Journées Nationales de Calcul Formel

Rencontre organisée par :
Paola Boito, Alin Bostan, Adrien Poteaux et Mohab Safey el Din

2014

Dario A. Bini

Matrix structures and applications

Vol. 4, n° 1 (2014), Course n° I, p. 1-45.

<http://ccirm.cedram.org/item?id=CCIRM_2014__4_1_A1_0>

Centre international de rencontres mathématiques
U.M.S. 822 C.N.R.S./S.M.F.
Luminy (Marseille) France

Les cours du C.I.R.M.
Vol. **4** n° 1 (2014) 1-45
Cours n° I

Matrix structures and applications

Dario A. Bini

Introduction

Matrix structures are encountered in various guises in a wide variety of mathematical problems from theory, applications and scientific computing. Structures are the expression, given in terms of the language of linear algebra, of the specific properties satisfied by the mathematical model.

As an example, the most simple structure widely encountered is the one of *banded matrices* where the entries $a_{i,j}$ of a matrix A are zero for $|i-j|$ larger than a given small constant value $k > 0$. This kind of structure reflects the locality features of some functions involved in the model or the local action of some operators, like derivatives in differential equations or point-spread functions in image restoration models.

Another example of structure, of particular interest in the world of Internet is *sparsity*. Informally, an $n \times n$ matrix is sparse if the number of its nonzero entries is of the order of n. In complex network analysis like in the social networks, or in search engines as in the Google PageRank problem, sparsity reflects the fact that not all the parties involved in the game, say web pages or member of a social network, are connected to each other but the set of connections starting from a node is independent of the overall number of nodes.

The analysis of matrix structures concerning both theoretical and computational properties, is fundamental for designing fast and highly effective algorithms to solve specific problems where the dimension is very large. In fact, when the size is huge, general purpose algorithms require a CPU time of years or centuries even by using the fastest super-computers available nowadays. An example is the PageRank problem where a system of size larger than 10^{10} must be solved to sort web pages according to their importance. In this case, Gaussian elimination would require many billions of years of CPU time, whereas iterative methods, specifically designed for sparse matrices allow to solve the problem in real time with a sufficiently good approximation.

Structure analysis is important not only for algorithm design and applications but also for the variety, richness and beauty of the mathematical tools that are involved in the research. Significant in this regard is the following sentence from the mathematician Alexander Grothendieck, who received the Fields medal in 1966: "If there is one thing in mathematics that fascinates me more than anything else (and doubtless always has), it is neither number nor size, but always form. And among the thousand-and-one faces whereby form chooses to reveal itself to us, the one that fascinates me more than any other and continues to fascinate me, is the structure hidden in mathematical things."

A class of important structured matrices is the one of *Toeplitz matrices*, characterized by the fact that their entries are invariant under shifts along the main diagonal direction. They are ubiquitous in applications and appear when some sort of shift invariance property is satisfied by the mathematical objects involved in the model. They are encountered in many different fields like probability, statistics, signal processing, image restoration, polynomials and power series computations, queueing models, stochastic processes, spline interpolation, and more.

It is nice to discover that the basic polynomial computations like polynomial multiplication, quotient and remainder, polynomial gcd, can be rephrased in terms of Toeplitz matrix computation.

Course taught during the meeting "Journées Nationales de Calcul Formel" organized by Paola Boito, Alin Bostan, Adrien Poteaux and Mohab Safey el Din. 3-7 novembre 2014, C.I.R.M. (Luminy).

A wide literature exists concerning Toeplitz matrices. It covers problems like the analysis of asymptotic spectral properties, where tools from functional analysis and operator theory are used; the study of related matrix algebras and fast discrete transforms; the analysis of preconditioners for the iterative solution of Toeplitz systems; the analysis of displacement operators, which enable one to represent the inverse of a Toeplitz matrix in a nice form; the analysis of fast and super-fast algorithms for solving Toeplitz and Toeplitz-like linear systems with their interplay with Cauchy-like matrices.

Another interesting class of structured matrices is the class of *quasi-separable matrices*. The interest in this class is mainly originated by the fact that the inverses of banded matrices have this property. Informally speaking, a quasi-separable matrix is a matrix where all its submatrices strictly contained in the upper triangular part or in the lower triangular part have low rank. This structure is more hidden and can be detected with more difficulty.

A wide literature exists concerning this class starting from the pioneering work of Gantmacher and Krein [45], and arriving at the most recent algorithms for processing quasi-separable matrices designed by the several research groups working in this field. For a detailed commented reference list we refer to the books [79], [80], [41], [42] and to the paper [78].

It is nice to discover that all the companion matrices like Frobenius, colleague and comrade matrices share this important property.

In this short course we provide an overview of structural and computational properties of Toeplitz matrices, quasi-separable matrices and of related matrix structures. Some examples of their applications are given. Concerning Toeplitz matrices we describe their asymptotic spectral properties, their interplay with FFT and trigonometric matrix algebras including ϵ-circulant and triangular Toeplitz matrices, the concept of displacement rank, the Gohberg-Semencul-Trench inversion formula, algorithms for solving Toeplitz systems, the interplay between Toeplitz matrices and Laurent power series.

Applications are shown in the fields of polynomial computations, stochastic models, image restoration, preconditioning, solving matrix equations. We also describe a recent application to compute the exponential of a block triangular block Toeplitz matrix.

Concerning quasi-separable matrices, we will recall the main properties, then we focus the attention to companion-like matrices and present some recent results concerning certain linearizations of matrix polynomials.

The spirit of this note is to give the flavor of the available results with pointers to the literature.

Notations. Throughout, given a field $\mathbb{F}$ and positive integers m, n we denote by $\mathbb{F}^{m\times n}$ the linear space of $m \times n$ matrices with entries in $\mathbb{F}$. The set $\mathbb{F}^{m\times 1}$ of m-dimensional column vectors is denoted more simply by $\mathbb{F}^m$. For our purposes $\mathbb{F}$ is limited to the real field $\mathbb{R}$ or to the complex field $\mathbb{C}$. The symbol $\underline{i}$ is the complex unit such that $\underline{i}^2 = -1$. We denote A^{T} the transpose matrix of A and A^* the transpose conjugate, where the conjugate of the complex number $z = a + \underline{i}b$ is $\overline{z} = a - \underline{i}b$ for $a, b \in \mathbb{R}$. The identity matrix is denoted by I; if the size n is not clear from the context we use the symbol I_n.

Moreover we use $\rho(A)$ for the spectral radius of the matrix A, i.e., the maximum modulus of the eigenvalues of A, while $\|x\|_\infty = \max|x_i|$, $\|x\|_2 = \sqrt{\sum |x_i|^2}$ are the infinity norm and the 2-norm, respectively. Given a vector norm $\|\cdot\|$ over $\mathbb{F}^n$ and a matrix $A \in \mathbb{F}^{n\times n}$ we denote by $\|A\|$ the matrix norm (operator norm) induced by the vector norm defined by $\|A\| = \sup_{x\neq 0} \frac{\|Ax\|}{\|x\|}$. The matrix operator norms induced by the vector norms $\|\cdot\|_\infty$ and $\|\cdot\|_2$ are given by $\|A\|_\infty = \max_i \sum_j |a_{i,j}|$, $\|A\|_2 = \sqrt{\rho(A^*A)}$, respectively. The Frobenius norm $\|A\|_F$ of a matrix A is defined as $\|A\|_F = (\sum_{i,j} |a_{i,j}|^2)^{\frac{1}{2}}$

We also denote by $\mathbb{T} = \{z \in \mathbb{C} : \quad |z| = 1\}$ the unit circle in the complex plane.

Given two matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ of size $m \times n$ and $p \times q$, respectively, we set $C = A \otimes B$ for the block matrix with blocks $a_{i,j}B$ having size $(mp) \times (nq)$. The symbol $\otimes$ is called the Kronecker product.

Finally, given a function $f : \mathbb{T} \to \mathbb{C}$, we define $\operatorname{ess\,sup} f(x)$ the sup of $f(x)$ up to a set of measure zero. The same we do with $\operatorname{ess\,inf} f(x)$.

CHAPTER 1

Toeplitz and Toeplitz-like matrices

1. Introduction

Given the $(2n+1)$-tuple $(a_{-n}, \ldots, a_0, \ldots, a_n)$, the matrix $T = (t_{i,j}) \in \mathbb{F}^{n\times n}$ such that $t_{i,j} = a_{j-i}$ for $i, j = 1, \ldots, n$ is said a *Toeplitz matrix*. A Toeplitz matrix has equal entries along its diagonals. An example of Toeplitz matrix is shown below

$$\begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_{-1} & a_0 & a_1 & a_2 \\ a_{-2} & a_{-1} & a_0 & a_1 \\ a_{-3} & a_{-2} & a_{-1} & a_0 \end{bmatrix}.$$

Given a formal Laurent series $a(z) = \sum_{i=-\infty}^{+\infty} a_i z^i$, we may associate with $a(z)$ the sequence of $n \times n$ Toeplitz matrices $\{T_n\}_n$ such that $T_n = (a_{j-i})_{i,j=0,n-1}$. This sequence is formed by the $n \times n$ leading principal submatrices of the semi-infinite matrix $T = (a_{j-i})_{i,j\in\mathbb{N}}$

The following classical result has been given by Otto Toeplitz (see Böttcher and Grudsky [26])

Proposition 1. *The semi-infinite matrix T defines a bounded operator in the space $\ell^2(\mathbb{N})$ of semi-infinite sequences over $\mathbb{F} = \mathbb{C}$ if and only if the numbers a_n are the Fourier coefficients of some function $a(z) : \mathbb{T} \to \mathbb{C}$ such that $a(z) \in L^\infty(\mathbb{T})$. The norm of the operator is*

$$\|T\|_\infty = \text{ess sup}_{z\in\mathbb{T}} |a(z)|.$$

The function $a(z)$ is called *symbol*.

1.1. Block matrices.

For the sake of notational simplicity sometimes it is convenient to represent matrices in terms of blocks. Given $m \times m$ matrices $A_{i,j} = (a_{r,s}^{(i,j)})$, $i, j = 1, \ldots, n$ we denote $A = (A_{i,j})$ the $mn \times mn$ matrix whose entries are given by $a_{(i-1)m+r,(j-1)m+s} = a_{r,s}^{(i,j)}$, where $a_{r,s}^{(i,j)}$ denotes the entry of $A^{(i,j)}$ in position (r, s). For instance,

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} = \left[\begin{array}{cc|cc} a_{1,1}^{(1,1)} & a_{1,2}^{(1,1)} & a_{1,1}^{(1,2)} & a_{1,2}^{(1,2)} \\ a_{2,1}^{(1,1)} & a_{2,2}^{(1,1)} & a_{2,1}^{(1,2)} & a_{2,2}^{(1,2)} \\ \hline a_{1,1}^{(2,1)} & a_{1,2}^{(2,1)} & a_{1,1}^{(2,2)} & a_{1,2}^{(2,2)} \\ a_{2,1}^{(2,1)} & a_{2,2}^{(2,1)} & a_{2,1}^{(2,2)} & a_{2,2}^{(2,2)} \end{array}\right].$$

We say that A is an $n \times n$ block matrix with $m \times m$ blocks. Block matrices are encountered in the mathematical models related to higher dimensional spaces.

Given the $(2n + 1)$-tuple $(A_{-n}, \ldots, A_0, \ldots, A_n)$ of $m \times m$ matrices the matrix $T \in \mathbb{F}^{n\times n}$, $T = (T_{i,j})$ such that $T_{i,j} = A_{j-i}$ for $i, j = 1, \ldots, n$ is said a *Block Toeplitz matrix*.

Similarly to the case of Toeplitz matrices, a matrix valued function $A(z) : \mathbb{T} \to \mathbb{C}^{m\times m}$ such that $A(z) = \sum_{i=-\infty}^{+\infty} A_i z^i$ is its Fourier series defines an infinite block Toeplitz matrix and a sequence $\{T_n\}$ of $n \times n$ block Toeplitz matrices with $m \times m$ blocks.

If the blocks A_k, $k \in \mathbb{Z}$ are Toeplitz matrices themselves, then the matrix $T = (A_{j-i})$ is said a *block Toeplitz matrix with Toeplitz blocks*. A function $a(z, w) : \mathbb{T} \times \mathbb{T} \to \mathbb{C}$ such that $a(z, w) = \sum_{r=-\infty}^{+\infty} \sum_{s=-\infty}^{+\infty} a_{r,s} z^r w^s$ is its Fourier series, generates a semi-infinite block Toeplitz matrix with semi-infinite blocks $T = (T_{j-i})_{i,j\in\mathbb{N}}$, $T_r = (a_{r,j-i})_{i,j\in\mathbb{N}}$.

For any pair of positive integers (m, n) the $n \times n$ block Toeplitz matrix with $m \times m$ blocks $T_{m,n}$ is derived from the symbol $a(z, w)$.

For instance, the celebrated discrete Laplacian matrix

$$L = -\begin{bmatrix} A & -I & & \\ -I & A & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & A \end{bmatrix}, \quad A = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix}$$

is a block Toeplitz matrix with Toeplitz blocks associated with the symbol $a(z,w) = -4 + z + z^{-1} + w + w^{-1}$.

Similarly, we may inductively define *multilevel Toeplitz matrices* formed by block Toeplitz matrices whose blocks are in turn multilevel block Toeplitz matrices. This kind of matrices are encountered in the mathematical modeling of multidimensional problems.

A function $a : \mathbb{T}^d \to \mathbb{C}$ having the Fourier expansion

$$a(x_1, x_2, \ldots, x_d) = \sum_{i_1,\ldots,i_d=-\infty}^{+\infty} a_{i_1,i_2,\ldots,i_d} x_{i_1}^{i_1} x_{i_2}^{i_2} \cdots x_{i_d}^{i_d}$$

defines a d-multilevel Toeplitz matrix: that is a block Toeplitz matrix with blocks that are themselves $(d-1)$-multilevel Toeplitz matrices.

1.2. **Banded matrices.** A matrix $A = (a_{i,j})$ is banded with bandwidth (h,k) if $a_{i,j} = 0$ for $j - i > k$ or $i - j > h$. A block matrix $A = (A_{i,j})$ is banded with bandwidth (h,k) if $A_{i,j} = 0$ for $j - i > k$ or for $i - j > h$.

