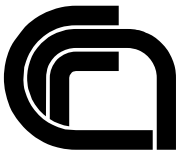

Consiglio Nazionale delle Ricerche

A divide and conquer algorithm for Toeplitz-like systems: stability analysis for the symmetric case

P.Favati, G.Lotti, O.Menchi

IIT TR-15/2014

Technical report

Novembre 2014

Istituto di Informatica e Telematica

A divide and conquer algorithm for Toeplitz-like systems: stability analysis for the symmetric case

P. Favati G. Lotti O. Menchi

Abstract

A superfast algorithm for the solution of Toeplitz-like systems has been introduced and tested in [4]. In this paper a theoretical error analysis of the algorithm is performed for the symmetric case. The analysis produces an upper bound to the norm of the residual vector, allowing the detection of some parameters which rule the stability behavior of the algorithm. These parameters take into account both the conditioning properties of the coefficient matrices at the different levels of recursion and the magnitude of some involved matrices measured through their generators. The experimentation confirms the theoretical results, pointing out that, in general, the upper bound to the norm of the residual vector is too pessimistic.

1 Introduction

In this paper the problem of solving the linear system

$$A\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where A is an $N \times N$ nonsingular symmetric Toeplitz-like matrix and $\boldsymbol{b}$ is a vector of size N, is addressed. The class of Toeplitz-like matrices is based on the concept of displacement rank introduced in [11] and studied by many authors (see for example [7, 9, 10]). The displacement operator allows a compact representation of the matrices of this class by means of a set of generators. For the Toeplitz-like matrices, fast and superfast algorithms have been devised (see the extensive bibliography in [10]), but the question of their stability is still matter of discussion. In [4] a new superfast algorithm, called `solvegen`, has been proposed. It is based on a divide and conquer strategy which combines the solutions of two half size Toeplitz-like systems with enlarged right-hand side increased by a constant number (proportional to the displacement rank of the given matrix) of columns at each recursion step. `solvegen` avoids the explicit inversion the matrix, which is often cause of instability. In fact, the experimentation shows that it is more stable than the superfast algorithm of [2, 12] which has $O(N \log^2 N)$ complexity. This stability improvement is obtained at the expense of an increase of the complexity to $O(N \log^3 N)$ floating point operations.

We are interested in carrying out a theoretical error analysis of `solvegen` with the aim of characterizing the parameters which rule its stability behavior. Considering the difficulty of performing such an analysis in the general case, we study the stability properties of `solvegen` rewritten for the symmetric case.

Some preliminaries on Toeplitz-like matrices are included in Section 2. In Section 3, after a first version of the proposed algorithm which does not yet make use of the displacement structure, the explicit expressions for the generators of the involved matrices and the version of the algorithm which fully exploits the displacement properties are given. The stability analysis of `solvegen` is discussed in Section 4. Finally, in Section 5 the results of the numerical experiments are shown, confirming the relevance of the detected parameters in the numerical stability of the algorithm.

2 Toeplitz-like matrices

The definition of Toeplitz-like structure is based on the concept of displacement rank, which measures how close a matrix is to a Toeplitz matrix. Given an $n \times n$ matrix A, we consider the *displacement operator*

$$\nabla(A) = A - ZAZ^T, \tag{2}$$

where Z is the $n \times n$ *down-shift* matrix

$$Z = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}.$$

The matrix A is said to be *Toeplitz-like* if the quantity $r_{\text{disp}}(A) = \operatorname{rank} \nabla(A)$ (called *displacement rank*) is small with respect to n (more formally $r_{\text{disp}}(A) = O(1)$ for $n \to \infty$). The set of Toeplitz-like matrices, unlike the set of Toeplitz matrices, is closed under the operations of multiplication and inversion. Let $r_{\text{disp}}(A) = \rho$, then

$$\nabla(A) = C \Sigma D^T, \tag{3}$$

for suitable $n \times \rho$ matrices C and D and a signature $\rho \times \rho$ matrix Σ, i.e. a diagonal matrix with diagonal elements σ_i either 1 or -1. The matrices C, Σ and D are called *generators* of A. Denoting by $\boldsymbol{c}_i$ and $\boldsymbol{d}_i$ the columns of C and D respectively, then

$$\nabla(A) = \sum_{i=1}^{\rho} \sigma_i \boldsymbol{c}_i \boldsymbol{d}_i^T. \tag{4}$$

For example, the representation of a Toeplitz matrix A of elements $a_{i,j}$ with $a_{1,1} \neq 0$ is

$$\nabla(A) = \boldsymbol{c}_1 \, \boldsymbol{e}_1^T + \boldsymbol{e}_1 \, \boldsymbol{d}_2^T, \quad \text{with} \quad \boldsymbol{c}_1 = A \, \boldsymbol{e}_1, \quad \boldsymbol{d}_2 = A^T \boldsymbol{e}_1 - a_{11} \boldsymbol{e}_1, \tag{5}$$

i.e. $\boldsymbol{c}_1$ is the first column of A and $\boldsymbol{d}_2^T$ is the first row of A without the first component. Hence, the displacement rank of a Toeplitz matrix is $\rho = 2$, except in some special case where $\rho = 1$.

From (2) and (3) it follows that

$$\nabla(A^T) = \nabla(A)^T = D\Sigma\, C^T, \tag{6}$$

hence the generators of A^T are obtained by swapping the generators of A. In the case of a symmetric matrix A, we have $D = C$ and

$$\nabla(A) = C\Sigma\, C^T. \tag{7}$$

For example, if the Toeplitz matrix whose displacement is given in (5) is symmetric and $\alpha = a_{1,1} \neq 0$, denoting $\beta = \mathrm{sgn}(\alpha)/\sqrt{|\alpha|}$, formula (7) holds with

$$C = \beta\,[\boldsymbol{c}_1, \boldsymbol{c}_2], \quad \Sigma = \mathrm{diag}\big(\mathrm{sgn}(\alpha),\, -\mathrm{sgn}(\alpha)\big), \quad \text{where} \quad \boldsymbol{c}_2 = \boldsymbol{c}_1 - \alpha\,\boldsymbol{e}_1.$$

The generators enable us to represent a Toeplitz-like matrix as the sum of products of lower and upper triangular Toeplitz factors. Denoting by $L(\boldsymbol{s})$ the lower triangular Toeplitz matrix whose first column is $\boldsymbol{s}$ and by $L^T(\boldsymbol{s})$ the upper triangular Toeplitz matrix whose first row is $\boldsymbol{s}$, then

$$A = \sum_{i=1}^{\rho} \sigma_i\, L(\boldsymbol{c}_i)\, L^T(\boldsymbol{d}_i), \tag{8}$$

with $\boldsymbol{d}_i = \boldsymbol{c}_i$ in the symmetric case. In this paper we assume that the coefficient matrix A of (1) is not explicitly given but only represented through its generators.

Formula (8) can be exploited to compute the product of a Toeplitz-like matrix A by a vector $\boldsymbol{v}$. In fact

$$A\boldsymbol{v} = \sum_{i=1}^{\rho} \sigma_i\, L(\boldsymbol{c}_i)\, L^T(\boldsymbol{d}_i)\, \boldsymbol{v},$$

and the product is obtained by multiplying first upper and then lower triangular Toeplitz matrices by vectors. If n is large, it is worthwhile to compute these products embedding the triangular Toeplitz matrices into circulant matrices and using FFT. In this way the product $A\boldsymbol{v}$ has a computational cost of $O(n \log n)$ for $n \to \infty$ (see [5] for the sketch of a function `prod` that can be used to multiply Toeplitz-like matrices by vectors).

When a new method for solving problem (1) is proposed, it is a good practice to look for conditions that ensure the stability of the method. If the elements of A are given through the generators, testing these conditions may require the explicit construction of some or all the elements of A by multiplying A with vectors of the canonical basis. For example, it is easy to verify if A is symmetric,

but if we want to test the diagonal dominance we have to compute the elements of A which in the symmetric case are given by

$$a_{k,j} = a_{j,k} = \sum_{i=1}^{\rho} \sigma_i \sum_{h=1}^{j} c_{h,i}\, c_{k-j+h,i}, \quad \text{for} \quad j \le k.$$

The decomposition (3) of $\nabla(A)$, and consequently the representation of A by means of the generators, is not unique. An important representation is the *orthogonal* one, obtained by computing the SVD decomposition $\nabla(A) = UWV^T$, where W is the $\rho \times \rho$ diagonal matrix of the nonzero singular values $w_1 \ge \ldots \ge w_\rho > 0$ and U and V are $n \times \rho$ matrices with orthogonal columns. If A is not symmetric, (3) is given by

$$\nabla(A) = \widehat{C}\,\widehat{\Sigma}\,\widehat{D}^T, \quad \text{where} \quad \widehat{C} = UW^{1/2}, \quad \widehat{\Sigma} = I, \quad \widehat{D} = VW^{1/2}.$$