Observe that if the symbol $a(z,w)$ is a Laurent polynomial, i.e., $a(z,w) = \sum_{r=-h}^{k} \sum_{s=-p}^{q} a_{r,s} z^r w^s$ for $h,k,p,q > 0$, then the associated block Toeplitz matrix with Toeplitz blocks is block banded with banded blocks.

1.3. **Toeplitz-like matrices.** Let L_i and U_i be lower triangular and upper triangular $n \times n$ Toeplitz matrices, respectively, where $i = 1, \ldots, k$ and k is independent of n then

$$A = \sum_{i=1}^{k} L_i U_i$$

is said *Toeplitz-like* matrix. In particular, if $k = 2$, $L_1 = I$, $U_2 = I$ the above expression provides a Toeplitz matrix.

2. Applications of Toeplitz matrices

In this section we briefly report on some applications of Toeplitz matrices. We deal with polynomial arithmetic, queuing models, image restoration and the numerical treatment of partial differential equations.

2.1. **Polynomial arithmetic.** Polynomial multiplication can be rephrased in terms of the product of a rectangular banded Toeplitz matrix and a vector as follows.

Let $a(x) = \sum_{i=0}^{n} a_i x^i$, $b(x) = \sum_{i=0}^{m} b_i x^i$, be two polynomials and let $c(x) := a(x)b(x)$, $c(x) = \sum_{i=0}^{m+n} c_i x^i$, be their product.

Relating the coefficients of $a(x), b(x)$ and $c(x)$ one finds that

$$\begin{aligned} c_0 &= a_0 b_0 \\ c_1 &= a_0 b_1 + a_1 b_0 \\ c_2 &= a_0 b_2 + a_1 b_1 + a_2 b_0 \\ &\ldots \end{aligned}$$

which can be rewritten in matrix form as

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ c_{m+n} \end{bmatrix} = \begin{bmatrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & \ddots & \ddots & \\ a_n & \ddots & \ddots & a_0 \\ & \ddots & \ddots & a_1 \\ & & \ddots & \vdots \\ & & & a_n \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

Similarly the computation of quotient and remainder of the division of two polynomials can be rewritten in the form of Toeplitz matrices.

If $a(x) = \sum_{i=0}^{n} a_i x^i$, $b(x) = \sum_{i=0}^{m} b_i x^i$, $b_m \neq 0$ are polynomials of degree n and m, respectively where $n \geq m$, denoting $q(x)$ the quotient of the division of $a(x)$ by $b(x)$ and $r(x)$ the remainder of this division, one has that the degree of $q(x)$ is $n-m$ while the degree of $r(x)$ is at most $m-1$.

Rewriting the relation $a(x) = b(x)q(x) + r(x)$ in matrix form yields

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \\ \vdots \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} b_0 & & & \\ b_1 & b_0 & & \\ \vdots & \ddots & \ddots & \\ b_m & \ddots & \ddots & b_0 \\ & \ddots & \ddots & b_1 \\ & & \ddots & \vdots \\ & & & b_m \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-m} \end{bmatrix} + \begin{bmatrix} r_0 \\ \vdots \\ r_{m-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Observe that the last $n-m+1$ equations form an upper triangular Toeplitz system

$$\begin{bmatrix} b_m & b_{m-1} & \dots & b_{2m-n} \\ & b_m & \ddots & \vdots \\ & & \ddots & b_{m-1} \\ & & & b_m \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-m} \end{bmatrix} = \begin{bmatrix} a_m \\ a_{m+1} \\ \vdots \\ a_n \end{bmatrix}.$$

Its solution provides the coefficients of the quotient.

The remainder can be computed as a difference:

$$\begin{bmatrix} r_0 \\ \vdots \\ r_{m-1} \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_{m-1} \end{bmatrix} - \begin{bmatrix} b_0 & & \\ \vdots & \ddots & \\ b_{m-1} & \dots & b_0 \end{bmatrix} \begin{bmatrix} q_0 \\ \vdots \\ q_{n-m} \end{bmatrix}.$$

For the sake of simplicity the above matrix has been written in the case where $n-m = m-1$

Also the polynomial greatest common divisor (gcd) can be easily rewritten in matrix form involving Toeplitz matrices. Recall that, by the Bézout identity, if $g(x) = \gcd(a(x), b(x))$, $\deg(g(x)) = k$, $\deg(a(x)) = n$, $\deg(b(x)) = m$ then there exist polynomials $r(x)$, $s(x)$ of degree at most $m-k-1$, $n-k-1$, respectively, such that

$$g(x) = a(x)r(x) + b(x)s(x).$$

Rewriting this expression by rephrasing polynomial products in terms of Toeplitz matrix-vector product, one finds the following $(m+n-k)\times(m+n-2k)$ system

$$\left[\begin{array}{cccc|cccc} a_0 & & & & b_0 & & & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ \vdots & \ddots & \ddots & & \vdots & \ddots & \ddots & \\ a_n & \ddots & \ddots & a_0 & b_m & \ddots & \ddots & b_0 \\ & \ddots & \ddots & a_1 & & \ddots & \ddots & b_1 \\ & & \ddots & \vdots & & & \ddots & \vdots \\ & & & a_n & & & & b_m \end{array}\right] \left[\begin{array}{c} r_0 \\ r_1 \\ \vdots \\ r_{m-k-1} \\ \hline s_0 \\ s_1 \\ \vdots \\ s_{n-k-1} \end{array}\right] = \begin{bmatrix} g_0 \\ \vdots \\ g_k \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}.$$

The matrix described in this system is called *Sylvester matrix*

The last $m+n-2k$ equations provide a linear system of the kind

$$S\begin{bmatrix} r \\ s \end{bmatrix} = \begin{bmatrix} g_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where S is the $(m+n-2k)\times(m+n-2k)$ submatrix formed by the first $m+n-2k$ rows of the above Sylvester matrix. This matrix is formed by two Toeplitz matrices.

2.2. **The Wiener-Hopf factorization: solving queueing problems.** Another interesting application concerns infinite Toeplitz matrices. Let $a(x), b(x)$ be polynomials of degree n, m with coefficients a_i and b_i, respectively. Define the Laurent polynomial, that is a polynomial in x and in x^{-1}, given by

$$c(x) = \sum_{i=-m}^{n} c_i x^i = a(x)b(x^{-1}).$$

Then one can verify that the following infinite UL factorization holds

$$\begin{bmatrix} c_0 & \dots & c_n & & \\ \vdots & c_0 & \ddots & c_n & \\ c_{-m} & \ddots & \ddots & \ddots & \ddots \\ & \ddots & \ddots & \ddots & \end{bmatrix} = \begin{bmatrix} a_0 & \dots & a_n & & \\ & a_0 & \ddots & a_n & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} b_m & & & & \\ b_{m-1} & b_m & & & \\ \vdots & \ddots & \ddots & & \\ b_0 & \ddots & \ddots & \ddots & \\ & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

where the first factor U is an infinite upper triangular Toeplitz matrix and the second factor L is an infinite lower triangular Toeplitz matrix.

If the zeros of $a(x)$ and of $b(x)$ lie in the unit disk this factorization is called *Wiener-Hopf factorization*. Wiener-Hopf factorizations are encountered in many applications, see for instance [52], [17].

The Wiener-Hopf factorization can be defined for matrix-valued functions $C(x) = \sum_{i=-\infty}^{+\infty} C_i x^i$, $C_i \in \mathbb{C}^{m\times m}$, belonging to the Wiener class, i.e, such that $\sum_{i=-\infty}^{+\infty} \|C_i\| < \infty$, provided that $\det C(x) \neq 0$ for $|x| = 1$ [27].

A *canonical Wiener-Hopf factorization* takes the form

$$C(x) = A(x)B(x^{-1}), \quad A(x) = \sum_{i=0}^{\infty} x^i A_i, \ B(x) = \sum_{i=0}^{\infty} B_i x^i$$

where $A(x)$ and $B(x)$ are invertible for $|x| \geq 1$.

Its matrix representation provides a block UL factorization of the infinite block Toeplitz matrix (C_{j-i})

$$\begin{bmatrix} C_0 & C_1 & \dots & \\ C_{-1} & C_0 & C_1 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} A_0 & A_1 & \dots & \\ & A_0 & A_1 & \ddots \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} B_0 & & & \\ B_{-1} & B_0 & & \\ \vdots & \ddots & \ddots & \end{bmatrix}.$$

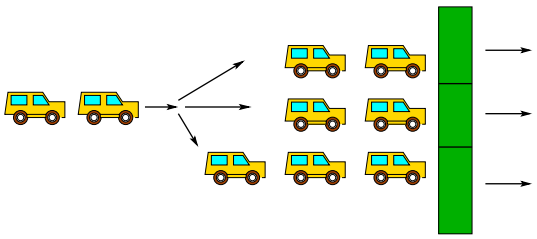

Figure 2.1: The shortest queue problem

For a concise survey on Wiener-Hopf factorizations we refer the reader to the paper [28] by Böttcher and Spitkovsky.

Computing canonical Wiener-Hopf factorizations is fundamental in the solution of many queuing models [17]. An example of application to queueing model is given by the shortest queue problem.

The problem: There are m gates at an exit of a highway. Cars arrive, join a line, pay the toll and exit the highway according to the following rules

- at each instant, k cars arrive with a known probability;
- each car follows the shortest line;
- at each instant, a car leaves the gate.

The question is to determine the probability π_ℓ that after a long time there is an overall number of ℓ cars in the lines waiting to be served. This problem is close to the important model which defines the wireless IEEE 802.11 protocol used in the wireless connections of a computer network.

Denoting $p_{i,j}$ the probability that after one instant of time the overall length of the queue changes from i to j then $p_{i,j} = a_{j-i}$ if $i \geq m$, that is if there are at least m cars in the queue, where a_k is the probability that $k+m$ cars arrive so that $0 \leq a_k \leq 1$, $\sum_{k=-m}^{\infty} a_k = 1$, $a_k = 0$ for $k < -m$.

Thus the problem turns into an infinite eigenvalue problem of the kind

$$\pi^T P = \pi^T,$$

$\pi \in \mathbb{R}^{\mathbb{N}}$ is a probability vector, i.e., $\sum \pi_i = 1$, $\pi_i \geq 0$ is the probability that there are i cars waiting in the lines, and $P = (p_{i,j})$ is almost Toeplitz in generalized upper Hessenberg form

$$P = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots \\ b_{m,1} & b_{m,2} & \dots & \dots \\ a_0 & a_1 & a_2 & \dots \\ & a_0 & a_1 & \ddots \\ & & \ddots & \ddots \end{bmatrix},$$

where $b_{i,j}$ are suitable boundary probabilities. This matrix can be partitioned into $m \times m$ blocks as follows

$$P = \begin{bmatrix} B_0 & B_1 & B_2 & \dots & \\ A_{-1} & A_0 & A_1 & \ddots & \ddots \\ 0 & A_{-1} & A_0 & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

Removing the first block row and the first block column of the above matrix yields the block Hessenberg block Toeplitz matrix

$$\widehat{P} = \begin{bmatrix} A_0 & A_1 & A_2 & \dots & \\ A_{-1} & A_0 & A_1 & \ddots & \ddots \\ 0 & A_{-1} & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}.$$

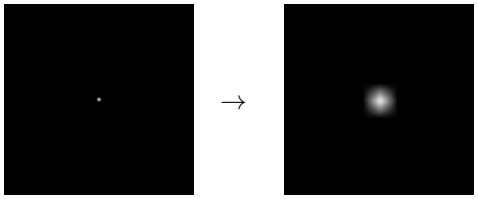

Figure 2.2: The Point Spread Function defines the way a point is blurred.

The Wiener-Hopf factorization of $\widehat{P} - I$ allows to solve easily the problem $\pi(P - I) = 0$ where π in turn is partitioned in subvectors $\pi^{(0)}, \pi^{(1)}, \ldots$, of length m. In fact, assuming that $\pi^{(0)}$ is known, then the computation of $\pi^{(i)}$ for $i > 0$ is reduced to solving two block triangular Toeplitz systems. The computation of $\pi^{(0)}$ can be performed by using suitable formulas [17].

The Wiener-Hopf factorization of $\widehat{P} - I$ takes the following form

$$\widehat{P} - I = \begin{bmatrix} U_0 & U_1 & \ldots & \\ & U_0 & U_1 & \ddots \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} I & & & \\ -G & I & & \\ & -G & I & \\ & & \ddots & \ddots \end{bmatrix},$$

where G is the solution of the following matrix equation

$$X = \sum_{i=-1}^{+\infty} A_i X^i$$

having nonnegative entries and spectral radius $\rho(G) = 1$.

A way for solving this equation is to reduce it to the following infinite linear block Toeplitz system

$$\begin{bmatrix} A_0 & A_1 & A_2 & \ldots \\ A_{-1} & A_0 & A_1 & \ldots \\ & A_{-1} & A_0 & \ldots \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} G \\ G^2 \\ G^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} -A_{-1} \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

2.3. **Image restoration.** Another interesting application concerns blurring and deblurring models in digital image restoration. Here one assumes that the blur of a single point of an image is independent of the position of the point, that is, it is *shift invariant*, and is defined by the *Point-Spread Function (PSF)*. This function has compact support, in fact, a point is blurred into a small spot of light with dark (null value of the function) everywhere except that in a small neighborhood of the point itself. An example of PSF is reported in Figure 2.2

The relation between the blurred and noisy image, stored as a vector b and the real image, represented by a vector x, has the form

$$Tx = b - \text{noise}.$$

Due to the shift invariance of the PSF, T is block Toeplitz with Toeplitz blocks. Due to the local effect of the blur, the PSF has compact support so that T is block banded with banded blocks.

Typically, T is ill-conditioned so that solving the system $Tx = b$ obtained by ignoring the noise provides a highly perturbed solution.

For instance the PSF which transforms a unit point of light into the 3×3 square $\frac{1}{15}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ leads to the following block Toeplitz matrix

$$T = \frac{1}{15} \begin{bmatrix} B & A & & & \\ A & B & A & & \\ & \ddots & \ddots & \ddots & \\ & & A & B & A \\ & & & A & B \end{bmatrix},$$

where

$$A = \begin{bmatrix} 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & 2 & 1 \\ & & & 1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 2 & & & \\ 2 & 3 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & 3 & 2 \\ & & & 2 & 3 \end{bmatrix}.$$

This way, restoring a blurred image is reduced to solving a block banded block Toeplitz system with banded Toeplitz blocks. According to the boundary conditions assumed in the blurring model, the matrix can take additional specific structures.

2.4. **Differential equations.** The numerical treatment of linear partial differential equations with constant coefficients by means of the finite difference technique leads to linear systems where the matrix is block Toeplitz with Toeplitz blocks.

For instance the discretization of the Laplace operator $\Delta u(x, y)$ applied to a function $u(x, y)$ defined over $[0, 1] \times [0, 1]$

$$\Delta u(x, y) = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{1}{h^2}(4u_{i,j} - u_{i+1,j} - u_{i-1,j} - u_{i,j+1}, -u_{i,j-1}) + O(h^2)$$

for $x_i = ih$, $y_j = jh$, $i, j = 1, n$, $h = 1/(n + 1)$, leads to the matrix

$$L = -\frac{1}{h^2}\begin{bmatrix} A & -I & & \\ -I & A & \ddots & \\ & \ddots & \ddots & -I \\ & & -I & A \end{bmatrix}, \quad A = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{bmatrix},$$

that is a block Toeplitz matrix with Toeplitz blocks associated with the symbol $a(z, w) = -\frac{1}{h^2}(4 - z - z^{-1} - w - w^{-1})$.

Similarly, the discretization of the three dimensional Laplacian $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}$ leads to a block tridiagonal block Toeplitz matrix whose diagonal blocks are block tridiagonal block Toeplitz matrices with tridiagonal blocks.

3. Asymptotic spectral properties

Here we consider spectral properties of real symmetric (or complex Hermitian) matrices defined by a symbol $a(z) : \mathbb{T} \to \mathbb{F}$. For the sake of simplicity denote $z = z(\theta) = \cos\theta + \underline{i}\sin\theta \in \mathbb{T}$ so that we can view the symbol $a(z)$ as a periodic function of θ defined over $[0, 2\pi]$ by $\theta \to a(z(\theta))$. With abuse of notation, we write $a(\theta)$ for $a(z(\theta))$.

Let $f(x) : [0, 2\pi] \to \mathbb{R}$ be a Lebesgue integrable function. We say that a sequence of sequences $\{\lambda_i^{(n)}\}_{i=1,n}$, $\lambda_i^{(n)} \in \mathbb{R}$, $n \in \mathbb{N}$, is distributed as $f(x)$ if

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} F(\lambda_i^{(n)}) = \frac{1}{2\pi}\int_0^{2\pi} F(f(x))dx$$

for any continuous function $F(x)$ with bounded support.

As an example, consider the sequence formed by the values of $f(x)$ sampled at equally spaced points in $[0, 2\pi]$ given by $\lambda_i^{(n)} = f(2i\pi/n)$, $i = 1, \ldots, n$, and let $n \in \mathbb{N}$. Then it is easy to see that $\{\lambda_i^{(n)}\}$ is distributed as $f(x)$.

Consider a real valued symbol $a(\theta) : [0 : 2\pi] \to \mathbb{R}$ so that $a(\theta) = a_0 + 2\sum_{k=1}^{\infty} a_k \cos k\theta$, and let T_n be the sequence of Toeplitz matrices associated with $a(\theta)$, that is, $T_n = (a_{|j-i|})_{i,j=1,n}$. Observe that T_n is real symmetric so that it has n real eigenvalues $\lambda_1^{(n)} \le \lambda_2^{(n)} \le \cdots \le \lambda_n^{(n)}$.

The following result was given by Szegő [53] for functions in L^∞ and extended by Tyrtyshnikov and Zamarashkin in [75] to functions in $L^1([0, 2\pi])$.

Proposition 2. *Let the symbol $a(\theta)$ be a real valued function belonging to $L^1([0, 2\pi])$, and let T_n be the sequence of Toeplitz matrices associated with $a(\theta)$. Denote $m_a = \operatorname{ess\,inf}_{\theta\in[0,2\pi]} a(\theta)$, $M_a = \operatorname{ess\,sup}_{\theta\in[0,2\pi]} a(\theta)$. Then for the eigenvalues $\lambda_i^{(n)}$ of T_n we have*

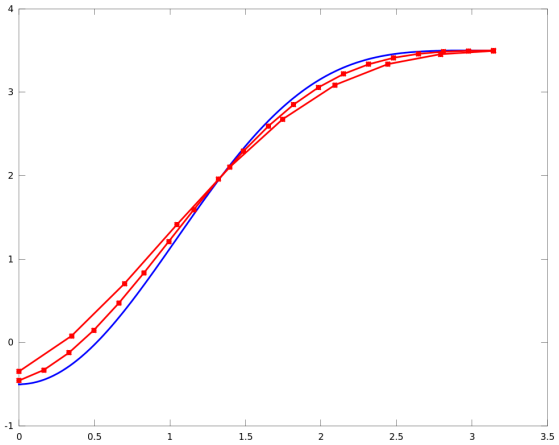

Figure 3.1: In this figure, the eigenvalues of the Toeplitz matrix T_n associated with the symbol $f(\theta) = 2 - 2\cos\theta - \frac{1}{2}\cos(2\theta)$ have been plotted (in red) for $n = 10$, $n = 20$, together with the graph of the symbol (in blue). One can see that as n grows, the values $\lambda_i^{(n)}$ for $i = 1, \ldots, n$ tend to be shaped as the graph of the symbol.

(1) *if* $n_a < M_a$ *then* $\lambda_i^{(n)} \in (m_a, M_a)$ *for any* n *and* $i = 1, \ldots, n$*; if* $m_a = M_a$ *then* $a(x)$ *is constant and* $T_n = m_a I_n$*;*
(2) $\lim_{n\to\infty} \lambda_1^{(n)} = m_a$*,* $\lim_{n\to\infty} \lambda_n^{(n)} = M_a$*;*
(3) *the eigenvalues sequence* $\{\lambda_1^{(n)}, \ldots, \lambda_n^{(n)}\}$ *are distributed as* $a(\theta)$*.*

A suggestive interpretation of the above result is as follows: a suitable ordering of the eigenvalues $\lambda_j^{(n)}$, $j = 1, \ldots, n$, can be seen as an approximation of the function $f(x)$ sampled on an equispaced grid on the domain $[0, 2\pi)$. Therefore, it is evident that the symbol $f(x)$ provides information on the definiteness of the matrices of the matrix sequence, on their inertia (number of positive and negative eigenvalues), on their conditioning.

In fact, as a consequence of the above results we find that

: – $\operatorname{ess\,inf} a(x) \geq 0$ iff T_n is positive definite for all $n \in \mathbb{N}$,
: – if $a(x) \geq 0$ the condition number $\mu^{(n)} = \|T^{(n)}\|_2 \|(T^{(n)})^{-1}\|_2$ of $T^{(n)}$ is such that $\lim_{n\to\infty} \mu^{(n)} = \operatorname{ess\,sup}_{\theta\in[0,2\pi]} a(\theta) / \operatorname{ess\,inf}_{\theta\in[0,2\pi]} a(\theta)$,
: – $a(\theta) > 0$ implies that $T^{(n)}$ is uniformly well conditioned,
: – $a(\theta) = 0$ for some θ implies that $\lim_{n\to\infty} \mu_n = \infty$.

A similar property (Avram-Parter theorem [6], [63]) holds for singular values. The requirement that $f(x)$ is in L^∞ given in the original result of [63] has been relaxed by Tyrtyshnikov [74].

Proposition 3. *If the symbol* $a(\theta)$ *is* L^∞ *then the singular values of the complex matrices* T_n *are distributed as* $|a(\theta)|$*.*

The same asymptotic property holds true for

: – block Toeplitz matrices generated by a matrix valued symbol $A(\theta)$,
: – block Toeplitz matrices with Toeplitz blocks generated by a bivariate symbol $a(\theta_1, \theta_2)$,
: – multilevel block Toeplitz matrices generated by a multivariate symbol $a(\theta_1, \theta_2, \ldots, \theta_d)$,
: – singular values of any of the above matrix classes.

In particular the following result holds

Proposition 4. *Let the symbol* $a(\theta) : [0 : 2\pi]^d \to \mathbb{R}$*,* $\theta = (\theta_1, \ldots, \theta_d)$ *be a real valued function belonging to* $L^1([0, 2\pi]^d)$ *and let* T_n *be the sequence of multilevel Toeplitz matrices of size* $n := (n_1, n_2, \ldots, n_d)$ *associated with* $a(\theta)$*. Denote* $m_a = \operatorname{ess\,inf} a(x)$*,* $M_a = \operatorname{ess\,sup} a(x)$ *and denote* $\lambda_i^{(n)}$*,* $i = 1, \ldots, N(n) = \prod_{i=1}^d n_i$ *the eigenvalues of* T_n *sorted in nondecreasing order. Then*

(1) T_n *is real symmetric;*

(2) *if* $n_a < M_a$ *then* $\lambda_i^{(n)} \in (m_a, M_a)$ *for any* $(n_1, \ldots, n_d)$ *and* $i = 1, \ldots, N(n)$*; if* $m_a = M_a$ *then* $a(\theta)$ *is constant and* $T_n = m_a I_{N(n)}$*;*

(3) $\lim_{n\to\infty} \lambda_1^{(n)} = m_a$, $\lim_{n\to\infty} \lambda_n^{(n)} = M_a$*, where* $n \to \infty$ *means that* $n_i \to \infty$ *for* $i = 1, \ldots, d$*;*

(4) *the eigenvalues* $\lambda_1^{(n)}, \ldots, \lambda_{N(n)}^{(n)}$ *are distributed as* $a(\theta)$ *in the sense that*

$$\lim_{n\to\infty} \frac{1}{N(n)} \sum_{j=1}^{N(n)} F(\lambda_j^{(n)}) = \frac{1}{(2\pi)^d} \int_0^{2\pi} \cdots \int_0^{2\pi} F(a(\theta_1, \ldots, \theta_d)) d\theta_1 \cdots d\theta_d,$$

for any continuous real function $F(x)$ *with bounded support.*

The same results hold true for the product $A_n = P_n^{-1} T_n$ where T_n and P_n are multilevel Toeplitz matrices of size $n = (n_1, \ldots, n_d)$ associated with symbols $a(\theta)$, $p(\theta)$, respectively.

In particular the following result holds

Proposition 5. *Let* $a(x)$ *and* $p(x)$ *be integrable functions in* $[0, 2\pi]^d$*, assume that* $p(x)$ *is non-negative and not identically zero. Let* $n = (n_1, \ldots, n_d)$*,* $N(n) = \prod_{i=1}^d n_i$*. Consider the sequence of multilevel Toeplitz matrices* T_n *and* P_n *associated with* $a(x)$ *and* $p(x)$*, respectively, and denote* $\lambda_j^{(n)}$ *the eigenvalues of* $G_n = P_n^{-1} T_n$ *sorted in non decreasing order. Let* r *and* R *be the essential infimum and supremum of* $a(x)/p(x)$*, respectively. Then*

(1) *for* $r < R$*,* $\lambda_j^{(n)} \in (r, R)$ *for any* n *and* $j = 1, \ldots, N(n)$*; for* $r = R$*,* $G_n = I_{N(n)}$*;*

(2) $\lim_{n\to\infty} \lambda_1^{(n)} = r$, $\lim_{n\to\infty} \lambda_N^{(n)} = R$*;*

(3) *the eigenvalues* $\lambda_1^{(n)}, \ldots, \lambda_N^{(n)}$ *are distributed as* $a(\theta)/g(\theta)$ *in the sense that*

$$\lim_{n\to\infty} \frac{1}{N(n)} \sum_{j=1}^{N(n)} F(\lambda_j^{(n)}) = \frac{1}{(2\pi)^d} \int_0^{2\pi} \cdots \int_0^{2\pi} F(\frac{a(\theta)}{p(\theta)}) d\theta_1 \cdots d\theta_d.$$

This property is crucial to reduce the condition number of Toeplitz systems by means of the preconditioning technique. More precisely, given $a(x) \geq 0$ such that $a(\theta_0) = 0$ for some θ_0, then as we have already pointed out, the condition number grows to infinity as $n \to \infty$. However, if there exists a trigonometric polynomial $p(\theta) = p_0 + 2\sum_{i=1}^k p_k \cos(k\theta)$ such that $p(\theta) \geq 0$, $p(\theta_0) = 0$ and $\lim_{\theta\to\theta_0} a(\theta)/p(\theta) \neq 0$ then the function $f(\theta) = a(\theta)/p(\theta)$ is such that $\operatorname{ess\,inf} a(x) > 0$. This way, denoting P_n the band Toeplitz matrix associated with the symbol $p(\theta)$, it follows that the matrix $P_n^{-1} T_n$ has condition number uniformly bounded from above by a constant independent of n. This way the system $Ax = b$ which is ill conditioned, can be replaced by the equivalent preconditioned system $P_n^{-1} A x = P_n^{-1} b$ which is better conditioned.

Further properties concerning the behavior of the extreme eigenvalues can be proved.

4. Some matrix algebras

From the computational point of view, Toeplitz and block Toeplitz matrices are easily manipulated by relying on certain matrix algebras and on their associated trigonometric transforms. We recall the main trigonometric transform related to Toeplitz computations.

4.1. **Trigonometric transform.**

In the framework of trigonometric transforms the lion role is played by the Discrete Fourier Transform (DFT) and by the fast algorithms for its computation known as Fast Fourier Transform (FFT) algorithms.

4.1.1. *Discrete Fourier Transform.* Let $\omega_n = \cos\frac{2\pi}{n} + \underline{i} \sin\frac{2\pi}{n}$ be a primitive nth root of 1, that is, such that $\omega_n^n = 1$ and $\{1, \omega_n, \ldots, \omega_n^{n-1}\}$ has cardinality n. Define the $n \times n$ matrix $\Omega_n = (\omega_n^{ij})_{i,j=0,n-1}$, $F_n = \frac{1}{\sqrt{n}}\Omega$. From the properties of the nth roots, one can easily deduce that $F_n^* F_n = I$ that is, F_n is a unitary matrix.

We say that y is the DFT of a vector $x \in \mathbb{C}^n$ and write $y = \mathrm{DFT}(x)$ if $y = \frac{1}{n}\Omega_n^* x$. Since $\Omega_n^{-1} = \frac{1}{n}\Omega_n^*$, we also write $x = \mathrm{IDFT}(y) = \Omega y$ for the Inverse Discrete Fourier Transform.

Since the matrix F_n is unitary then $\|F_n\|_2 = \|F_n^*\|_2 = 1$, that is, the condition number $\mathrm{cond}_2(F_n) = \|F_n\|_2 \|F_n^{-1}\|_2$ of F_n is 1. This shows that the DFT and IDFT are numerically well conditioned when the perturbation errors are measured in the 2-norm.