If A is symmetric, let $\widehat{\Sigma}$ be the signature matrix such that $V = U\,\widehat{\Sigma}$. Then (3) is given by

$$\nabla(A) = \widehat{C}\,\widehat{\Sigma}\,\widehat{D}^T, \quad \text{where} \quad \widehat{C} = \widehat{D} = UW^{1/2}.$$

In [3, 6] the stability of a method for solving a linear system, whose Toeplitz-like matrix A is not explicitly given but only represented through its generators, is recognized depending on how large the generators are with respect to the magnitude of A. So, when stability is analyzed, we suggest to consider the function

$$\psi(A) = \sum_{i=1}^{\rho} \|\boldsymbol{c}_i\|_1 \, \|\boldsymbol{d}_i\|_1, \tag{9}$$

which verifies $\|\nabla(A)\|_1 \le \psi(A)$. Since for any vector $\boldsymbol{s}$ it is $\|L(\boldsymbol{s})\|_1 = \|\boldsymbol{s}\|_1$ then from (8) it follows that

$$\|A\|_1 \le \psi(A). \tag{10}$$

Orthogonal generators have in general smaller 1-norms; if they are used for evaluating the function $\psi(A)$, a more accurate measure of $\|A\|_1$ is often obtained. For this reason, we will resort to the replacing of the generators with their orthogonal counterparts if the latter ones give a smaller value of $\psi(A)$. This replacement does not contribute to the global computational cost, since the cost of obtaining the orthogonal representation from a generic one is $O(n\rho^2)$ for $n \to \infty$, lower than the cost of the matrix by vector product.

Notation: A matrix is denoted by an upper-case letter, possibly followed by indices, and its columns are denoted by the corresponding lower-case and bold-face letter, followed by the index of the matrix and the index of the column. For example, $\boldsymbol{c}_{F,i}$ is the ith column of the matrix C_F and $\boldsymbol{d}_{12,i}$ is the ith column of the matrix D_{12}. Moreover, to simplify the list of the input parameters of the programs, the notation $\{A\}$ will be used to indicate the set $\{n, \rho, C, \Sigma, D\}$. For simplicity the subscript 1 is dropped from the 1-norm for both vectors and matrices.

3 The algorithm

As already said, the coefficient matrix A in (1) is assumed to be a nonsingular symmetric Toeplitz-like matrix, not given explicitly but only through its representation, which we assume to be orthogonal. In [4] a divide and conquer superfast algorithm, called `solvegen`, has been described for solving system (1) in the general Toeplitz-like (i.e. possibly nonsymmetric) case and we rewrite here the algorithm for the symmetric case.

We assume $N = 2^p\, n_e$, where n_e represents the size of the systems (called *elementary*) to be solved at the last level of the recursion, with the restriction that $n_e \geq r_{\rm disp}(A) = \rho$. At each recursion level, the algorithm combines the solutions of two half size Toeplitz-like systems with enlarged right-hand side and makes use of the displacement properties of the involved matrices.

Before writing the code of `solvegen` it is advisable to sketch the code implementing the divide and conquer strategy in terms of matrices.

3.1 The divide and conquer strategy

Initially we set $A^{(0)} = A$ and $B^{(0)} = \boldsymbol{b}$. At the kth level of recursion, $k = 0, \ldots, p$, there are 2^k symmetric systems of size $n_k = 2^{p-k} n_e$ to be solved. Let

$$A^{(k)} X^{(k)} = B^{(k)} \tag{11}$$

be one of these systems. For $k \leq p-1$, let

$$A^{(k)} = \begin{bmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ A_{21}^{(k)} & A_{22}^{(k)} \end{bmatrix}, \quad X^{(k)} = \begin{bmatrix} \overline{X}^{(k)} \\ \underline{X}^{(k)} \end{bmatrix}, \quad B^{(k)} = \begin{bmatrix} \overline{B}^{(k)} \\ \underline{B}^{(k)} \end{bmatrix}, \tag{12}$$

with $A_{11}^{(k)}$ and $A_{22}^{(k)}$ symmetric and $A_{21}^{(k)} = A_{12}^{(k)T}$. The solution of (11) can be found by computing

$$\begin{aligned} &\left[U\,, F\right] = A_{11}^{(k)-1} \left[\overline{B}^{(k)}\,, A_{12}^{(k)}\right], \\ &Y = \underline{B}^{(k)} - A_{21}^{(k)} U, \quad S = A_{22}^{(k)} - A_{21}^{(k)} F, \\ &\underline{X}^{(k)} = S^{-1} Y, \quad \overline{X}^{(k)} = U - F \underline{X}^{(k)}, \end{aligned} \tag{13}$$

i.e. by solving two subsystems, the first one with matrix $A_{11}^{(k)}$ and an increased number of right-hand sides, the second one with the Schur complement S of $A_{11}^{(k)}$ without increase of right-hand sides.

The following recursive routine `solvemat` shows how the computation can be performed. The first time the function is applied with $k = 0$ to matrix $A^{(0)}$ and right-hand side $B^{(0)}$. Subsequent calls are made to matrices of halved dimension, until the preassigned size n_e is reached. At the kth level of recursion, with $k = 0, \ldots, p$, we assume that all the 2^k subsystems of the form (11) have a nonsingular coefficient matrix $A^{(k)}$. At the last level of recursion there are 2^p

elementary systems with matrices of size n_e. These systems are solved directly, by means of a routine `elem` (implementing, for example, Gauss method). The initial partition of the matrix $A^{(k)}$ and of the right-hand side $B^{(k)}$ is performed according to (12) (in the following code the colon notation is used).

`function` $X^{(k)} =$ `solvemat` $\left(n_k, A^{(k)}, B^{(k)}\right)$

% computes recursively the solution of the system $A^{(k)}\, X = B^{(k)}$ of size n_k

`if` $k = p, \quad X^{(k)} =$ `elem` $(A^{(k)}, B^{(k)})$

`else`

% partition

$n = n_k; \quad r_1 = (1 : n/2); \quad r_2 = (n/2 + 1 : n);$

$A_{11} = A^{(k)}(r_1, r_1); \quad A_{12} = A^{(k)}(r_1, r_2);$

$A_{21} = A^{(k)}(r_2, r_1); \quad A_{22} = A^{(k)}(r_2, r_2);$

$\overline{B} = B^{(k)}(r_1, :); \quad \underline{B} = B^{(k)}(r_2, :);$

% recursion

$[\, U, F\,] =$ `solvemat` $\left(n/2,\, A_{11}, \left[\overline{B}\,, A_{12}\right]\right);$

$Y = \underline{B} - A_{21}U; \quad S = A_{22} - A_{21}F;$

$\underline{X} =$ `solvemat` $\left(n/2,\, S\,, Y\right); \quad \overline{X} = U - F\underline{X};$

$$X^{(k)} = \left[\begin{array}{c} \overline{X} \\ \underline{X} \end{array} \right];$$

`end`

It is easy to recognize that the divide and conquer strategy used for `solvemat` leads to a block $\mathcal{LU}$ factorization of A. In general, an excessive growth of the elements of F can be an important factor for the instability of the method. If the matrix A has some special structural properties, then all the matrices F arising at the different levels can be nicely bounded. For example, if A is positive definite, then $A^{(k)}$ is positive definite for any k and $\|F\|_2 \leq \sqrt{\kappa_2(A^{(k)})}$, where $\kappa_2(A^{(k)})$ is the condition number of $A^{(k)}$. If $A^{(k)}$ has diagonal dominance, then $\|F\| \leq 1$.

3.2 Generators of the matrices used in the recursion

We give now the representation, in terms of generators, of the matrices involved in the computation. It is important to have at any recursion level the representation with the minimal number of generators. When writing the generators of one of the matrices required at recursion level k, for simplicity of notation the level superscript k is omitted, so

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right],$$

with A_{11} and A_{22} symmetric and $A_{21} = A_{12}^T$. Let C and Σ satisfy (7) and partition C compatibly

$$C = \begin{bmatrix} \overline{C} \\ \underline{C} \end{bmatrix}.$$

For simplicity, we still denote by Z be the down-shift matrix of order $n/2$ and $\boldsymbol{e}_i$ the ith canonical vector of length $n/2$.

The following scalar and vectors are computed by means of (8)

$$\begin{aligned} \alpha = \boldsymbol{e}_{n/2}^T A_{11} \boldsymbol{e}_{n/2} = \sum_{i=1}^{\rho} \sigma_i \overline{\boldsymbol{c}}_i^T \overline{\boldsymbol{c}}_i, \quad \boldsymbol{v}_1 = Z A_{11} \boldsymbol{e}_{n/2} = Z \sum_{i=1}^{\rho} \sigma_i L(\overline{\boldsymbol{c}}_i) J \overline{\boldsymbol{c}}_i, \\ \boldsymbol{v}_2 = Z A_{21} \boldsymbol{e}_{n/2} = Z \sum_{i=1}^{\rho} \sigma_i L(\underline{\boldsymbol{c}}_i) J \overline{\boldsymbol{c}}_i, \end{aligned} \tag{14}$$

where J is the reversal matrix.