If n is an integer power of 2 then the IDFT of a vector can be computed with the cost of $\frac{3}{2}n\log_2 n$ arithmetic operations (ops) by using, say, the Cooley-Tukey or the Sandey-Tukey FFT algorithm. A similar complexity bound holds for the DFT.

The Cooley-Tukey algorithm is backward numerically stable in the 2-norm. That is, for $x = \mathrm{IDFT}(y)$, if $\widetilde{x}$ is the vector obtained by performing the Cooley-Tukey algorithm in floating point arithmetic with precision μ, then $\|x-\widetilde{x}\|_2 \le \mu\gamma\|x\|_2\log_2 n$ for a moderate constant γ [54].

The norm-wise well conditioning of DFT and the norm-wise stability of FFT algorithms make this tool very effective for most numerical computations.

Unfortunately, the *norm-wise stability* of FFT does not imply the *component-wise stability*. That is, the inequality $|x_i-\widetilde{x}_i| \le \mu\gamma|x_i|\log_2 n$ is not generally true for all the components x_i. In particular, when the moduli $|x_i|$ of the components are very unbalanced, the inequality is not satisfied by the components with the smallest modulus. In particular, if x_i is such that $|x_i| < \mu\|x\|_2$, the relative error in x_i can grow over 100%.

This is a drawback of DFT and of FFT when numerically used for symbolic computations since, in order to guarantee a sufficiently accurate relative precision in the result, one has to choose a suitable value of the machine precision of the floating point arithmetic. This value typically depends on the ratio between the maximum and the minimum absolute value of the output. This fact implies that the complexity bounds are depending on this ratio. When using FFT in this framework one should be aware of this fact.

Remark 1 (An example of misuse of FFT)**.** Let $p(x) = \sum_{i=0}^n p_i x^i$ be a polynomial of degree n with zeros x_i, $i = 1,\dots,n$ such that

$$|x_1| < \cdots < |x_m| < 1 < |x_{m+1}| < \cdots < |x_n|.$$

Set $p^{(0)}(x) := p(x)$ and define the sequence

$$q(x^2) = p^{(k)}(x)p^{(k)}(-x), \quad p^{(k+1)}(x) = q(x)/q_m, \quad \text{for } k = 0,1,2,,\dots$$

The iteration generating this sequence is known as the *Graeffe iteration*. It was introduced independently by Lobachevsky, Dandelin and by Graeffe. See the papers [61], [62] by Ostrowski.

Clearly, the zeros of $p^{(k)}(x)$ are $x_i^{2^k}$, so that $\lim_{k\to\infty} x_i^{2^k} = 0$ for $i = 1,\dots,m$, while $\lim_{k\to\infty} x_i^{2^k} = \infty$ for $i = m+1,\dots,n$, that is, $\lim_{k\to\infty} p^{(k)}(x) = x^m$. Moreover, denoting $p_i^{(k)}$ the coefficients of $p^{(k)}(x)$, that is, $p^{(k)}(x) = \sum_{i=0}^n p_i^{(k)} x^i$, we have

$$\lim_{k\to\infty} |p_{n-1}^{(k)}/p_n^{(k)}|^{1/2^k} = |x_n|$$

where convergence is very fast since the approximation error converges double exponentially to zero. Similar equations hold for approximating the moduli of the other zeros $|x_i|$. In fact, the Graeffe iteration is a useful tool for computing the moduli of the zeros of a polynomial $p(x)$.

Now, observe that for $m < n-1$ one has

$$\lim_{k\to\infty} |p_{n-1}^{(k)}| = \lim_{k\to\infty} |p_n^{(k)}| = 0$$

with double exponential convergence.

Computing $p^{(k)}(x)$ given $p^{(k-1)}(x)$ by using FFT, that is, by means of evaluation interpolation at the roots of unity, costs $O(n\log n)$ ops. But as soon as $|p_n^{(k)}|$ and $|p_{n-1}^{(k)}|$ are below the machine precision the relative errors in these two coefficients can be greater than 1. That is, it is very likely that **no digit is correct** in the computed estimate of $|x_n|$. This fact is clearly shown in Figure 1 where the values of $\log_{10}|p_i^{(6)}|$ for $i = 0,\dots,n$ are reported for a random polynomial $p^{(0)}(x)$ of degree 100. The line in red describes the values obtained by applying FFT while the line in blue represents the values computed with the customary algorithm for polynomial multiplication.

This features is particularly evident at the border of the interval $[0,n]$ when the moduli of some coefficients (in blue) go beyond the machine precision $\epsilon \approx 2.2\times 10^{-16}$, the values computed by means of FFT (in red) have moduli which keep oscillating around ϵ. In the center of the interval, where the coefficients have modulus still greater than ϵ, the red and the blue curves overlap.

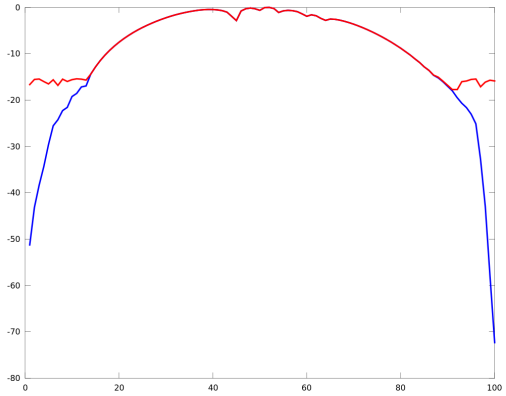

Figure 4.1: The values of $\log_{10}|p_i^{(6)}|$ for $i = 0, \ldots, n$ for the polynomial obtained after 6 Graeffe steps starting from a random polynomial of degree 100. In red the case where the coefficients are computed with FFT, in blue the coefficients computed with the customary algorithm.

step	custom	FFT
1	1.40235695	1.40235695
2	2.07798429	2.07798429
3	2.01615072	2.01615072
4	2.01971626	2.01857621
5	2.01971854	1.00375471
6	2.01971854	0.99877589

Table 4.1: Approximations to $|x_1|$ obtained by using the Graeffe iteration where polynomial multiplication is performed by means of FFT (third column) and by means the customary method (second column).

In Table 1 we report the approximations to $|x_1|$ obtained by applying the Graeffe iteration implemented by means of FFT and by means of the customary algorithm for polynomial multiplication. One can see that as soon as convergence starts to be numerically evident, the errors generated by the floating point arithmetic grow larger for the method based on FFT.

A specific analysis shows that in order to have d correct digits in the computed approximation, one must use a floating point arithmetic with c digits, where

$$c = d * \left(1 + \gamma \frac{\log(|x_n|/|x_1|)}{\log(|x_n|/|x_{n-1}|)}\right), \quad \gamma > 1.$$

Problems are encountered if $|x_n| \approx |x_{n-1}|$ or $|x_n/x_1|$ is large. In the situation where the separation from two consecutive zeros is uniform, i.e., $|x_{i+1}/x_i| = |x_n/x_1|^{1/n}$ then the number of digits is $c = d * (1 + \gamma n)$. Clearly, performing $O(n \log n)$ ops with $O(nd)$ digits is more expensive than performing $O(n^2)$ ops with d digits. □

There are algorithms for computing the DFT in $O(n \log n)$ ops whatever is the value of n.

The DFT and FFT can be defined over finite fields where there exists a primitive root of 1. For instance, $\mathbb{Z}_{17}$ is a finite field and 3 is a primitive 16th root of 1. DFT and FFT can be defined over certain rings.

Working with block matrices, it is useful to define the DFT and the IDFT of block vectors. If $Y_1, \ldots, Y_n$ are $m \times m$ matrices, denoting $y = (Y_i)$ the $nm \times m$ matrix (block column vector) formed by the blocks Y_i, we write $x = \mathrm{IDFT}(y)$ for $x = (X_i)$, $X_i = \sum_{j=0}^{n-1} \omega_n^{ij} Y_j$.

By using the Kronecker product notation, one has $\mathrm{IDFT}(y) = (\Omega_n \otimes I_m)x$. Similarly, one defines the DFT.

4.1.2. *The sine transform.* The $n \times n$ matrix $S = (\sqrt{\frac{2}{n+1}} \sin \frac{\pi ij}{n+1})$ is orthogonal, that is $S^{\mathrm{T}} S = I$. The sine transform $x \to y = Sx$ can be computed in $O(n \log n)$ ops if $n+1$ is an integer power of 2. There are 8 different types of sine transforms, see [39].

4.1.3. *The cosine transforms.* The $n \times n$ matrix $C = (\sqrt{\frac{2}{n}} \cos \frac{(2i+1)(2j+1)\pi}{4n})$ is orthogonal and defines the cosine transform $x \to y = Cx$. There are 8 versions of cosine transforms which can be computed with fast algorithms in $O(n \log n)$ ops. For more details see [39].

4.1.4. *The Hartley transform.* The matrix $H = \frac{1}{\sqrt{n}}(\cos \frac{2\pi ij}{n} + \sin \frac{2\pi ij}{n})_{i,j=0,n-1}$ is orthogonal, the application $x \to y = Hx$ is called *Hartley transform*. The Hartley transform can be computed with $O(n \log n)$ ops. For more details see [13].

4.2. **Circulant matrices and FFT.** Given the row vector $[a_0, a_1, \ldots, a_{n-1}]$, the $n \times n$ matrix

$$A = (a_{j-i \mod n})_{i,j=1,n} = \begin{bmatrix} a_0 & a_1 & \ldots & a_{n-1} \\ a_{n-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ a_1 & \ldots & a_{n-1} & a_0 \end{bmatrix}$$

is called *circulant* matrix associated with $[a_0, a_1, \ldots, a_{n-1}]$ and is denoted by $\mathrm{Circ}(a_0, a_1, \ldots, a_{n-1})$.

A circulant matrix is fully defined by its first row $r^{\mathrm{T}} = [a_0, a_1, \ldots, a_{n-1}]$ or its first column $c = [a_0, a_{n-1}, \ldots, a_1]^{\mathrm{T}}$. Any other row or column is obtained from the preceding one by applying a cyclic permutation to its elements: the last element is moved to the first position and the remaining ones are shifted by one position. With S denoting the circulant matrix associated with $[0, 1, 0, \ldots, 0]$, i.e.,

$$S = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & 1 \\ 1 & 0 & \ldots & 0 & 0 \end{bmatrix}, \tag{4.1}$$

it can easily be verified that

$$A = \sum_{i=0}^{n-1} a_i S^i. \tag{4.2}$$

That is, any circulant matrix can be viewed as a polynomial in S.

Given polynomials $a(x) = \sum_{i=0}^{n-1} a_i x^i$, $b(x) = \sum_{i=0}^{n-1} b_i x^i$, define the circulant matrices $A = a(S)$, $B = b(S)$ and set $C = AB$. Since the minimal polynomial of S is $x^n - 1$, one finds that $C = c(S)$ where $c(x) = a(x)b(x) \mod x^n - 1$. That is, circulant matrices form a commutative matrix algebra isomorphic to the set of polynomials of degree less than n with the product modulo $x^n - 1$.

By direct inspection one finds that

$$S\overline{\Omega}_n = \overline{\Omega}_n \,\mathrm{Diag}(1, \overline{\omega}_n, \overline{\omega}_n^2, \ldots, \overline{\omega}_n^{n-1});$$

multiplying the latter expression on the left by $\frac{1}{n}\Omega_n$ yields

$$\frac{1}{n}\Omega_n S\overline{\Omega}_n = \mathrm{Diag}(1, \overline{\omega}_n, \overline{\omega}_n^2, \ldots, \overline{\omega}_n^{n-1});$$

moreover, taking the conjugate transpose of both sides, we find

$$\frac{1}{n}\Omega_n S^{\mathrm{T}}\overline{\Omega}_n = \mathrm{Diag}(1, \omega_n, \omega_n^2, \ldots, \omega_n^{n-1}),$$

since Ω_n is symmetric. From the above two equations and (4.2) we deduce the following property

Proposition 6. *If A is a circulant matrix with first row r^{T} and first column c, then*

$$A = \frac{1}{n}\overline{\Omega}_n \operatorname{Diag}(w)\Omega_n,$$

where $w = \Omega_n c = \overline{\Omega}_n r$.

An immediate corollary of the theorem above is that we can compute the product Ax of an $n \times n$ circulant matrix A and a vector x by means of two IDFTs of length n and a DFT. In fact, the above result can be rephrased in the form

$$Ax = \mathrm{DFT}_n(\mathrm{IDFT}_n(c) * \mathrm{IDFT}_n(x)), \tag{4.3}$$

where "$*$" denotes the Hadamard, or component-wise product of vectors.

From Proposition 6 we also find that the product of two circulant matrices is still circulant and the inverse of a nonsingular circulant matrix is circulant.

The definition of circulant matrix is naturally extended to block matrices.

Definition 1. *Given the block row vector $[A_0, A_1, \ldots, A_{n-1}]$ where A_i, $i = 0, \ldots, n-1$, are $m \times m$ matrices, the $n \times n$ block matrix $A = \mathrm{Circ}(A_0, A_1, , \ldots, A_{n-1})$*

$$A = (A_{j-i \mod n})_{i,j=1,n} = \begin{bmatrix} A_0 & A_1 & \ldots & A_{n-1} \\ A_{n-1} & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_1 \\ A_1 & \ldots & A_{n-1} & A_0 \end{bmatrix},$$

is called block circulant *matrix associated with $[A_0, A_1, \ldots, A_{n-1}]$.*

Similarly to the scalar case we have

$$A = \sum_{i=0}^{n-1} S^i \otimes A_i, \tag{4.4}$$

and Proposition 6 is generalized to the following property

Proposition 7. *If A is a block circulant matrix with first block row r^{T} and with first block column c we have*

$$A = \frac{1}{n}(\overline{\Omega}_n \otimes I_m) \operatorname{Diag}(W_1, \ldots, W_n)(\Omega_n \otimes I_m)$$

where

$$[W_1, \ldots, W_n] = r^{\mathrm{T}}(\overline{\Omega}_n \otimes I_m),$$
$$\begin{bmatrix} W_1 \\ \vdots \\ W_n \end{bmatrix} = (\Omega_n \otimes I_m)c.$$

Like circulant matrices, the class of block-circulant matrices is closed under matrix multiplication and inversion.

Equation (4.3) becomes

$$Ax = \mathrm{DFT}_n(\mathrm{IDFT}_n(c) * \mathrm{IDFT}_n(x)) \tag{4.5}$$

where c is the first block column of A, which shows that one can compute the product Ax of an $n \times n$ block circulant matrix A and a block vector x with block components of size $m \times p$ by means of two block IDFTs and one block DFT of length n, and n products of $m \times m$ times $m \times p$ matrices.

We synthesize (4.3) and (4.5) with Algorithm 1 for multiplying a block circulant matrix and a block vector. For $m = p = 1$, the algorithm reduces to the scalar case.

The cost of computing y, given x and c is clearly $O((m+p)mn \log n + nm^2 p)$ ops. If the elements of c are real, then it follows that W_1 and $W_{n/2+1}$ are real and that $W_i = \overline{W}_{n-i+2}$, $i = 2, \ldots, n/2$. The same property holds for the block components V_i of v if x is real. Thus, if both c and x are real, then u also has this property and the computation of U_i, $i = 1, 2, \ldots, n$, is reduced to computing two products of real matrices and $n/2 - 1$ products of complex matrices. Since a product of complex matrices can be performed with three multiplications and five additions of real

Algorithm 1: Block circulant matrix-vector product

Input : Positive integers m, n, p, where $n = 2^k$, k a positive integer, the n-dimensional block vector $c = (C_i)_{i=0,n-1}$ with $m \times m$ block components which is the first block column of the block circulant matrix A, and the n-dimensional block vector $x = (X_i)_{i=0,n-1}$ with $m \times p$ block components.

Output: The block vector $y = Ax = (Y_i)_{i=0,n-1}$.

1 $w = (W_i)_{i=0,n-1} = \mathrm{IDFT}_n(c)$.
2 $v = (V_i)_{i=0,n-1} = \mathrm{IDFT}_n(x)$.
3 $U_i = W_i V_i$, $i = 0, 2, \ldots, n-1$.
4 $y = \mathrm{DFT}_n(u)$ for $u = (U_i)_{i=0,n-1}$.

matrices, the overall cost of stage 3 is $3(n/2-1)+2$ real matrix multiplications between $m \times m$ and $m \times p$ matrices, $n/2-1$ additions of $m \times m$ matrices, and $4(n/2-1)$ additions of $m \times p$ matrices. Therefore for a real input the cost of Algorithm 1 is

$$n(3m^2p + m(m+3p)/2)) + \frac{5}{2}(m^2+2mp)n\log n \tag{4.6}$$

ops up to lower order terms. If m and p are large enough, the dominant part of the complexity is $3m^2np$. By using the customary algorithm for a matrix-vector product the cost would be $2m^2n^2p$ ops.

The inverse of a block circulant matrix A can be easily computed by means of

$$A^{-1} = \frac{1}{n}(\overline{\Omega}_n \otimes I_m)\,\mathrm{Diag}(W_0^{-1}, \ldots, W_{n-1}^{-1})(\Omega_n \otimes I_m).$$

In fact, it is enough to compute the first block column of A^{-1} given by

$$Y = \frac{1}{n}(\overline{\Omega}_n \otimes I_m) \begin{bmatrix} W_0^{-1} \\ \vdots \\ W_{n-1}^{-1}. \end{bmatrix}.$$

Algorithm 2 synthesizes this computation

Algorithm 2: Inversion of a block circulant matrix

Input : Positive integers m, n, where $n = 2^k$, k a positive integer, the n-dimensional block vector $r = (A_i)_{i=0,n-1}$ with $m \times m$ block components defining the first block row of the block circulant matrix A.

Output: The block column vector $Y = (Y_i)$ defining the first block column of A^{-1}.

1 $w = (W_i)_{i=0,n-1} = \mathrm{IDFT}_n(c)$, $c = (C_i)$, $C_i = A_{n-i-1}$.
2 $V_i = W_i^{-1}$, $i = 0, \ldots, n-1$.
3 $y = (Y_i)_{i=0,n-1} = \mathrm{DFT}_n(v)$, $v = (V_i)$.

The cost of Algorithm is $O(m^2 n \log n + nm^3)$.

4.3. **z-circulant matrices.** A generalization of circulant matrices is provided by the class of z-circulant matrices.

Definition 2. *Given a scalar $z \neq 0$ and the row vector $[a_0, a_1, \ldots, a_{n-1}]$, the $n \times n$ matrix*

$$A = \begin{bmatrix} a_0 & a_1 & \ldots & a_{n-1} \\ za_{n-1} & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_1 \\ za_1 & \ldots & za_{n-1} & a_0 \end{bmatrix}$$

is called the z-circulant *matrix associated with $[a_0, a_1, \ldots, a_{n-1}]$.*

Observe that a z-circulant matrix is fully defined by z and by the elements in its first row $r^{\mathrm{T}} = [a_0, a_1, \ldots, a_{n-1}]$ or in its first column $c = [a_0, za_{n-1}, \ldots, za_1]^{\mathrm{T}}$.

We denote by S_z the z-circulant matrix whose first row is $[0, 1, 0, \ldots, 0]$, i.e.,

$$S_z = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & 1 \\ z & 0 & \ldots & 0 & 0 \end{bmatrix},$$

and we easily verify that

$$A = \sum_{i=0}^{n-1} a_i S_z^i. \tag{4.7}$$

That is, any z-circulant matrix can be viewed as a polynomial in S_z.

It is simple to verify that

$$S_{z^n} = zD_z S D_z^{-1},\ D_z = \mathrm{Diag}(1, z, z^2, \ldots, z^{n-1}),$$

where S is the circulant matrix in (4.1). Therefore, if A is z^n-circulant, from (4.7) we deduce that

$$A = D_z \left(\sum_{i=0}^{n-1} a_i z^i S^i \right) D_z^{-1}$$

where $\sum_{i=0}^{n-1} a_i z^i S^i$ is circulant. Whence, from Proposition 6 we obtain the following

Proposition 8. *If A is the z^n-circulant matrix with first row r^{T} and first column c then*

$$A = \frac{1}{n} D_z \overline{\Omega}_n \,\mathrm{Diag}(w) \Omega_n D_z^{-1},$$

with $w = \overline{\Omega}_n D_z r = \Omega_n D_z^{-1} c$.

The above theorem states that, like circulant matrices, all the matrices in the z^n-circulant class can be simultaneously diagonalized by means of a combination of DFT and diagonal scaling with the integer powers of z. Therefore, for any given z, z-circulant matrices are closed under matrix multiplication and inversion.

The extension to block matrices trivially applies to z-circulant matrices.

Definition 3. *Given a scalar $z \neq 0$ and the block row vector $[A_0, A_1, \ldots, A_{n-1}]$, the $n \times n$ matrix*

$$A = \begin{bmatrix} A_0 & A_1 & \ldots & A_{n-1} \\ zA_{n-1} & A_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & A_1 \\ zA_1 & \ldots & zA_{n-1} & A_0 \end{bmatrix}$$

*is called the block z-*circulant *matrix associated with $[A_0, A_1, \ldots, A_{n-1}]$.*

The analog of Proposition 8 for block z-circulant matrices is stated below.

Proposition 9. *If A is the block z^n-circulant matrix with first block column c and with first block row $[A_0, A_1, \ldots, A_{n-1}]$, then*

$$A = \frac{1}{n} (D_z \otimes I_m)(\overline{\Omega}_n \otimes I_m)\,\mathrm{Diag}(w)(\Omega_n \otimes I_m)(D_z^{-1} \otimes I_m),$$

where

$$w = (\overline{\Omega}_n \otimes I_m)(D_z \otimes I_m) \begin{bmatrix} A_0 \\ \vdots \\ A_{n-1} \end{bmatrix} = (\Omega_n \otimes I_m)(D_z^{-1} \otimes I_m)c.$$

Likewise the case of circulant matrices, the product of a z-circulant matrix and a vector can be performed by means of FFT in $O(n \log n)$ ops. Similarly, the product of a block z-circulant matrix with $m \times m$ blocks and a block vector can be performed with $O(m^2 n \log n + nm^3)$ ops.

The inverse of a block z^n-circulant matrix A is given by

$$A^{-1} = \frac{1}{n}(D_z \otimes I_m)(\overline{\Omega}_n \otimes I_m)\,\mathrm{Diag}(\widehat{w})(\Omega_n \otimes I_m)(D_z^{-1} \otimes I_m),$$
$$\widehat{w} = (W_i^{-1}), \quad w = (W_i),$$

where w is defined in Proposition 9.

The first block column of A^{-1} is given by

$$\frac{1}{n}(D_z \otimes I_m)(\overline{\Omega}_n \otimes I_m)\widehat{w}, \quad \widehat{w} = (\widehat{W}_i),\ \widehat{W}_i = W_i^{-1}.$$

This expression provides an algorithm for inverting a z-circulant matrix which is reported in Algorithm 3.

Algorithm 3: Inversion of a block z-circulant matrix

Input : A complex number $z \neq 0$ and positive integers m, k, $n = 2^k$, and the n-dimensional block vector $r = (A_i)_{i=0,n-1}$ with $m \times m$ block components defining the first block row of the block z-circulant matrix A.

Output: The block column vector $Y = (Y_i)$ defining the first block column of A^{-1}.

1 $w = (W_i)_{i=0,n-1} = \mathrm{IDFT}_n(c)$, $c = (C_i)_{i=0,n-1}$, $C_0 = 0$, $C_i = z^{1-\frac{i}{n}} A_{n-i}$, $i = 1, \ldots, n-1$.

2 $V_i = W_i^{-1}$, $i = 0, \ldots, n-1$.

3 $y = (Y_i)_{i=0,n-1} = \mathrm{DFT}_n(v)$, $v = (z^{\frac{i}{n}} V_i)_{i-0,n-1}$.

4.4. **Embedding Toeplitz matrices into circulants.** An $n \times n$ Toeplitz matrix $A = (t_{i,j})$, $t_{i,j} = a_{j-i}$, can be embedded into the $2n \times 2n$ circulant matrix B defined by its first row given by $[a_0, a_1, \ldots, a_{n-1}, *, a_{-n+1}, \ldots, a_{-1}]$, where $*$ denotes any number. We observe that the leading $n \times n$ submatrix of B coincides with A. An example with $n = 3$ is shown below

$$B = \left[\begin{array}{ccc|ccc} a_0 & a_1 & a_2 & * & a_{-2} & a_{-1} \\ a_{-1} & a_0 & a_1 & a_2 & * & a_{-2} \\ a_{-2} & a_{-1} & a_0 & a_1 & a_2 & * \\ \hline * & a_{-2} & a_{-1} & a_0 & a_1 & a_2 \\ a_2 & * & a_{-2} & a_{-1} & a_0 & a_1 \\ a_1 & a_2 & * & a_{-2} & a_{-1} & a_0 \end{array}\right].$$

More generally, an $n \times n$ Toeplitz matrix can be embedded into a $q \times q$ circulant matrix for any $q \geq 2n-1$: it is sufficient to replace $*$ with $q - 2n + 1$ arbitrary elements. If $q = 2n-1$ there is no extra element. Similarly, an $n \times n$ block Toeplitz matrix A with $m \times m$ blocks can be embedded into a $q \times q$ block circulant matrix B with $m \times m$ blocks for any $q \geq 2n-1$.

This embedding property allows one to compute the product $y = Ax$ of a (block) Toeplitz matrix A and a (block) vector x by means of Algorithm 1 in the following manner. First we embed A into a circulant matrix B. Second we define the q-dimensional block vector $z = (Z_i)$ obtained by filling up x with zeros, i.e., $Z_i = X_i$, $i = 1, 2, \ldots, n$, $Z_i = 0$ elsewhere. The first n block components of the block vector $w = Bz$ coincide with y. If q is chosen as an integer power of 2, then the product Bz can be efficiently computed by means of Algorithm 1 which is based on the FFT.

We synthesize this computation in Algorithm 4 which includes the scalar case when $m = p = 1$. The complexity analysis of Algorithm 4 can be carried out similarly to the case of Algorithm 1 and leads to the computational cost of $O((m+p)mn \log n + nm^2 p)$ ops.

If $p = m = 1$, that is, A is a Toeplitz matrix and x is a vector, then the asymptotic cost reduces to $O(n \log n)$ ops, versus the $O(n^2)$ cost of the customary algorithm for matrix-vector multiplication. If $m = p$, the asymptotic cost is $O(m^2 n \log n + m^3 n) = O(m^2 n(\log n + m))$; thus,

Algorithm 4: Block Toeplitz matrix-vector product

Input : Positive integers m, n, p, the $m \times m$ matrices A_i, $i = -n+1, \ldots, n-1$, which define the $n \times n$ block Toeplitz matrix $A = (A_{j-i})_{i,j=1,n}$; the n-dimensional block vector $x = (X_i)_{i=1,n}$ with $m \times p$ block components.

Output: The block vector $y = Ax = (Y_i)_{i=1,n}$.

1 Compute the least integer k such that $2^k \geq 2n$; set $q = 2^k$.

2 Define the q-dimensional block column vector $v = (V_i)_{i=1,q}$ such that $V_i = A_{-i+1}$ if $i = 1, \ldots, n$, $V_{q-i+1} = A_i$ if $i = 1, 2, \ldots, n-1$, and $V_i = 0$ elsewhere, and define the $q \times q$ block circulant matrix B having the first block column v. The block Toeplitz matrix A is embedded in B.

3 Define the q-dimensional block column vector $z = (Z_i)_{i=1,q}$ such that $Z_i = X_i$ if $i = 1, \ldots, n$, $X_i = 0$ elsewhere.

4 Compute $w = Bz = (W_i)_{i=1,q}$ by means of Algorithm 1.

5 Set $Y_i = W_i$, $i = 1, \ldots, n$.

if m is large with respect to $\log n$, the cost of computing FFTs is negligible with respect to the cost of computing the matrix products.

From the complexity bound (4.6) of the product of a circulant matrix and a vector we deduce that for real input the complexity bound of the product of a Toeplitz matrix and a vector is

$$q(3m^2p + m(m+3p)/2)) + \frac{5}{2}(m^2 + 2mp)q \log q \tag{4.8}$$

up to terms of lower order, where q is the minimum integer power of 2 greater than $2n-1$.

4.5. **Triangular Toeplitz matrices.** Let $Z = (z_{i,j})_{i,j=1,n}$ be the $n \times n$ matrix

$$Z = \begin{bmatrix} 0 & & & 0 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & 1 & 0 \end{bmatrix}, \tag{4.9}$$

with $z_{i+1,1} = 1$ for $i = 1, \ldots, n-1$, $z_{i,j} = 0$ elsewhere.