- The displacements of the blocks are

$$\begin{aligned} \nabla(A_{11}) &= \overline{C}\, \Sigma\, \overline{C}^T, \\ \nabla(A_{12}) &= \overline{C} \Sigma \underline{C}^T + \boldsymbol{v}_1 \boldsymbol{e}_1^T, \\ \nabla(A_{21}) &= \nabla(A_{12})^T = \underline{C} \Sigma \overline{C}^T + \boldsymbol{e}_1 \boldsymbol{v}_1^T, \\ \nabla(A_{22}) &= \underline{C} \Sigma \underline{C}^T + \boldsymbol{v}_3\, \boldsymbol{e}_1^T + \boldsymbol{e}_1 \boldsymbol{v}_2^T, \quad \boldsymbol{v}_3 = \alpha \boldsymbol{e}_1 + \boldsymbol{v}_2. \end{aligned} \tag{15}$$

Then

$$\nabla(A_{21}) = C_{21} \Sigma_{21} D_{21}^T, \text{ where } C_{21} = \big[\underline{C}, \boldsymbol{e}_1\big],\ D_{21} = \big[\overline{C}, \boldsymbol{v}_1\big],\ \Sigma_{21} = \begin{bmatrix} \Sigma & \\ & 1 \end{bmatrix}.$$

In general

$$r_{\text{disp}}(A_{11}) = \rho, \quad r_{\text{disp}}(A_{12}) = r_{\text{disp}}(A_{21}) = \rho + 1, \quad r_{\text{disp}}(A_{22}) = \rho + 2.$$

- The matrix $F = A_{11}^{-1} A_{12}$ is not symmetric. Its displacement is

$$\nabla(F) = F - ZFZ^T = A_{11}^{-1}\big(A_{12} - A_{11} Z F Z^T\big).$$

Since

$$A_{12} - A_{11} Z F Z^T = \nabla(A_{12}) + Z A_{12} Z^T - \Big(\nabla(A_{11}) + Z A_{11} Z^T\Big) Z F Z^T,$$

using $Z^T Z = I - \boldsymbol{e}_{n/2} \boldsymbol{e}_{n/2}^T$, we have

$$\begin{aligned} Z A_{12} Z^T - Z A_{11} Z^T Z F Z^T &= Z A_{12} Z^T - Z A_{11} F Z^T + Z A_{11} \boldsymbol{e}_{n/2} \boldsymbol{e}_{n/2}^T F Z^T \\ &= \boldsymbol{v}_1 \boldsymbol{e}_{n/2}^T F Z^T, \end{aligned}$$

and

$$\nabla(F) = A_{11}^{-1}\big(\overline{C}\,\Sigma\,\underline{C}^T + \boldsymbol{v}_1\boldsymbol{e}_1^T - \overline{C}\,\Sigma\,\overline{C}^T ZFZ^T + \boldsymbol{v}_1\boldsymbol{e}_{n/2}^T FZ^T\big).$$

Then

$$\nabla(F) = C_+\,\Sigma\,D_+^T + \boldsymbol{v}_+\,\boldsymbol{f}^T,$$

where

$$C_+ = A_{11}^{-1}\overline{C},\ D_+ = \underline{C} - ZA_{21}A_{11}^{-1}Z^T\overline{C},\ \boldsymbol{v}_+ = A_{11}^{-1}\boldsymbol{v}_1,\ \boldsymbol{f} = \boldsymbol{e}_1 + ZA_{21}A_{11}^{-1}\boldsymbol{e}_{n/2}.$$

Setting $K = \big[Z^T\overline{C}, -\boldsymbol{e}_{n/2}\big]$ and $G = A_{11}^{-1}K$, we have

$$\nabla(F) = C_F\,\Sigma_F\,D_F, \tag{16}$$

where

$$C_F = A_{11}^{-1}D_{21}, \quad D_F = C_{21} - ZA_{21}G, \quad \Sigma_F = \Sigma_{21}.$$

In general $r_{\rm disp}(F) = \rho + 1$.

• The Schur complement of A_{11} is $S = A_{22} - A_{21}F$. It is symmetric and its displacement is

$$\nabla(S) = \nabla(A_{22}) - \nabla(A_{21}F).$$

It is

$$\nabla(A_{22}) = \underline{C}\,\Sigma\,\underline{C}^T + \boldsymbol{v}_3\boldsymbol{e}_1^T + \boldsymbol{e}_1\boldsymbol{v}_2^T = \underline{C}\,\Sigma\,D_+^T + H_1, \tag{17}$$

where

$$H_1 = \underline{C}\,\Sigma\,\overline{C}^T ZFZ^T + \boldsymbol{v}_3\,\boldsymbol{e}_1^T + \boldsymbol{e}_1\boldsymbol{v}_2^T,$$

and

$$\nabla(A_{21}F) = A_{21}F - ZA_{21}FZ^T = A_{21}\nabla(F) + \big(A_{21}Z - ZA_{21}\big)FZ^T. \tag{18}$$

Since

$$A_{21}Z - ZA_{21} = A_{21}Z - ZA_{21}(Z^TZ + \boldsymbol{e}_{n/2}\boldsymbol{e}_{n/2}^T) = \nabla(A_{21})Z - \boldsymbol{v}_2\boldsymbol{e}_{n/2}^T, \tag{19}$$

we have

$$\nabla(A_{21}F) = A_{21}C_+\,\Sigma\,D_+^T + H_2, \tag{20}$$

where

$$H_2 = \nabla(A_{21})ZFZ^T + A_{21}\boldsymbol{v}_+\,\boldsymbol{f}^T - \boldsymbol{v}_2\boldsymbol{e}_{n/2}^T FZ^T.$$

Now

$$\begin{aligned} H_1 - H_2 &= \boldsymbol{v}_3\,\boldsymbol{e}_1^T + \boldsymbol{e}_1\boldsymbol{v}_2^T - \boldsymbol{e}_1\boldsymbol{v}_1^T ZFZ^T - A_{21}\boldsymbol{v}_+\,\boldsymbol{f}^T + \boldsymbol{v}_2\boldsymbol{e}_{n/2}^T FZ^T \\ &= \boldsymbol{g}\boldsymbol{f}^T, \quad \text{where} \quad \boldsymbol{g} = \boldsymbol{v}_3 - A_{21}\boldsymbol{v}_+. \end{aligned}$$

From (17) and (20) it follows that

$$\nabla(S) = \big(\underline{C} - A_{21}C_+\big)\,\Sigma\,D_+^T + \boldsymbol{g}\boldsymbol{f}^T = C_S\Sigma_S\,D_S^T,$$

where

$$C_S = \big[\underline{C}, \boldsymbol{v}_3\big] - A_{21}C_F, \quad \Sigma_S = \Sigma_F, \quad D_S = D_F. \tag{21}$$

Actually, the matrix $\underline{C} - A_{21}C_+$ has ρ columns but rank $\rho - 1$. In fact, if we multiply it by the vector $\Sigma\overline{C}^T\boldsymbol{e}_1$ we obtain the vector

$$\big(\underline{C} - A_{21}C_+\big)\Sigma\overline{C}^T\boldsymbol{e}_1 = \nabla(A_{21})\boldsymbol{e}_1 - \boldsymbol{e}_1\boldsymbol{v}_1^T\boldsymbol{e}_1 - A_{21}A_{11}^{-1}\nabla(A_{11})\boldsymbol{e}_1,$$

which is null because $\boldsymbol{v}_1^T\boldsymbol{e}_1 = 0$, $\nabla(A_{21})\boldsymbol{e}_1 = A_{21}\boldsymbol{e}_1$ and $\nabla(A_{11})\boldsymbol{e}_1 = A_{11}\boldsymbol{e}_1$. Then $r_{\rm disp}(S) = \rho$ and the representation (21) is not minimal. Moreover, we expect a symmetric representation for S, while the generators C_S and D_S given in (21) do not coincide. For this reason the decomposition (16) is replaced by its orthogonal decomposition $\nabla(S) = \widehat{C}_S\,\widehat{\Sigma}_S\,\widehat{D}_S$.