Clearly $Z^n = 0$, moreover, given the polynomial $a(x) = \sum_{i=0}^{n-1} a_i x^i$, the matrix $a(Z) = \sum_{i=0}^{n-1} a_i Z^i$ is a lower triangular Toeplitz matrix defined by its first column $(a_0, a_1, \ldots, a_{n-1})^{\mathrm{T}}$

$$a(Z) = \begin{bmatrix} a_0 & & & 0 \\ a_1 & a_0 & & \\ \vdots & \ddots & \ddots & \\ a_{n-1} & \ldots & a_1 & a_0 \end{bmatrix}.$$

The set of lower triangular Toeplitz matrices is closed under matrix multiplication. More precisely, if $A = a(Z)$, $B = b(Z)$ for two polynomials $a(x)$, $b(x)$, then the matrix $C = c(Z)$, where $c(x) = a(x)b(x) \mod x^n$, is such that $C = AB = BA$.

In other words, the set of lower triangular Toeplitz matrices is a matrix algebra isomorphic to the set of polynomials with the product modulo x^n.

Similarly, we can define the algebra of upper triangular Toeplitz matrices.

The definition of lower (upper) triangular Toeplitz matrix can be extended to the case of block matrices.

Observe that by the Cayley–Hamilton theorem the inverse of any nonsingular matrix A can be written as a polynomial in A, therefore T_n^{-1} is still a lower triangular Toeplitz matrix and the computation of T_n^{-1} is equivalent to computing the elements in the first column of T_n^{-1}. Similarly, the class of block triangular Toeplitz matrices is closed under matrix product and matrix inversion.

Now, assume $n = 2h$, h a positive integer, and partition T_n into $(n/2) \times (n/2)$ blocks, writing

$$T_n = \begin{bmatrix} T_h & 0 \\ W_h & T_h \end{bmatrix}, \tag{4.10}$$

where T_h, W_h are $h \times h$ Toeplitz matrices and T_h is lower triangular. If T_n is nonsingular then T_h also is nonsingular and

$$T_n^{-1} = \begin{bmatrix} T_h^{-1} & 0 \\ -T_h^{-1} W_h T_h^{-1} & T_h^{-1} \end{bmatrix}.$$

Thus, the first column v_n of T_n^{-1} is given by

$$v_n = T_n^{-1} e_1 = \begin{bmatrix} v_h \\ -T_h^{-1} W_h v_h \end{bmatrix} = \begin{bmatrix} v_h \\ -L(v_h) W_h v_h \end{bmatrix}, \tag{4.11}$$

where $L(v_h) = T_h^{-1}$ is the lower triangular Toeplitz matrix whose first column is v_h.

The same relation holds if T_n is block triangular Toeplitz. In this case, the elements $a_0, \ldots, a_{n-1}$ are replaced with the $m \times m$ blocks $A_0, \ldots, A_{n-1}$ and v_n denotes the first block column of T_n^{-1}.

The representation (4.11) of v_n leads to a recursive algorithm for its computation (Algorithm 5), which we describe for block triangular Toeplitz matrices of block size $n = 2^k$, for a positive integer k.

Algorithm 5: Block lower triangular Toeplitz matrix inversion

Input : The positive integer k and the $m \times m$ block elements $A_0, \ldots, A_{n-1}$, $n = 2^k$, of the first block column of the block lower triangular Toeplitz matrix T_n, where $\det A_0 \neq 0$.

Output: The first block column v_n of T_n^{-1}.

1 Set $v_1 = A_0^{-1}$

2 For $i = 0, \ldots, k-1$, given v_h, $h = 2^i$:

- Compute the block Toeplitz matrix-vector products $w = W_h v_h$ and $u = -L(v_h) w$.
- Set $v_{2h} = \begin{bmatrix} v_h \\ u \end{bmatrix}$.

The computation of the block vector u at the ith step of Algorithm 5 requires the computation of two products of a block Toeplitz matrix and a block vector. Since this can be performed in $O(2^i m^3 + i 2^i m^2)$ ops by means of Algorithm 4, the overall cost of Algorithm 5 is $O(nm^3 + nm^2 \log n)$. More precisely, if the input is real, from the complexity estimate (4.8) with $p = m$, we deduce that the complexity bounds of the ith stage reduce to $2(3m^3 q + \frac{15}{2} m^2 q \log q + 2m^2 q)$ for $q = 2^{i+1}$. Moreover, in this complexity estimate, we counted twice the cost of the computation of the DFT of the vector v_h (filled with zeros) which appears in both the Toeplitz matrix-vector products $W_h v_h$ and $L(v_h) w$ (see Steps 2 and 3 of Algorithm 4). Taking this into consideration, the complexity bound is reduced to $\frac{25}{2} m^2 h \log(h) + 6m^3 h + 4m^2 h$. Therefore, since $\sum_{i=0}^{k-1} 2^{i+1} = 2(2^k - 1)$, the overall cost is less than

$$25 m^2 n \log(2n) + (12m^3 + 8m^2) n \tag{4.12}$$

where we have bounded $\log 2^i$ with $\log n$.

The block triangular Toeplitz system $T_n x = b$ can be solved in $O(m^2 n \log n + m^3 n)$ ops, by first computing the inverse matrix T_n^{-1} by means of Algorithm 5 and then computing the product $T_n^{-1} b$ by means of Algorithm 4.

Alternatively, the computation of the first block column of the inverse of T_n might be performed by using the customary approach, i.e, by inverting the diagonal block A_0 and by applying forward substitution. This amounts to computing $n(n+1)/2$ matrix products and $n(n-1)/2$ matrix sums. The cost in the case of real input is $n^2 m^3$ ops, up to terms of lower order.

Algorithm 5 can be easily adjusted to invert a block upper triangular Toeplitz matrix at the same cost. This is described as Algorithm 6.

If T_n is block upper triangular Toeplitz, then the system $T_n x = b$ can be solved in $O(m^2 n \log n + m^3 n)$ ops, by first computing the inverse matrix T_n^{-1} by means of Algorithm 7 and then computing the product $T_n^{-1} x$ by means of Algorithm 4.

Algorithm 6: Block upper triangular Toeplitz matrix inversion

Input : The positive integer k and the $m \times m$ block elements $A_0, \ldots, A_{n-1}$, $n = 2^k$, of the first block row of the upper block triangular Toeplitz matrix T_n, where $\det A_0 \neq 0$.

Output: The first block row v_n^{T} of T_n^{-1}.

1 Set $B_i = A_i^{\mathrm{T}}$, $i = 0, \ldots, n-1$, and $b = (B_{i-1})_{i=1,n}$.
2 Apply Algorithm 5 to the block lower triangular matrix $L(b) = T_n^{\mathrm{T}}$ and compute the first block column v_n of $(T_n^{\mathrm{T}})^{-1}$.
3 Output the block row vector v_n^{T}.

4.6. **Triangular Toeplitz and z-circulant matrices.** Observe that a z-circulant matrix C_z coincides with an upper triangular Toeplitz matrix if $z = 0$. Moreover, if $z \neq 0$ and $|z|$ is small enough, then C_z provides a "good" approximation of C_0. By continuity, C_z^{-1} provides a good approximation to the inverse of an upper triangular Toeplitz matrix. Based on this observation, we may construct an algorithm to approximate the inverse of an upper triangular Toeplitz matrix up to any desired precision. The algorithm, which relies on computing two IDFTs and one DFT plus some scalings, is reported in Algorithm 7.

Algorithm 7: Block upper triangular Toeplitz matrix inversion by means of z-circulants

Input : The positive integer k and the $m \times m$ block elements $A_0, \ldots, A_{n-1}$, $n = 2^k$, of the first block row of the upper block triangular Toeplitz matrix T_n, where $\det A_0 \neq 0$; a real $\epsilon > 0$ and a number $z \in \mathbb{C}$ such that $|z| = \epsilon$.

Output: An approximation to the first block row v_n^{T} of T_n^{-1}.

1 Set $B_i = A_i^{\mathrm{T}}$, $i = 0, \ldots, n-1$, and $b = (B_{i-1})_{i=1,n}$.
2 Apply Algorithm 5 to the block lower triangular matrix $L(b) = T_n^{\mathrm{T}}$ and compute the first block column v_n of $(T_n^{\mathrm{T}})^{-1}$.
3 Output the block row vector v_n^{T}.

The execution of the above algorithm in floating point arithmetic provides an approximation of the inverse where the error has two components: the analytic error given by a function of z, and the round-off error generated by the floating point arithmetic. A suitable error analysis shows that the round-off error is bounded in modulus by a quantity proportional to μ/ϵ, where μ is the machine precision, while the analytic error in its linear part is bounded in modulus by a quantity proportional to ϵ. The smaller ϵ, the smaller the analytic error. However, small values of ϵ generate large numerical errors. Asymptotically speaking when $\mu \to 0$, the best choice of ϵ is a value such that μ/ϵ and ϵ have the same order of magnitude, that is $\epsilon = O(\sqrt{\mu})$ so that the total error is $O(\sqrt{\mu})$.

In this way, asymptotically, using this algorithm halves the number of correct digits that one can get in the output. However, there are some tricks to improve the precision.

A first possibility, introduced in [10] relies on the fact that the analytic error is a polynomial in z. The trick consists in making different computations with different values of z, say $z_1, \ldots, z_k$ such that $|z_i| = \epsilon$ and then take a weighted average of the different results in such a way that all the positive power of z with exponent from 1 to $k-1$, cancel out. For instance, choose $z_i = \epsilon\omega_k^i$ and then take the arithmetic mean of the computed values. In this case the analytic error is $O(\epsilon^k)$ since all the terms z^i for $i = 1, \ldots, k$ cancel out (if $k > n$ then the analytic error of the computation is zero).

Thus, by equating the numerical error to the analytic error we get a larger value of the optimal ϵ which leads to a total error of $O(\mu^{k/(k+1)})$. That is, the loss of digits is $O(1/(k+1))$ and becomes negligible for large values of k. Indeed the complexity of the computation grows by a factor of k. However, in a parallel model of computation this growth is negligible since all the computations are independent and can be performed in parallel.

Another technique to reduce the error, due to Nick Higham [1], works for a real input and consists in choosing a pure imaginary value for z, that is, $z = \epsilon\underline{i}$. In this way the analytic error

affecting the real part is just $O(\epsilon^2)$. This way we have a total error $O(\mu^{2/3})$ with a loss of only $1/3$ digits. The number of operations are the same as before. The difference is that we have to use complex arithmetic which is more expensive than real arithmetic.

It is possible to combine the two tricks in the following way. Choose $z_1 = \epsilon(1 + \underline{i})/\sqrt{2}$ and $z_2 = -z_1$; apply the algorithm with $z = z_1$ and $z = z_2$; take the arithmetic mean of the results. The analytic error on the real part turns out to be $O(\epsilon^4)$. The total error is $O(\mu^{4/5})$. Only $1/5$ of digits are lost.

In general choosing $z_j = (\underline{i})^{1/k}\omega_k^j\epsilon$ where $(\underline{i})^{1/k}$ is a primitive kth root of $\underline{i}$, by applying k times the algorithms with these values and taking the arithmetic mean of the results obtained this way, we find that the real part of the approximation error is $O(\epsilon^{2k})$ and the overall error in the output is $O(\mu^{2k/(2k+1)})$, i.e., only $1/(2k+1)$ of digits are lost. We call this trick the *averaging technique*.

One other different technique, introduced by [58], consists in interpreting triangular matrix inversion like a trigonometric interpolation problem.

4.7. **The class τ.** Recall that circulant matrices can be diagonalized by a similarity transformation given by the unitary matrix F which defines the DFT. We can define other matrix classes relying on different discrete transform.

For instance, it is not complicated to show that the matrix

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

is such that $S^T H S = D$ is a diagonal matrix where $S = (\sqrt{\frac{2}{n+1}} \sin \frac{\pi ij}{n+1})$ is the orthogonal matrix associated with the discrete sine transform (DST).

It turns out that the all the polynomials in H form a matrix algebra diagonalizable by the DST. This algebra, called class τ in [11] has very interesting structural and computational properties, moreover it is effectively used as preconditioner for the iterative solution of positive Toeplitz systems.

4.8. **Other algebras.** Other algebras can be constructed with the 8 cosine transforms and with the other 7 sine transforms. In particular the algebra based on the discrete cosine transform has particular relevance in solving Toeplitz systems coming from image processing.

A different algebra is related to the discrete Hartley transform. The discrete Hartley transform is defined as $x \to y = Gx$ where $G = \frac{1}{\sqrt{n}}(\cos(ij\frac{\pi}{n}) + \sin(ij\frac{\pi}{n}))$. The algebra associated with this transform has been studied in [13] where in particular it is shown that the Hartley algebra contains symmetric circulant matrices.

5. Displacement operators

In general, the inverse of a Toeplitz matrix is not Toeplitz as the following simple example shows:

Example 1. Let $n = 4$ and

$$A = \begin{bmatrix} 4 & 3 & 2 & 1 \\ 0 & 4 & 3 & 2 \\ 1 & 0 & 4 & 3 \\ 0 & 1 & 0 & 4 \end{bmatrix}.$$

We have

$$A^{-1} = \frac{1}{265}\begin{bmatrix} 65 & -50 & 5 & 5 \\ 12 & 56 & -48 & 5 \\ -14 & 23 & 56 & -50 \\ -3 & -14 & 12 & 65 \end{bmatrix}.$$

However, it is possible to introduce a more general structure which is preserved under inversion. This structure relies on the concept of *displacement rank*. The concepts of displacement operator and of displacement rank, introduced in [57], and successively elaborated by other authors (see the survey [55] and [56]), is a powerful tool for dealing with Toeplitz matrices.

Here, we recall the main results concerning displacement rank. Throughout this section we refer to section 2.11 of [20].