For what concerns the function ψ, since

$$|\alpha| \le \sum_{i=1}^{\rho}\|\overline{\boldsymbol{c}}_i\|^2, \quad \|\boldsymbol{v}_1\| \le \sum_{i=1}^{\rho}\|\overline{\boldsymbol{c}}_i\|^2, \quad \|\boldsymbol{v}_2\| \le \sum_{i=1}^{\rho}\|\underline{\boldsymbol{c}}_i\|\,\|\overline{\boldsymbol{c}}_i\|,$$

we have

$$\begin{aligned}
&\psi(A_{11}) \le \sum_{i=1}^{\rho}\|\overline{\boldsymbol{c}}_i\|^2 \le \sum_{i=1}^{\rho}\|\boldsymbol{c}_i\|^2 = \psi(A),\\
&\psi(A_{21}) = \psi(A_{12}) \le \sum_{i=1}^{\rho}\big(\|\overline{\boldsymbol{c}}_i\|\,\|\underline{\boldsymbol{c}}_i\| + \|\overline{\boldsymbol{c}}_i\|^2\big) \le \sum_{i=1}^{\rho}\|\overline{\boldsymbol{c}}_i\|\,\|\boldsymbol{c}_i\| \le \sum_{i=1}^{\rho}\|\boldsymbol{c}_i\|^2 = \psi(A),\\
&\psi(A_{22}) \le \sum_{i=1}^{\rho}\big(\|\underline{\boldsymbol{c}}_i\|^2 + 2\|\overline{\boldsymbol{c}}_i\|\,\|\underline{\boldsymbol{c}}_i\| + \|\overline{\boldsymbol{c}}_i\|^2\big) = \sum_{i=1}^{\rho}\|\boldsymbol{c}_i\|^2 \le \psi(A).
\end{aligned}$$

Analogously

$$\|\nabla(A_{21}) - \boldsymbol{v}_2\boldsymbol{e}_{n/2}^T\| \le \psi(A),$$

and from (18) and (19) it follows that

$$\nabla(S) = \nabla(A_{22}) - \nabla(A_{21}F) = \nabla(A_{22}) - A_{21}\nabla(F) - \big(\nabla(A_{21}) - \boldsymbol{v}_2\boldsymbol{e}_{n/2}^T\big)FZ^T,$$

then

$$\psi(S) \le \psi(A_{22}) + \|A_{21}\|\psi(F) + \psi(A)\|F\| \le 2\,\psi(A)\,\big(1 + \psi(F)\big). \tag{22}$$

3.3 The algorithm using the generators

We are now ready to rewrite the code of the function `solvemat` in terms of the generators. The following functions are required:

– `elemgen` to solve the elementary systems with symmetric matrices of size n_e.

– `replace` to replace the generators using the SVD.

– `prod` to multiply a Toeplitz-like matrix by vectors.

For stability reason, also the generators of A_{21} and F are replaced by the orthogonal ones, if these result in a smaller value of $\psi(A_{21})$ and $\psi(F)$, respectively.

`function` $X^{(k)}$ = `solvegen` $\left(\{A^{(k)}\}, B^{(k)}\right)$

% computes recursively the solution of the system $A^{(k)}\,X = B^{(k)}$

% $A^{(k)}$ has size n_k, displacement rank ρ and $\nabla(A^{(k)}) = C^{(k)}\Sigma^{(k)}C^{(k)T}$.

`if` $n = n_e$, $X^{(k)}$ = `elemgen` $\left(A^{(k)}, B^{(k)}\right)$;

`else`

$n = n_k$;

% partition into matrices of size n_{k+1}

$r_1 = (1 : n/2)$; $r_2 = (n/2 + 1 : n)$; $\overline{B} = B^{(k)}(r_1, :)$; $\underline{B} = B^{(k)}(r_2, :)$;

$\overline{C} = C^{(k)}(r_1, :)$; $\underline{C} = C^{(k)}(r_2, :)$; $\Sigma = \Sigma^{(k)}$;

compute the scalar α and the vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ and $\boldsymbol{v}_3$ defined in (14) and (15),

$C_{21} = \left[\underline{C}, \boldsymbol{e}_1\right]$; $D_{21} = \left[\overline{C}, \boldsymbol{v}_1\right]$; $K = \left[Z^T\,\overline{C}, -\boldsymbol{e}_{n/2}\right]$; $\Sigma_{21} = \begin{bmatrix} \Sigma & 0 \\ 0 & 1 \end{bmatrix}$;

$\left(\widehat{C}_{21}, \widehat{\Sigma}_{21}, \widehat{D}_{21}\right)$ = `replace` $\left(C_{21},\, \Sigma_{21}, D_{21}\right)$;

% first recursion substep, with $\nabla(A_{11}) = \overline{C}\Sigma\,\overline{C}^T$

$[U, C_F, G]$ = `solvegen`$\left(\{A_{11}\}, \left[\overline{B}\,, D_{21}\,, K\right]\right)$;

$D_F = C_{21} - Z$ `prod` $\left(\{A_{21}\}, G\right)$;

$\left(\widehat{C}_F, \widehat{\Sigma}_F, \widehat{D}_F\right)$ = `replace` $\left(C_F,\, \Sigma_{21}, D_F\right)$;

$C_S = \left[\underline{C}, \boldsymbol{v}_3\right]$ − `prod` $\left(\{A_{21}\}, C_F\right)$;

$\left(\widehat{C}_S, \widehat{\Sigma}_S, \widehat{D}_S\right)$ = `replace` $\left(C_S,\, \Sigma_S, D_S\right)$;

% second recursion substep, with $\nabla(S) = \widehat{C}_S\widehat{\Sigma}_S\widehat{D}_S^T$

$Y = \underline{B}$ − `prod` $\left(\{A_{21}\}, U\right)$;

$\underline{X}$ = `solvegen` $\left(\{S\}, Y\right)$;

$\overline{X} = U$ − `prod` $\left(\{F\}, \underline{X}\right)$;

$X^{(k)} = \begin{bmatrix} \overline{X} \\ \underline{X} \end{bmatrix}$;

`end`

The first time, the function is applied with $k = 0$, the generators of $A^{(0)} = A$ and the right-hand side $B^{(0)} = \boldsymbol{b}$. At the elementary level, any routine can be chosen as `elemgen`. A natural choice would be Levinson algorithm, which can be applied directly to the generators. If stability is at risk, a more stable method, like Cholesky method, can be chosen, but in this case the reconstruction of the elementary matrix from the generators is required.

In [4] the computational cost and the memory requirement of `solvegen` are given. Since the computational cost is of order $O(N \log_2^3 N)$ for $N \to \infty$, `solvegen` belongs to the class of superfast methods.

4 Analysis of stability

For the stability analysis we assume that the computations are carried out in a floating point arithmetic with unit roundoff ϵ such that $N\epsilon << 1$. The computed value of a variable (scalar, vector or matrix) v is denoted by $\widetilde{v}$ or by "$fl(v)$". We assume also that the quantities which appear in the bounds are not so large to invalidate a first order error analysis. For simplicity the term "$+O(\epsilon^2)$", which appears in the thesis of the theorems, is omitted in the proofs. Consequently, any expression of the form $x\,\widetilde{y}$, where $x = O(\epsilon)$ and $\widetilde{y} - y = O(\epsilon)$, is replaced by $x\,y$.

From ([8], Ch. 3) the following bound can be derived. Let A be an $n \times n$ matrix and $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ be three vectors, with $\boldsymbol{w} = \boldsymbol{u} - A\boldsymbol{v}$. If standard multiplication techniques are used to compute $\widetilde{\boldsymbol{w}} = fl(\boldsymbol{u} - fl(A\boldsymbol{v}))$, a matrix $\Theta_{A,\boldsymbol{v}}$ exists such that

$$\widetilde{\boldsymbol{w}} = \boldsymbol{u} - \big(A + \Theta_A\big)\boldsymbol{v} + \mathrm{diag}(\boldsymbol{\epsilon})\,\boldsymbol{w}, \quad \text{with} \quad \|\Theta_A\| \leq \epsilon\, n\, \|A\| + O(\epsilon^2), \tag{23}$$

where $\boldsymbol{\epsilon}$ is a vector whose components are bounded in modulus by ϵ. In the case A is Toeplitz-like, the product $A\boldsymbol{v}$ is computed by means of a function `prod` and

$$\widetilde{\boldsymbol{w}} = fl(\boldsymbol{u} - \texttt{prod}(\big\{A\big\}, \boldsymbol{v})).$$

Due to the well-known stability properties of FFT [1], we assume that a bound similar to (23) holds

$$\widetilde{\boldsymbol{w}} = \boldsymbol{u} - \big(A + \Theta_A\big)\boldsymbol{v} + \mathrm{diag}(\boldsymbol{\epsilon})\,\boldsymbol{w}, \quad \text{with} \quad \|\Theta_A\| \leq \epsilon\, \gamma(n)\, \psi(A) + O(\epsilon^2), \tag{24}$$

where $\gamma(n) = c\; n \log_2 n$ (c being a function of ρ, for a proof of this bound see [5]).

The present analysis aims at detecting the parameters which play the major role in the stability of `solvegen`. Algorithm `solvegen` can be regarded as composed of two parts:

- the *outer* part, which forms the framework of the algorithm and implements the recursion (already outlined in `solvemat`) using the generators of F and S,
- the *inner* part, which computes the generators of F and S (this part makes the essential difference of `solvegen` with respect to `solvemat`).

Following this approach, the error analysis of `solvegen` will deal first in Section 4.1 with the computation of the recursion, assuming that the errors of the computed generators are sufficiently bounded from above. Then the conditions upon which the computation of the generators meets these stability requirements will be analyzed in Section 4.2.

In this analysis we neglect the error due to the computation of the orthogonal generators by the SVD algorithm, which is commonly considered stable.

4.1 Stability of the outer part

At the kth level of recursion, with $k = 0, \ldots, p$, there are 2^k systems to be solved, each one of size $n_k = 2^{p-k} n_e$. For $r = 1, \ldots, 2^k$, let $\mathcal{S}_r^{(k)}$ be one of these systems. Let $A_r^{(k)}$ be its coefficient matrix, decomposed as in (12), $B_r^{(k)}$ be its right-hand side having $m_r^{(k)}$ columns and $X_r^{(k)}$ be its solution. To simplify the notation, we drop the indices k and r, denoting the system

$$A\,X = B, \quad \text{with} \quad A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad X = \begin{bmatrix} \overline{X} \\ \underline{X} \end{bmatrix}, \quad B = \begin{bmatrix} \overline{B} \\ \underline{B} \end{bmatrix}. \tag{25}$$

From $\mathcal{S}_r^{(k)}$ two subsystems $\mathcal{S}_{2r-1}^{(k+1)}$ and $\mathcal{S}_{2r}^{(k+1)}$ are derived, the first one with matrix A_{11}, the second one with the Schur matrix $S = A_{22} - A_{21}\,F$. Denoting by $\boldsymbol{x}_i$ and $\boldsymbol{b}_i$ the ith column of X and B, system (25) combines the m systems

$$A\,\boldsymbol{x}_i = \boldsymbol{b}_i, \quad \text{for} \quad i = 1, \ldots, m.$$

To further simplify the notation, we also drop the column index i, and consider the system

$$A\,\boldsymbol{x} = \boldsymbol{b}, \quad \text{with} \quad \boldsymbol{x} = \begin{bmatrix} \overline{\boldsymbol{x}} \\ \underline{\boldsymbol{x}} \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \overline{\boldsymbol{b}} \\ \underline{\boldsymbol{b}} \end{bmatrix} \tag{26}$$

to represent the ith system having $A_r^{(k)}$ as coefficient matrix, the ith column $\boldsymbol{b}_{r,i}^{(k)}$ of $B_r^{(k)}$ as right-hand side, and $\boldsymbol{x}_{r,i}^{(k)}$ as solution, for selected i, k and r.

At the last level of the recursion, for $k = p$, the size of (26) is n_e and we assume that the computed solution $\widetilde{\boldsymbol{x}}$ satisfies

$$\big(A + \Delta_A\big)\,\widetilde{\boldsymbol{x}} = \boldsymbol{b}, \quad \text{with} \quad \|\Delta_A\| \le \epsilon\,\pi_e\,\psi(A) + O(\epsilon^2), \tag{27}$$

where π_e is a function of n_e, related to the method used at the elementary level (a similar bound appears in Theorem 10.4 of [8] for Cholesky method).

The framework of the outer part of `solvegen` applied to (27) is composed of the following instructions

$$\begin{aligned} \boldsymbol{u} &= \texttt{solvegen}\big(\{A_{11}\}, \overline{\boldsymbol{b}}\big), & \boldsymbol{y} &= \underline{\boldsymbol{b}} - \texttt{prod}\,\big(\{A_{21}\}, \boldsymbol{u}\big), \\ \underline{\boldsymbol{x}} &= \texttt{solvegen}\,\big(\{S\}, \boldsymbol{y}\big), & \overline{\boldsymbol{x}} &= \boldsymbol{u} - \texttt{prod}\,\big(\{F\}, \underline{\boldsymbol{x}}\big). \end{aligned} \tag{28}$$

Obviously, the errors of this computation depend on how well the generators $\widetilde{C}_F$ and $\widetilde{D}_F$ of F and $\widetilde{C}_S$ and $\widetilde{\Sigma}_S$ of S have been determined in the inner part. Thus, in this outer part we assume that the errors of $\widetilde{F}$ and $\widetilde{S}$ are sufficiently bounded from above.

Theorem 1 *Consider the $n_k \times n_k$ system (26) for selected i, k and r. If the generators $\widetilde{C}_F$, $\widetilde{D}_F$, $\widetilde{C}_S$, $\widetilde{\Sigma}_S$ computed in the inner part are such that the corresponding matrices $\widetilde{F}$ and $\widetilde{S}$ satisfy*

$$\|A_{11}\,(\widetilde{F} - F)\| + \|\widetilde{S} - S'\| \le \epsilon\,\omega_{k,r}\,\psi(A)\,\big(1 + \psi(F)\big) + O(\epsilon^2), \tag{29}$$

for $S' = A_{22} - A_{21}\,\widetilde{F}$ and a suitable $\omega_{k,r}$, then the solution of system (26) computed by $\widetilde{\boldsymbol{x}} = fl\Big(\texttt{solvegen}\big(\{A\},\boldsymbol{b}\big)\Big)$ satisfies

$$\big(A + \Delta_A\big)\,\widetilde{\boldsymbol{x}} = \boldsymbol{b},\ \textit{with}\ \|\Delta_A\| \le \epsilon\,\pi_k\,\psi(A) + O(\epsilon^2), \tag{30}$$

for a suitable π_k not depending on the indices i and r.

Proof. The proof is by induction on k. For $k = p$, the thesis follows from (27), with $\pi_p = \pi_e$. For $k = p-1,\ldots,0$, the quantities effectively computed by instructions (28) are

- $\widetilde{\boldsymbol{u}} = fl\Big(\texttt{solvegen}\big(\{A_{11}\},\overline{\boldsymbol{b}}\big)\Big)$. By the inductive hypothesis (working to first order)

$$\big(A_{11} + \Delta_{A_{11}}\big)\,\widetilde{\boldsymbol{u}} = \overline{\boldsymbol{b}}, \quad \text{with} \quad \|\Delta_{A_{11}}\| \le \epsilon\,\pi_{k+1}\,\psi(A_{11}). \tag{31}$$

- $\widetilde{\boldsymbol{y}} = fl\Big(\underline{\boldsymbol{b}} - \ fl\big(\texttt{prod}\ \big(\{A_{21}\},\widetilde{\boldsymbol{u}}\big)\big)\Big)$. From (24), setting $\gamma_{k+1} = \gamma(n_k/2)$

$$\widetilde{\boldsymbol{y}} = \underline{\boldsymbol{b}} - \big(A_{21} + \Theta_{A_{21}}\big)\,\widetilde{\boldsymbol{u}} + \mathrm{diag}(\boldsymbol{\epsilon})\,\widetilde{\boldsymbol{y}}, \quad \text{with} \quad \|\Theta_{A_{21}}\| \le \epsilon\,\gamma_{k+1}\,\psi(A_{21}). \tag{32}$$

- $\underline{\widetilde{\boldsymbol{x}}} = fl\Big(\texttt{solvegen}\ \big(\{\widetilde{S}\},\widetilde{\boldsymbol{y}}\big)\Big)$. By the inductive hypothesis

$$\big(\widetilde{S} + \Delta_S\big)\,\underline{\widetilde{\boldsymbol{x}}} = \widetilde{\boldsymbol{y}} \quad \text{with} \quad \|\Delta_S\| \le \epsilon\,\pi_{k+1}\,\psi(S). \tag{33}$$

- $\overline{\widetilde{\boldsymbol{x}}} = fl\Big(\widetilde{\boldsymbol{u}} - \ fl\big(\texttt{prod}\ \big(\{\widetilde{F}\},\underline{\widetilde{\boldsymbol{x}}}\big)\big)\Big)$. From (24)

$$\overline{\widetilde{\boldsymbol{x}}} = \widetilde{\boldsymbol{u}} - \big(\widetilde{F} + \Theta_F\big)\,\underline{\widetilde{\boldsymbol{x}}} + \mathrm{diag}(\boldsymbol{\epsilon})\,\overline{\widetilde{\boldsymbol{x}}} \quad \text{with} \quad \|\Theta_F\| \le \epsilon\,\gamma_{k+1}\,\psi(F). \tag{34}$$

Replacing $\widetilde{\boldsymbol{u}}$ from (34) into (31) we get

$$\overline{\boldsymbol{b}} = A_{11}\,\big(\overline{\widetilde{\boldsymbol{x}}} + \big(\widetilde{F} + \Theta_F\big)\,\underline{\widetilde{\boldsymbol{x}}} - \mathrm{diag}(\boldsymbol{\epsilon})\,\overline{\widetilde{\boldsymbol{x}}}\big) + \Delta_{A_{11}}\big(\overline{\widetilde{\boldsymbol{x}}} + \widetilde{F}\,\underline{\widetilde{\boldsymbol{x}}}\big)$$

replacing $\widetilde{\boldsymbol{y}}$ from (33) and $\widetilde{\boldsymbol{u}}$ from (34) into (32) we get

$$\underline{\boldsymbol{b}} = \big(\widetilde{S}+\Delta_S\big)\,\underline{\widetilde{\boldsymbol{x}}} + A_{21}\big(\overline{\widetilde{\boldsymbol{x}}} + \big(\widetilde{F}+\Theta_F\big)\,\underline{\widetilde{\boldsymbol{x}}} - \mathrm{diag}(\boldsymbol{\epsilon})\,\overline{\widetilde{\boldsymbol{x}}}\big) + \Theta_{A_{21}}\big(\overline{\widetilde{\boldsymbol{x}}} + \widetilde{F}\,\underline{\widetilde{\boldsymbol{x}}}\big) - \mathrm{diag}(\boldsymbol{\epsilon})\,\widetilde{S}\,\underline{\widetilde{\boldsymbol{x}}}.$$