Define the *displacement operator*

$$\Delta(A) = AZ - ZA, \tag{5.1}$$

applied to an $n \times n$ matrix A, where Z is the lower shift matrix of (4.9). Multiplying the matrix A on the left by Z shifts down each row of A by one position. Similarly, multiplying the matrix A on the right by Z shifts each column of A by one position to the left. In particular, if $A = (a_{j-i})_{i,j=1,n}$ is Toeplitz then

$$\Delta(A) = \left[\begin{array}{cccc|c} a_1 & a_2 & \dots & a_{n-1} & 0 \\ \hline & & & & -a_{n-1} \\ & & 0 & & \vdots \\ & & & & -a_2 \\ & & & & -a_1 \end{array}\right] = e_1 e_1^{\mathrm{T}} AZ - ZAe_n e_n^{\mathrm{T}}, \tag{5.2}$$

where e_1 and e_n denote the first and the last column of the $n \times n$ identity matrix. Therefore, $\Delta(A)$ has at most rank 2. We say that a matrix A has *displacement rank* (at most) k with respect to the operator Δ if $\operatorname{rank}\Delta(A) = k$ ($\operatorname{rank}\Delta(A) \leq k$). As a particular case, Toeplitz matrices have displacement rank at most 2, so that the class of matrices with "low" displacement rank are a generalization of Toeplitz matrices.

It is important to recall that an $n \times n$ matrix X has rank k if and only if there exist $n \times k$ matrices V, W of full rank such that $X = VW^{\mathrm{T}}$. Therefore, if $\Delta(A)$ has rank k then there exist two $n \times k$ matrices V and W of full rank such that $\Delta(A) = VW^{\mathrm{T}}$ and vice versa. Any pair (V, W) of such matrices is called *displacement generator* of A with respect to the operator Δ. For instance, in the case of (5.2) one has $\Delta(A) = VW^{\mathrm{T}}$ where

$$V = \left[\begin{array}{c|c} 1 & 0 \\ 0 & a_{n-1} \\ \vdots & \vdots \\ 0 & a_1 \end{array}\right], \; W^{\mathrm{T}} = \left[\begin{array}{cccc} a_1 & \dots & a_{n-1} & 0 \\ \hline 0 & \dots & 0 & -1 \end{array}\right].$$

Observe that we might have a pair of $n \times h$ matrices (V, W) such that $h > k$, $VW^{\mathrm{T}} = \Delta(A)$ and $\operatorname{rank}(V) = \operatorname{rank}(W) = \operatorname{rank}\Delta(A) = k$. We call such a pair a *displacement generator of nonminimal rank* of A. A generator of nonminimal rank stores the information about $\Delta(A)$ in a redundant way. Using numerical linear algebra tools like the singular value decomposition any generator can be reduced to minimal rank.

The displacement generator of a matrix A, together with the first column of A, contains all the information which allows one to represent all the elements of A, as shown in the next proposition.

Proposition 10. *Let A be an $n \times n$ matrix having first column a and displacement rank k. If (V, W) is a displacement generator of A, then*

$$A = L(a) + \sum_{i=1}^{k} L(v_i) U(w_i^{\mathrm{T}} Z^{\mathrm{T}}),$$

where v_i and w_i are the ith column of V and W, respectively.

Equivalent representations can be given in terms of different operators. Besides the Sylvester type operator Δ we may consider operators of the Stein type like $\Delta(A) = A - ZAZ^{\mathrm{T}}$. More generally, we may consider operators like $AZ_1 - Z_2 A$ or $A - Z_1 A Z_2^{\mathrm{T}}$ where Z_1 and Z_2 can be different, say z-circulant, or $(-z)$-circulant. We refer the reader to [55], [56] and to [20, Section 2.11] for more details on this regard.

A simple but important consequence of Proposition 10 is that any matrix with displacement rank k can be decomposed as the sum of at most $k + 1$ matrices, each of them the product of a

lower and an upper triangular Toeplitz matrix. Therefore, the product $y = Ax$ can be split into at most $2k+1$ products of triangular Toeplitz matrices and vectors. Each one of these products can be efficiently performed by using Algorithm 4 in $O(n \log n)$ ops. The overall cost of the algorithm for computing y in this way is $O(kn \log n)$.

Another nice consequence of Proposition 10 concerns the inverse matrix of A. Observe that if A is nonsingular then pre- and post-multiplying (5.1) by A^{-1} yields the simple relation

$$\Delta(A^{-1}) = -A^{-1}\Delta(A)A^{-1}, \tag{5.3}$$

from which we conclude that the displacement ranks of A and of A^{-1} coincide. Moreover, given a displacement generator (V, W) of the nonsingular matrix A, the pair $(-A^{-1}V, (A^{-1})^{\mathrm{T}}W)$ is a displacement generator for A^{-1}. This allows one to represent A^{-1} in a compact way by means of Proposition 10 as

$$A^{-1} = L(A^{-1}e_1) - \sum_{i=1}^{k} L(A^{-1}v_i)U(w_i^{\mathrm{T}}A^{-1}Z^{\mathrm{T}}). \tag{5.4}$$

Observe that even though the inverse of a Toeplitz matrix A is not generally Toeplitz, its displacement rank is at most 2.

Example 2. For the Toeplitz matrix A of Example 1 we have $\Delta(A^{-1}) = VW^{\mathrm{T}}$ where

$$V = \frac{1}{265}\begin{bmatrix} -65 & 25 \\ -12 & 25 \\ 14 & 15 \\ 3 & -205 \end{bmatrix},\ W^{\mathrm{T}} = \frac{1}{265}\begin{bmatrix} 205 & -15 & -25 & -25 \\ 3 & 14 & -12 & -65 \end{bmatrix}.$$

Computing the displacement representation of A^{-1} is reduced to solving at most $2k+1$ linear systems. This is particularly convenient when k is small with respect to n. Moreover, once A^{-1} is represented by means of its displacement generator, solving any additional system of the kind $Ax = b$ is reduced to computing the product $x = A^{-1}b$, with the cost of $O(kn \log n)$ ops.

Displacement representations are also useful for computing products of matrices with low displacement rank. It is a simple matter to prove that

$$\Delta(AB) = A\Delta(B) + \Delta(A)B \tag{5.5}$$

so that a displacement generator (possibly of nonminimal rank) of $C = AB$ is given by (V_C, W_C) where

$$V_C = \left[\ AV_B \mid V_A\ \right],\ W_C = \left[\ W_B \mid B^{\mathrm{T}}W_A\ \right],$$

and (V_A, W_A) and (V_B, W_B) are displacement generators of A and B, respectively. Therefore, to compute the displacement generator (possibly of nonminimal rank) of the product AB given the displacement generators of A and B, one only needs to compute the products AV_B and $B^{\mathrm{T}}W_A$, at the cost $O(kn \log n)$, where k is the maximum displacement rank of A and B, if we use the displacement representation of A and B and Algorithm 4.

The concept of displacement rank is easily extended to block matrices. Let $\mathcal{Z} = Z \otimes I$, where I is the $m \times m$ identity matrix, and consider the operator $A \to A\mathcal{Z} - \mathcal{Z}A$ applied to an $n \times n$ block matrix with $m \times m$ blocks. Observe that this operator coincides with Δ if $m = 1$. Therefore, for notational simplicity, we will denote it with the same symbol and write that $\Delta(A) = A\mathcal{Z} - \mathcal{Z}A$.

It is easy to see that, if $A = (A_{j-i})_{i,j=1,n}$ is block Toeplitz, then

$$\Delta(A) = \left[\begin{array}{cccc|c} A_1 & A_2 & \dots & A_{n-1} & 0 \\ \hline & & & & -A_{n-1} \\ & & 0 & & \vdots \\ & & & & -A_2 \\ & & & & -A_1 \end{array}\right]$$
$$= (e_1 \otimes I)(e_1 \otimes I)^{\mathrm{T}}A\mathcal{Z} - \mathcal{Z}A(e_n \otimes I)(e_n \otimes I)^{\mathrm{T}}.$$

We say that the block matrix A has *block displacement rank* k if k is the minimum integer such that there exist $n \times k$ block matrices V and W with $m \times m$ blocks satisfying $\Delta(A) = VW^{\mathrm{T}}$. Any such pair (V, W) is called a *block displacement generator* of A. The representation theorem 10

still holds as well as equations (5.3), (5.4) and (5.5) suitably adjusted to the block notation. We synthesize these properties in the following

Proposition 11. *Let $\Delta(A) = A\mathcal{Z} - \mathcal{Z}A$. If A is nonsingular then*

$$\Delta(A^{-1}) = -A^{-1}\Delta(A)A^{-1}.$$

Given matrices A, B, C such that $A = BC$ then

$$\Delta(A) = B\Delta(C) + \Delta(B)C.$$

Moreover, if $\Delta(A) = VW^{\mathrm{T}}$, where V and W are $n \times k$ block matrices with $m \times m$ blocks, then

$$A = L(a) + \sum_{i=1}^{k} L(v_i)U(w_i^{\mathrm{T}}\mathcal{Z}^{\mathrm{T}})$$

where a is the first block column of A, v_i and w_i denote the ith block column of V and W, respectively, for $i = 1, \ldots, k$. In particular, if A is nonsingular, then

$$A^{-1} = L(a') - \sum_{i=1}^{k} L(A^{-1}v_i)U(w_i^{\mathrm{T}}A^{-1}\mathcal{Z}^{\mathrm{T}})$$

where a' is the first block column of A^{-1}.

It is interesting to point out that, $\Delta(A) = 0$ if and only if A is block lower triangular Toeplitz. Observe also that the "dual" operator $\Delta_2(A) = A\mathcal{Z}^{\mathrm{T}} - \mathcal{Z}^{\mathrm{T}}A$, which satisfies the same formal properties of Δ, is such that $\Delta_2(A) = 0$ if and only if A is block upper triangular Toeplitz.

We summarize the properties of $\Delta_2(A)$ in the following theorem.

Proposition 12. *Let $\Delta_2(A) = A\mathcal{Z}^{\mathrm{T}} - \mathcal{Z}^{\mathrm{T}}A$. If A is nonsingular then*

$$\Delta_2(A^{-1}) = -A^{-1}\Delta_2(A)A^{-1}.$$

Given matrices A, B, C such that $A = BC$ then

$$\Delta_2(A) = B\Delta_2(C) + \Delta_2(B)C.$$

Moreover, if $\Delta_2(A) = VW^{\mathrm{T}}$, where V and W are $n \times k$ block matrices with $m \times m$ blocks, then

$$A = U(a^{\mathrm{T}}) - \sum_{i=1}^{k} L(\mathcal{Z}v_i)U(w_i^{\mathrm{T}})$$

where v_i and w_i denote the ith block column of V and W, respectively, for $i = 1, \ldots, k$, and a^{T} is the first block row of A. In particular, if A is nonsingular, then

$$A^{-1} = U(a''^{\mathrm{T}}) + \sum_{i=1}^{k} L(\mathcal{Z}A^{-1}v_i)U(w_i^{\mathrm{T}}A^{-1}).$$

where a''^{T} is the first block row of A^{-1}.

Another interesting property which relates the operators $\Delta_1 = \Delta$ and Δ_2 is expressed by the following

Proposition 13. *If $\Delta_1(A) = A\mathcal{Z} - \mathcal{Z}A$ and $\Delta_2(A) = A\mathcal{Z}^{\mathrm{T}} - \mathcal{Z}^{\mathrm{T}}A$ then*

$$\Delta_1(A) = -\mathcal{Z}\Delta_2(A)\mathcal{Z} - \mathcal{Z}A(e_m \otimes I)(e_m \otimes I)^{\mathrm{T}} + (e_1 \otimes I)(e_1 \otimes I)^{\mathrm{T}}A\mathcal{Z}.$$

The Toeplitz structure, and more generally, the displacement structure, can be effectively used for computing matrix inverses. For solving general block Toeplitz systems there are algorithms based on Schur complementation and displacement rank.

An alternative to direct algorithms are iterative algorithms which provide a sequence of successive approximations to the solution of the linear system $Ax = b$. For positive definite systems, particular attention is paid to conjugate gradient iteration which provides the exact solution after mn steps, but which may provide reasonable approximations after just a few iterations. Furthermore, convergence can be accelerated by means of preconditioning techniques.

Each step of the conjugate gradient requires the multiplication of a Toeplitz matrix and a vector. This computation is inexpensive if performed by means of Algorithm 4. For nonsymmetric matrices, iterative algorithms like GMRES and Krylov methods should be used. All these algorithms for Toeplitz inversion can be extended to the case of matrices having a low displacement rank.

5.1. **The Gohberg-Semencul-Trench formula.** Proposition 12 provides a representation of the inverse of a Toeplitz matrix as the sum of two terms, each term is the product of a lower triangular Toeplitz matrix and an upper triangular Toeplitz matrix. After some manipulations, this expression can be written as

$$T^{-1} = \frac{1}{x_0}\left(L(x)L^T(Jy) - L(Zy)L^T(ZJx)\right),$$

$$x = T^{-1}e_1,\ y = T^{-1}e_n,\ J = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}$$

which is known as the Gohberg-Semencul-Trench formula.

Observe that, once the vectors defining the four matrices have been computed, the product between the inverse of a Toeplitz matrix and a vector costs only $O(n\log n)$ operations. On the other hand, these vectors can be obtained by solving Toeplitz systems with suitable known term vector.

5.2. **Other operators.** Different operators can be defined by following the same line of displacement operators. For instance, if D_1 and D_2 are diagonal matrices with pairwise different entries $d_i^{(1)}$, $d_j^{(2)}$ respectively, and such that the diagonal entries of D_1 are different from those of D_2, then the operator $X \to \Delta(X) = D_1X - XD_2$ is invertible. Moreover if $\Delta(X) = uv^T$ for two vectors $u = (u_i)$ and $v = (v_i)$, then $x_{i,j} = u_iv_j/(d_i^{(1)} - d_j^{(2)})$. Similarly, if $\Delta(X) = \sum_{r=1}^k u^{(r)}v^{(r)T}$ then $x_{i,j} = \sum_{r=1}^k u_i^{(r)}v_j^{(r)}/(d_i^{(1)} - d_j^{(2)})$. Matrices with this structure are said Cauchy-like matrices.