Then

$$\begin{bmatrix} \overline{\boldsymbol{b}} \\ \underline{\boldsymbol{b}} \end{bmatrix} = \begin{bmatrix} A_{11}\,\overline{\widetilde{\boldsymbol{x}}} + A_{12}\,\underline{\widetilde{\boldsymbol{x}}} \\ A_{21}\,\overline{\widetilde{\boldsymbol{x}}} + A_{22}\,\underline{\widetilde{\boldsymbol{x}}} \end{bmatrix} + \Delta_A \begin{bmatrix} \overline{\widetilde{\boldsymbol{x}}} \\ \underline{\widetilde{\boldsymbol{x}}} \end{bmatrix} \tag{35}$$

where

$$\Delta_A = K_1 + K_2, \quad \text{with} \quad K_1 = \begin{bmatrix} O & A_{11}(\widetilde{F} - F) \\ O & \widetilde{S} - S' + \Delta_S - \mathrm{diag}(\boldsymbol{\epsilon})\,\widetilde{S} \end{bmatrix},$$

$$K_2 = \begin{bmatrix} -A_{11}\mathrm{diag}(\boldsymbol{\epsilon}) + \Delta_{A_{11}} & A_{11}\Theta_F + \Delta_{A_{11}}\widetilde{F} \\ -A_{21}\mathrm{diag}(\boldsymbol{\epsilon}) + \Theta_{A_{21}} & A_{21}\Theta_F + \Theta_{A_{21}}\widetilde{F} \end{bmatrix}.$$

For the 1-norm of K_1 we get from (29), (33) and (22)

$$\|K_1\| \leq \epsilon \left(\omega_{k,r} + 2\pi_{k+1} + 1\right) \psi(A) \left(1 + \psi(F)\right).$$

For the 1-norm of K_2 we get from (31), (32) and (34)

$$\|K_2\| \leq \epsilon \left(\pi_{k+1} + 3\,\gamma_{k+1}\right) \psi(A) \left(1 + \psi(F)\right).$$

Then

$$\|\Delta_A\| \leq \epsilon\, \pi_k\, \psi(A),$$

where

$$\pi_k = (3\pi_{k+1} + 3\gamma_{k+1} + \omega_{k,r} + 1) \left(1 + \psi(F)\right). \tag{36}$$

□

4.2 Stability of the inner part

The following theorem analyzes how the generators of $\widetilde{F}$ and $\widetilde{S}$ computed by the instructions

$$\begin{aligned}
&C_F = \texttt{solvegen}\big(\{A_{11}\}, D_{21}\big),\\
&D_F = C_{21} - Z\,\texttt{prod}\,\big(\{A_{21}\}, G\big), \text{ where } G = \texttt{solvegen}\big(\{A_{11}\}, K\big),\\
&C_S = \big[\underline{C}, \boldsymbol{v}_3\big] - \texttt{prod}\,\big(\{A_{21}\}, C_F\big).
\end{aligned}$$

satisfy the requirement of Theorem 1.

Theorem 2 *The matrices $\widetilde{F}$ such that $\nabla(\widetilde{F}) = \widetilde{C}_F \Sigma_F \widetilde{D}_F^T$ and $\widetilde{S}$ such that $\nabla(\widetilde{S}) = \widetilde{C}_S \Sigma_S \widetilde{D}_S$, considered at the kth inductive step of the proof of Theorem 1, satisfy relation (29) for a suitable $\omega_{k,r}$.*

Proof. For $j = 1, \ldots, \rho + 1$, consider the jth component of $\nabla(F)$, i.e. $\sigma_{F,j}\, \boldsymbol{c}_{F,j}\, \boldsymbol{d}_{F,j}^T$, with $\sigma_{F,j} = \sigma_j$ for $j = 1, \ldots, \rho$ and $\sigma_{F,\rho+1} = 1$.

- The jth left generator $\boldsymbol{c}_{F,j}$ is computed by

$$\widetilde{\boldsymbol{c}}_{F,j} = fl\Big(\texttt{solvegen}\big(\{A_{11}\}, \boldsymbol{d}_{21}\big)\Big).$$

The error $\delta \boldsymbol{c}_{F,j} = \widetilde{\boldsymbol{c}}_{F,j} - \boldsymbol{c}_{F,j}$ satisfies $\|\delta \boldsymbol{c}_{F,j}\| \leq \epsilon\, \eta_{c_j}\, \|\boldsymbol{c}_{F,j}\|$, where $\epsilon\, \eta_{c_j}$ is an upper bound to the relative error.

- The jth right generator $\boldsymbol{d}_{F,j}$ is computed by

$$\widetilde{\boldsymbol{g}}_j = fl\Big(\texttt{solvegen}\big(\{A_{11}\}, \boldsymbol{k}_j\big)\Big), \quad \widetilde{\boldsymbol{d}}_{F,j} = fl\Big(C_{21} - Z\, fl\big(\texttt{prod}\,\big(\{A_{21}\}, \widetilde{\boldsymbol{g}}_j\big)\big)\Big).$$

The error $\delta \boldsymbol{g}_j = \widetilde{\boldsymbol{g}}_j - \boldsymbol{g}_j$ satisfies $\|\delta \boldsymbol{g}_j\| \leq \epsilon\, \eta_{g_j}\, \|\boldsymbol{g}_j\|$, where $\epsilon\, \eta_{g_j}$ is an upper bound to the relative error. By (24)

$$\widetilde{\boldsymbol{d}}_{F,j} = \boldsymbol{c}_{21,j} - Z\big(A_{21} + \Theta_{A_{21}}\big)\widetilde{\boldsymbol{g}}_j + \mathrm{diag}\big(\boldsymbol{\epsilon}_j\big)\boldsymbol{d}_{F,j}, \quad \|\Theta_{A_{21}}\| \leq \epsilon\, \gamma_{k+1}\, \psi(A_{21}).$$

Replacing $\widetilde{\boldsymbol{g}}_i$ we get $\widetilde{\boldsymbol{d}}_{F,j} = \boldsymbol{d}_{F,j} + \delta\boldsymbol{d}_{F,j}$, where

$$\delta\boldsymbol{d}_{F,j} = -Z\big(A_{21}\delta\boldsymbol{g}_j + \Theta_{A_{21}}\boldsymbol{g}_j\big) + \mathrm{diag}\big(\boldsymbol{\epsilon}_j\big)\boldsymbol{d}_{F,j},$$

then

$$\|\delta\boldsymbol{d}_{F,j}\| \le \epsilon\,\big(\eta_{g_j}\,\|A_{21}\| + \gamma_{k+1}\,\psi(A_{21})\big)\|\boldsymbol{g}_j\| + \epsilon\,\|\boldsymbol{d}_{F,j}\|.$$

We introduce now the parameter

$$\tau_{k+1,r} = \max_{j=1,\dots\rho+1} \frac{\|\boldsymbol{g}_j\|}{\|\boldsymbol{d}_{F,j}\|}\,\psi(A_{21})$$

to measure the largest cancellation error in the computation of the columns of $D_F = C_{12} - Z\,A_{21}\,G$. We get

$$\|\delta\boldsymbol{d}_{F,j}\| \le \epsilon\,\big(\tau_{k+1,r}\,(\eta_{g_j} + \gamma_{k+1}) + 1\big)\,\|\boldsymbol{d}_{F,j}\|.$$

Since

$$\widetilde{\boldsymbol{c}}_{F,j}\,\widetilde{\boldsymbol{d}}_{F,j}^T - \boldsymbol{c}_{F,j}\,\boldsymbol{d}_{F,j}^T = \boldsymbol{c}_{F,j}\,\delta\boldsymbol{d}_{F,j}^T + \delta\boldsymbol{c}_{F,j}\,\boldsymbol{d}_{F,j}^T,$$

we get

$$\begin{aligned}\nabla(\widetilde{F}) - \nabla(F) &= \sum_{j=1}^{\rho+1}\sigma_j\,\widetilde{\boldsymbol{c}}_{F,j}\,\widetilde{\boldsymbol{d}}_{F,j}^T - \sum_{j=1}^{\rho+1}\sigma_j\,\boldsymbol{c}_{F,j}\,\boldsymbol{d}_{F,j}^T \\ &= \sum_{j=1}^{\rho+1}\boldsymbol{c}_{F,j}\,\delta\boldsymbol{d}_{F,j}^T + \sum_{j=1}^{\rho+1}\delta\boldsymbol{c}_{F,j}\,\boldsymbol{d}_{F,j}^T.\end{aligned}$$

Setting

$$\eta_{k+1,r} = \max_{j=1,\dots\rho+1}\max\{\eta_{c_j},\eta_{g_j}\},$$

from (10) we have

$$\|\widetilde{F} - F\| \le \sum_{j=1}^{\rho+1}\|\boldsymbol{c}_{F,j}\|\,\|\delta\boldsymbol{d}_{F,j}\| + \sum_{j=1}^{\rho+1}\|\delta\boldsymbol{c}_{F,j}\|\,\|\boldsymbol{d}_{F,j}\| \le \epsilon\,\omega_F\,\psi(F),$$

where

$$\omega_F = \eta_{k+1,r}\,(\tau_{k+1,r} + 1) + \gamma_{k+1}\,\tau_{k+1,r} + 1. \tag{37}$$

- Setting $Q = \big[\underline{C}, \boldsymbol{v}_3\big]$, from (21) we have

$$\nabla(S') = C_{S'}\,\Sigma_F\,D_{S'}^T, \quad \text{where} \quad C_{S'} = Q - A_{21}\widetilde{C}_F, \quad D_{S'} = \widetilde{D}_F.$$

The generators of $\widetilde{S}$ are $\widetilde{C}_S = \widetilde{C}_{S'}$ and $\widetilde{D}_S = \widetilde{D}_{S'}$. The jth left generator $\boldsymbol{c}_{S,j}$ is effectively computed by

$$\widetilde{\boldsymbol{c}}_{S,j} = fl\Big(\boldsymbol{q}_j - \ fl\big(\texttt{prod}\ (\{A_{21}\},\widetilde{\boldsymbol{c}}_{F,j})\big)\Big).$$

By (24) we have

$$\widetilde{\boldsymbol{c}}_{S,j} = \boldsymbol{q}_j - \big(A_{21} + \Theta_{A_{21}}\big)\widetilde{\boldsymbol{c}}_{F,j} + \mathrm{diag}\big(\boldsymbol{\epsilon}_j\big)\boldsymbol{c}_{S,j}, \quad \text{with} \quad \|\Theta_{A_{21}}\| \le \epsilon\,\gamma_{k+1}\psi(A_{21}),$$

hence

$$\widetilde{\boldsymbol{c}}_{S,j} = \boldsymbol{c}_{S',j} - \Theta_{A_{21}}\, \boldsymbol{c}_{F,j} + \operatorname{diag}\big(\boldsymbol{\epsilon}_j\big)\boldsymbol{c}_{S,j},$$

and

$$\begin{aligned}\nabla(\widetilde{S}) - \nabla(S') = \big(\widetilde{C}_S - C_{S'}\big)\Sigma_F\, \widetilde{D}_F^T &= \sum_{j=1}^{\rho+1} \sigma_j \left(\widetilde{\boldsymbol{c}}_{S,j} - \boldsymbol{c}_{S',j}\right)\widetilde{\boldsymbol{d}}_{F,j}^T \\ &= -\sum_{j=1}^{\rho+1} \sigma_j\, \Theta_{A_{21}}\, \boldsymbol{c}_{F,j}\, \boldsymbol{d}_{F,j}^T + \sum_{j=1}^{\rho+1} \sigma_j \operatorname{diag}\big(\boldsymbol{\epsilon}_j\big)\boldsymbol{c}_{S,j}\, \boldsymbol{d}_{S,j}^T.\end{aligned} \tag{38}$$

From (10) it follows that

$$\begin{aligned}\|\widetilde{S} - S'\| &\le \sum_{j=1}^{\rho+1} \|\Theta_{A_{21}}\, \boldsymbol{c}_{F,j}\|\, \|\boldsymbol{d}_{F,j}\| + \epsilon \sum_{j=1}^{\rho+1} \|\boldsymbol{c}_{S,j}\|\, \|\boldsymbol{d}_{S,j}\| \\ &\le \epsilon \left(\gamma_{k+1}\psi(A_{21})\psi(F) + \psi(S)\right),\end{aligned}$$

and from (22) we have

$$\|\widetilde{S} - S')\| \le \epsilon\, \omega_S\, \psi(A)\left(1 + \psi(F)\right), \quad \text{where} \quad \omega_S = \gamma_{k+1} + 2. \tag{39}$$

From (37) and (39) it follows that relation (29) holds with

$$\omega_{k,r} = \omega_F + \omega_S = \left(\eta_{k+1,r} + \gamma_{k+1}\right)\left(\tau_{k+1,r} + 1\right) + 3. \tag{40}$$

□

4.3 Stability of the whole algorithm

We are now able to investigate the stability of `solvegen`.

Theorem 3 *Let $\widetilde{\boldsymbol{x}}$ be the solution of (1) computed by* `solvegen`*. Assuming that the solutions of the elementary systems satisfy (27), a matrix Δ_A exists such that*

$$\left(A + \Delta_A\right)\widetilde{\boldsymbol{x}} = \boldsymbol{b}, \quad \textit{with} \quad \|\Delta_A\| \le \epsilon\, \pi_0\, \psi(A) + O(\epsilon^2),$$

for a suitable π_0.

Proof. The solution of (1) computed by `solvegen` is $\widetilde{\boldsymbol{x}} = \widetilde{\boldsymbol{x}}_{1,1}^{(0)}$. Using the result of Theorem 1 we can give an estimate of the quantity π_0 which appears in the bound (30) for $k = 0$, i.e.

$$\left(A^{(0)} + \Delta_{A^{(0)}}\right)\widetilde{\boldsymbol{x}}^{(0)} = \boldsymbol{b}^{(0)}, \quad \|\Delta_{A^{(0)}}\| \le \epsilon\, \pi_0\, \psi(A^{(0)}),$$

where $A^{(0)} = A$ and $\boldsymbol{b}^{(0)} = \boldsymbol{b}$. At the kth level of recursion, when the systems to be solved have $n_k = 2^{p-k}$ rows, restoring the notation with the indices of the

involved matrices and replacing $\omega_{k,r}$ in (36) with (40), the thesis of Theorem 1 becomes

$$\left(A_r^{(k)} + \Delta_{A_r^{(k)},i}\right) \widetilde{\boldsymbol{x}}_{r,i}^{(k)} = \boldsymbol{b}_{r,i}^{(k)}, \quad \|\Delta_{A_r^{(k)},i}\| \leq \epsilon\, \pi_k\, \psi(A_r^{(k)}), \quad i = 1, \ldots, m_r,$$

where

$$\pi_k = \left(1 + \psi(F_r^{(k)})\right) \left(3\pi_{k+1} + \gamma_{k+1}\,(\tau_{k+1,r} + 4) + \eta_{k+1,r}\,(\tau_{k+1,r} + 1) + 4\right).$$

Setting

$$\psi_k = \max_r \left(1 + \psi(F_r^{(k)})\right), \quad \nu_k = n_k \log_2 n_k,$$
$$\alpha_k = c_1 \max_r \tau_{k,r}, \quad \beta_k = c_2 \max_r \eta_{k,r}\, \tau_{k,r},$$

for suitable constants c_1 and c_2, the recursive relation becomes

$$\pi_k \leq \left(3\,\pi_{k+1} + \alpha_{k+1}\,\nu_{k+1} + \beta_{k+1}\right) \psi_k, \quad \text{for} \quad k = 0, \ldots, p-1.$$

Then

$$\begin{aligned} \pi_0 &\leq 3^p\, \pi_e \prod_{k=1}^{p} \psi_{k-1} + \sum_{k=1}^{p} \left(3^{k-1}\,(\alpha_k \nu_k + \beta_k) \prod_{i=1}^{k} \psi_{i-1}\right) \\ &\leq \Psi \left(3^p\, \pi_e + \sum_{k=1}^{p} 3^{k-1}\,(\alpha_k \nu_k + \beta_k)\right), \quad \text{where} \quad \Psi = \prod_{k=1}^{p} \psi_{k-1}. \end{aligned} \tag{41}$$

□

From (41) the stability of `solvegen` appears to depend on the parameters ψ_k, α_k and β_k. Unlike parameters α_k and β_k, parameter ψ_k can be easily estimated during the computation. It measures the magnitude of the matrices $F_r^{(k)}$ through the norms of the generators and can be used to monitor the behavior of the error. The parameter β_k has been introduced in the analysis of the errors arising in the computation of the generators of the matrices $F_r^{(k)}$. We expect it to be related to the conditioning of the matrices $A_r^{(k)}$ involved during the computation. If the matrix A has some special structural properties, the conditioning of all the matrices $A_r^{(k)}$ can be upper bounded by the conditioning of A. In particular, this happens if A is positive definite. Moreover, when A is diagonally dominant the norms $\|F_r^{(k)}\|$ are bounded and we expect that if A is well conditioned all the ψ_k are upper bounded.

Limiting our analysis to classes of matrices for which the parameters α_k, β_k and ψ_k are bounded by quantities $\widehat{\alpha}$, $\widehat{\beta}$ and $\widehat{\psi}$ independent from N, we have

$$\pi_0 \leq \widehat{\psi}^{\,p} \left(3^p \pi_e + \widehat{\alpha} \sum_{k=1}^{p} 3^{k-1}\, n_k \log_2 n_k + \widehat{\beta} \sum_{k=1}^{p} 3^{k-1}\right), \quad \text{where} \quad n_k = 2^{p-k}\, n_e.$$

Then

$$\pi_0 \leq c_3\, N^m, \quad \text{where} \quad m = \log_2 (3\widehat{\psi}),$$

showing that π_0 depends at most polynomially on N.

5 Numerical experiments

The experiments have been conducted on an Intel Core Duo @ 3 GHz, 2GB RAM, using double precision arithmetic. Three sets of matrices have been considered:

(M1) The first set consists of 250 Toeplitz-like matrices of size 2^8 with displacement rank $\rho = 5$ and elementary size $n_e = 8$. These matrices are randomly generated with positive diagonal by means of the generators $C = D$ and of the signature matrix Σ, in such a way to give different conditioning and different values of the quantity $d = \min_i \left(1 - \sum_{j \neq i} |a_{i,j}|/|a_{ii}|\right)$ which is related to the point diagonal dominance of A. If $d > 0$, the matrix is diagonally dominant and positive definite, but positive definite matrices may occur also for negative values of d.

(M2) The second set consists of 250 symmetric positive definite Toeplitz matrices (displacement rank $\rho = 2$) of size 2^7 with elementary size $n_e = 4$. These matrices are generated as described in [13], through Schur parameters ρ_j, $j = 1, \ldots, N$, distributed uniformly over symmetric intervals $\mathcal{I}_\ell = [-\ell, \ell]$, with $0.01 \leq \ell \leq 0.5$, and ρ_{10} and ρ_{15} changed to values with module close to one. The spectral properties of these matrices depend on ℓ and on the closeness to 1 of $|\rho_{10}|$ and $|\rho_{15}|$.

(M3) The 210 matrices of M3 are still generated through Schur parameters ρ_j, $j = 1, \ldots, N$, distributed uniformly over the intervals $\mathcal{I}_\ell$, with $0.001 \leq \ell \leq 0.35$, without further changes. When the endpoints of the intervals $\mathcal{I}_\ell$ are close to zero the generated matrices are diagonally dominant.

The left-hand side vector $\boldsymbol{b}$ for the sets of matrices is computed from an exact solution, with elements randomly generated in $[0, 1]$. The considerations of the previous section suggest that the difficulty of solving a Toeplitz-like system by `solvegen` can be related to the conditioning of the matrices $A_r^{(k)}$ for any k. Since in general, the computation of the conditioning of all the matrices is unfeasible, we associate to each problem the parameter

$$\kappa = \max_r \texttt{cond}\left(A_r^{(p)}\right),$$

where `cond` is the condition number. This parameter measures the greatest conditioning among the elementary blocks. In our experiments, the value of κ varies between $10^{0.9}$ and $10^{5.2}$ for M1 matrices and between $10^{0.5}$ and $10^{2.5}$ for M2 matrices. For M3 matrices the value of κ is bounded by $10^{1.5}$.

The scaled-independent relative residual

$$R = \frac{\|\boldsymbol{b} - A\widetilde{\boldsymbol{x}}\|}{\|A\|\|\widetilde{\boldsymbol{x}}\| + \|\boldsymbol{b}\|}$$

can be taken as an experimental estimate of the perturbation $\|\Delta_A\|$ occurring in Theorem 3. Of course, R is expected to have a better behavior than the bound

appearing in the theorem. The aim of the experimentation is to point out that the quantity Ψ, which appears in the theoretical bound (41) of the perturbation $\|\Delta_A\|$ plays a relevant role also in practice.

Figure 1 shows the residuals R (black points) plotted versus κ in Log-Log scale for problems M1 on the left and M2 on the right. The figure shows also the corresponding values of Ψ (gray points), suitably scaled by a factor, in order to put the graphs in the same figure. It appears that parameter Ψ well describes the behavior of the residual. Moreover the points lie in a strip around a straight line, pointing out a polynomial growth of R and Ψ with respect to κ.

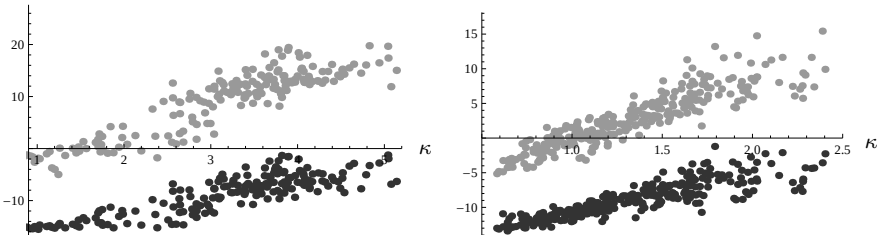

Figure 1: Log-Log plot of the relative residuals R (black points) and of $\Psi/10^{10}$ (gray points) as functions of κ, for problems M1 (on the left) and problems M2 (on the right).

Figure 2 shows the residual in Log scale plotted versus d for matrices M3. More precisely, the figure on the left refers to matrices generated through ρ_j belonging to $\mathcal{I}_\ell$, with $0.01 \leq \ell \leq 0.35$. These matrices, even if not diagonally dominant, are well conditioned with small absolute values of parameter d. The figure on the right refers to diagonally dominant matrices generated through ρ_j belonging to $\mathcal{I}_\ell$, with $0.001 \leq \ell \leq 0.015$. It is evident that for these matrices `solvegen` has a good performance, the stronger the diagonally dominance, the better the performance. From the figure on the left it appears that parameter Ψ may result in a pessimistic estimate of the residual especially for problems with good spectral properties.

6 Conclusions

In this paper we have performed a theoretical error analysis of a superfast algorithm recently introduced by the authors in [4]. This analysis allows the characterization of some parameters which rule the stability behavior of the algorithm. Among them, the parameters ψ_k, measuring the magnitude of the matrices $F_k^{(r)}$ through their generators, can be easily estimated during the computation. The numerical experimentation points out that the quantity $\Psi = \prod_{k=1}^{p} \psi_{k-1}$, which

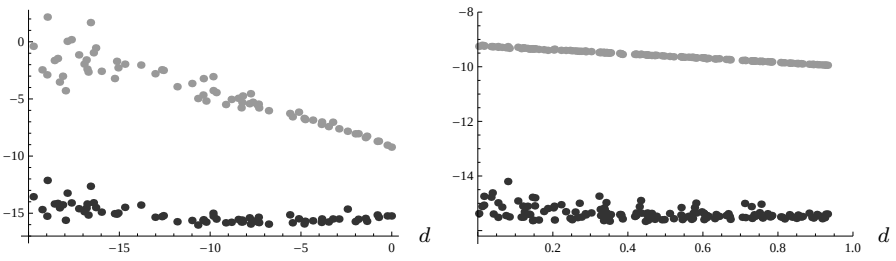

Figure 2: Log plot of the relative residuals R (black points) and of $\Psi/10^{10}$ (gray points) as functions of d, for problems M2.

appears in the theoretical bound (41), well describes also in practice the residual behavior. Anyway, the use of Ψ for estimating the residual may be too pessimistic, especially when `solvegen` has a good performance.

References

[1] M. Arioli, H. Munthe-Kaas, L. Valdettaro, Componentwise error analysis for FFT's with applications to fast Helmholtz solvers, *Numer. Algorithms*, 12, (1996), pp. 65-88.

[2] R. R. Bitmead and B. D. O. Anderson, "Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations", *Linear Algebra Appl.*, 34, pp. 103-116, 1980.

[3] P. Favati, G. Lotti and O. Menchi, "Stability of the Levinson algorithm for Toeplitz-like systems", *SIAM Journal on Matrix Analysis and Applications*, 31, pp. 2531-2552, 2010.

[4] P. Favati, G. Lotti and O. Menchi, "Superfast solution of Toeplitz-like systems", Tech. Report IIT TR-24/2011.

[5] P. Favati, G. Lotti and O. Menchi, "Stability analysis of the product via FFT of a Toeplitz-like matrix", Tech. Report IIT TR-10/2014.

[6] M. Gu, "Stable and Efficient Algorithms for Structured Systems of Linear Equations", *SIAM Journal on Matrix Analysis and Applications*, 19, pp. 279-306, 1998.

[7] G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin, 1984.

[8] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, PA, 1996.

[9] T. Kailath, S.-Y. Kung and M. Morf, "Displacement ranks of matrices and linear equations", *J. Math. Anal. Appl.*, 68, pp. 395-407, 1979.

[10] T. Kailath and A. H. Sayed, "Displacement structure: theory and applications", *SIAM Rev.*, 37, pp. 297-386, 1995.

[11] T. Kailath, A. Viera, and M. Morf, Inverses of Toeplitz operators, innovations and orthogonal polynomials, *SIAM Rev.*, 20, pp. 106-119, 1978.

[12] M. Morf, Doubling algorithms for Toeplitz and related equations, in *Proc. IEEE Internat. conf. on Acoustics, Speech, & Signal Processing*, pp. 954-959, Denver, Colorado, 1980.

[13] M. Stewart, "A Superfast Toeplitz Solver with Improved Numerical Stability ", *SIAM J. Matrix Anal. Appl.*, 25, pp. 669-693, 2003.