A nice feature of Cauchy-like matrices is that their Schur complement is still a Cauchy-like matrix [47]. For instance, in the case where $k = 1$ the Cauchy-like matrix C can be partitioned as

$$C = \left[\begin{array}{c|ccc} \frac{u_1v_1}{d_1^{(1)}-d_1^{(2)}} & \frac{u_1v_2}{d_1^{(1)}-d_2^{(2)}} & \cdots & \frac{u_1v_n}{d_1^{(1)}-d_n^{(2)}} \\ \hline \frac{u_2v_1}{d_2^{(1)}-d_1^{(2)}} & & & \\ \vdots & & \widehat{C} & \\ \frac{u_nv_1}{d_n^{(1)}-d_1^{(2)}} & & & \end{array}\right]$$

where $\widehat{C}$ is still Cauchy-like matrix. Its Schur complement is given by

$$\widehat{C} - \begin{bmatrix} \frac{u_2v_1}{d_2^{(1)}-d_1^{(2)}} \\ \vdots \\ \frac{u_nv_1}{d_n^{(1)}-d_1^{(2)}} \end{bmatrix} \frac{d_1^{(1)} - d_1^{(2)}}{u_1v_1} \begin{bmatrix} \frac{u_1v_2}{d_1^{(1)}-d_2^{(2)}} & \cdots & \frac{u_1v_n}{d_1^{(1)}-d_n^{(2)}} \end{bmatrix}$$

and has entries that can be written in the form

$$\frac{\widehat{u}_i\widehat{v}_i}{d_i^{(1)} - d_j^{(2)}},\quad \widehat{u}_i = u_i\frac{d_1^{(1)} - d_i^{(1)}}{d_i^{(1)} - d_1^{(2)}},\quad \widehat{v}_j = v_j\frac{d_j^{(2)} - d_1^{(2)}}{d_1^{(1)} - d_j^{(2)}}.$$

This property can be exploited for designing an algorithm which computes the LU factorization of a Cauchy-like matrix with a number of operations which grows as $O(n^2k)$. The algorithm is due to Gohberg, Kailath and Olshevsky and is known as *GKO algorithm* [47]. Modified and more efficient versions of this algorithm have been devised [65]. Once the LU factorization is available, the solution of a linear system with a Cauchy-like matrix can be computed in $O(n^2)$ ops.

5.3. **Algorithm for Toeplitz inversion.** Algorithms for solving Toeplitz systems are usually classified in *fast algorithms* if their complexity is $O(n^2)$ and *super-fast algorithms* if their complexity is $O(n\log_n^2)$.

Among the fast classical algorithms we recall Levinson, and Trench-Zohar algorithms [73],[84]. A more recent fast algorithm is based on reducing a Toeplitz-like matrix to a Cauchy-like matrix. The idea of the algorithm is simple.

Consider the operator $\Delta(A) = S_1 A - A S_{-1}$ where S_1 is the unit circulant matrix and S_{-1} is the unit -1-circulant matrix. Recall that $S_1 = F^* D_1 F$ where $F = \frac{1}{\sqrt{n}}(\omega_n^{ij})$ and D_1 is the diagonal matrix defined by $D_1 = \mathrm{diag}(1, \bar{\omega}_n, \ldots, \bar{\omega}_n^{n-1})$, while $S_{-1} = D F^* D_{-1} F D^{-1}$, where $D = \mathrm{diag}(1, \delta, \ldots, \delta^{n-1})$, $\delta = \omega_n^{1/2} = \omega_{2n}$, and $D_{-1} = \delta D_1$.

We have already pointed out that if A is a Toeplitz matrix then $\Delta(A)$ has rank at most 2. More generally, if A is Toeplitz-like the rank of $\Delta(A)$ is independent of n, say k. On the other hand,

$$\Delta(A) = S_1 A - A S_{-1} = F^* D_1 F A - A D F^* D_{-1} F D^{-1},$$

so that, multiplying the above expression to the left by F and to the right by DF^* it turns out that the matrix $B = FADF^*$ is such that the transformed matrix $D_1 B - B D_{-1}$ has rank at most 2. Thus, B is a Cauchy-like matrix of rank at most 2. This way, the linear system $Ax = b$ can be transformed into

$$By = Fb,$$
$$x = DF^* y,$$

and reduced to computing two DFTs and solving a Cauchy-like system for the overall cost of $O(n^2 k)$ ops by means of the GKO algorithm.

5.4. **Divide and conquer techniques: super-fast algorithms.** Here we give an idea of super-fast Toeplitz solvers. In particular, we provide a general description of the classical approach that is the Bitmead-Anderson algorithm [23].

Other different approaches exist. Recently in [81] a super-fast and stable Toeplitz solver has been designed relying on the reduction to Cauchy matrices described in the previous section and on the properties of rank structured matrices.

Consider the operator $F_+(A) = A - ZAZ^T$ and partition the matrix A as $A = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$ so that

$$A = \begin{bmatrix} I & 0 \\ A_{2,1}A_{1,1}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{1,1} & A_{1,2} \\ 0 & B \end{bmatrix}, \quad B = A_{2,2} - A_{2,1}A_{1,1}^{-1}A_{1,2}.$$

A fundamental property is that the Schur complement B keeps the same displacement rank as A, that is, $\mathrm{rank}F_+(A) = \mathrm{rank}F_+(B)$; the other blocks of the LU factorization have almost the same displacement rank of A.

Therefore, solving two systems with the matrix A, for computing the displacement representation of A^{-1}, is reduced to

- : – solving two systems with the matrix $A_{1,1}$ for computing the displacement representation of $A_{1,1}^{-1}$,
- : – solving two systems with the matrix B which has displacement rank 2, plus performing some Toeplitz-vector products.

This way, the overall cost $C(n)$ of this recursive step is given by $C(n) = 2C(n/2) + O(n \log n)$ and this implies that $C(n) = O(n \log^2 n)$.

6. Trigonometric matrix algebras and preconditioning

The solution of a large positive definite $n \times n$ Toeplitz system $A_n x = b$ can be efficiently approximated with the *Preconditioned Conjugate Gradient (PCG) method* [66].

It is known from the classical theory that the conjugate gradient method applied to an $n \times n$ positive definite system $Ax = b$ provides a sequence of approximations $\{x_i\}_i$ to the solution x which converge to x after n steps. However, the residual error $\|Ax_i - b\|_2$ is bounded from above by $\gamma\theta^i$, where γ is a positive constant and $\theta = (\sqrt{\mu}-1)/(\sqrt{\mu}+1)$ where $\mu = \mu(A)$ is the spectral condition number of the matrix A. That is, $\mu(A) = \max_i \lambda_i(A) / \min_i \lambda_i(A)$ where $\lambda_i(A)$ for $i = 1, \ldots, n$ are the eigenvalues of A.

This way, if A has condition number close to 1 then after a few number i of iterations the vector x_i provides a good approximation to the solution x. The cost of each iteration is dominated by the computation of a matrix vector product. If A is Toeplitz this cost amounts just to $O(n \log_2 n)$ ops. Therefore this iteration is very convenient for well conditioned Toeplitz systems.

On the other hand, if A is ill-conditioned then the conjugate gradient needs a large number of iterations. In this case, the PCG method comes into help since one has to find a good preconditioner P having the following three features

(1) the cost of computing the matrix-vector product Pv is comparable with that of computing Av;
(2) the matrix P is easily invertible;
(3) the matrix P mimics the spectral properties of A in such a way that $P^{-1}A$ is "close" to the identity matrix.

The latter condition aims to realize the clustering of the spectrum of $P^{-1}A$ around 1, in the sense that the condition number of $P^{-1}A$ is close to 1, so that the CG iteration applied to $P^{-1}A$ converges quickly. This condition can be relaxed thanks to the Axelsson-Lindskög theorem [7] that we report informally below.

Informally speaking, the Axelsson-Lindskög theorem says that if A has all the eigenvalues in the interval $[\alpha, \beta]$ where $0 < \alpha < 1 < \beta$ except for q eigenvalues (outliers) which stay outside this interval and are greater than β, then the residual error after $i > q$ steps is bounded by $\gamma_1 \theta_1^{i-q}$ for $\theta_1 = (\sqrt{\mu_1} - 1)/(\sqrt{\mu_1} + 1)$, for $\mu_1 = \beta/\alpha$.

This way, no matter if the condition number of the matrix is large if this largeness is due to a few outliers. In fact, it is enough to find a preconditioner P for which the spectrum of $P^{-1}A$ is "mostly" clustered around 1.

Well, it is possible to show that choosing P in a suitable way inside any trigonometric algebra described in Section 4, one can satisfy the three properties reported above, no matter if the symbol associated with the Toeplitz matrix has some isolated zero. In fact, in this case, even though $\lim_{n\to\infty} \mu(A_n) = \infty$ since zero is an accumulation point for the set of eigenvalues of A_n, for the preconditioned matrix $P_n^{-1}A_n$ the number of eigenvalues that do not lie in the interval $[1-\epsilon, 1+\epsilon]$ is negligible with respect to n.

This makes PCG a very effective method for the approximated solution of large Toeplitz systems whose cost is $O(n \log_2 n)$. In the case of $n \times n$ block Toeplitz matrices with $m \times m$ blocks PCG can be used as well; the cost per step is $O(mn \log mn)$ ops, but unfortunately, if the bivariate symbol is zero in some points of its domain, then the number of outliers for the preconditioned matrix is $O(m + n)$. This makes this technique less effective for the multidimensional case.

More information in this regard together with pointers to the literature can be found in [39], and in the survey [67]. See also [68], [69], [70], the seminal paper by Gilbert Strang [72], and the classical results on the circulant preconditioners of [30], [33], [32].

We show a simple example of preconditioning. Consider the $n \times n$ band Toeplitz matrix A_n associated with the symbol $a(\theta) = 6 - 2(4\cos\theta - \cos(2\theta))$. Observe that $a(\theta) = 0$ for $\theta = 0$ with high multiplicity. Choose as preconditioner the circulant matrix P_n which minimizes $\|A_n - P_n\|_F$ where $\|\cdot\|_F$ is the Frobenius norm. The eigenvalues of A_n are distributed as the associated symbol $a(\theta)$. The eigenvalues of the preconditioned matrix are clustered around 1 as it is evident from Figure 6.1 where we plot the logarithms of the eigenvalues of $P^{-1}A$ in blue and of A in red.

7. Wiener-Hopf factorization and matrix equations

Here we outline an algorithm for solving matrix equations through the solution of an infinite Toeplitz system. For more details, we refer the reader to the papers [18] and to the books [15], [17].

Consider the equations

$$BX^2 + AX + C = 0, \quad CY^2 + AY + B = 0, \tag{7.1}$$

where we assume that A, B, C are $n \times n$ matrices and that there exist solutions X, Y with spectral radius $\rho(X) = \eta < 1$, $\rho(Y) = \nu < 1$. In this case, the matrix Laurent polynomial $\varphi(z) = z^{-1}C + A + zB$ is invertible in the annulus $\mathcal{A} = \{z \in \mathbb{C} : \quad \eta < |z| < \nu^{-1}\}$, that is, $\det \varphi(z) \neq 0$ for $z \in \mathcal{A}$, so that $\psi(z) = \varphi(z)^{-1}$ is analytic over $\mathcal{A}$.

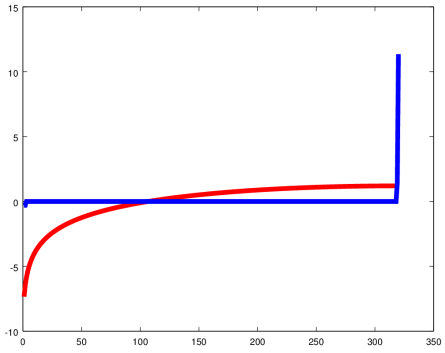

Figure 6.1: Plot of the logarithms of the eigenvalues of $P^{-1}A$ in blue and of A in red

The two equations can be rewritten in terms of infinite block Toeplitz systems. For instance, the first equation takes the form

$$\begin{bmatrix} A & B & & & \\ C & A & B & & \\ & C & A & B & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^2 \\ X^3 \\ \vdots \end{bmatrix} = \begin{bmatrix} -C \\ 0 \\ 0 \\ \vdots \end{bmatrix}.$$

Similarly we can do for the second equation.

This infinite system can be solved by means of the *Cyclic Reduction (CR)* method introduced by Gene Golub (see [18] for bibliographic references and for general properties of CR) for the numerical solution of the discrete Poisson equation over a rectangle and here adjusted to the infinite block Toeplitz case. The CR technique works this way:

- permute block rows and block columns in the above equation by writing the even numbered ones first, followed by the odd numbered ones and get the system

$$\left[\begin{array}{ccc|ccc} A & & & C & B & \\ & A & & & C & \ddots \\ & & \ddots & & & \ddots \\ \hline B & & & A & & \\ C & B & & & A & \\ & \ddots & \ddots & & & \ddots \end{array}\right] \left[\begin{array}{c} X^2 \\ X^4 \\ \vdots \\ \hline X \\ X^3 \\ \vdots \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ \hline -C \\ 0 \\ \vdots \end{array}\right];$$

- eliminate the unknowns $X^2, X^4, \ldots$ by taking a Schur complement and arrive at the system

$$\begin{bmatrix} \widehat{A}_1 & B_1 & & & \\ C_1 & A_1 & B_1 & & \\ & C_1 & A_1 & B_1 & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^3 \\ X^5 \\ \vdots \end{bmatrix} = \begin{bmatrix} -C \\ 0 \\ 0 \\ \vdots \end{bmatrix},$$

where, assuming that A is nonsingular, we have

$$\begin{aligned} A_1 &= A_0 - B_0 A_0^{-1} C_0 - C_0 A_0^{-1} B_0, \\ B_1 &= -B_0 A_0^{-1} B_0, \\ C_1 &= -C_0 A_0^{-1} C_0, \\ \widehat{A}_1 &= \widehat{A}_0 - B_0 A_0^{-1} C_0, \end{aligned}$$

with $A_0 = A, B_0 = B, C_0 = C, \widehat{A}_0 = A$.

This latter system has almost the block Toeplitz structure of the original one except that the $(1,1)$ block is different from the remaining diagonal blocks. Therefore we can repeat the same

procedure by generating the sequence of block triangular systems with blocks C_i, A_i, B_i and $\widehat{A}_i$ such that

$$\begin{bmatrix} \widehat{A}_i & B_i & & & \\ C_i & A_i & B_i & & \\ & C_i & A_i & B_i & \\ & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} X \\ X^{2^i+1} \\ X^{2*2^i+1} \\ X^{3*2^i+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} -C \\ 0 \\ 0 \\ \vdots \end{bmatrix},$$

where

$$\begin{aligned} A_{i+1} &= A_i - B_i A_i^{-1} C_i - C_i A_i^{-1} B_i, \\ B_{i+1} &= -B_i A_i^{-1} B_i, \\ C_{i+1} &= -C_i A_i^{-1} C_i, \\ \widehat{A}_{i+1} &= \widehat{A}_i - B_i A_i^{-1} C_i. \end{aligned}$$

Here, we assume that all the blocks A_i generated this way are nonsingular.

The first equation of this system takes the form

$$\widehat{A}_i X + B_i X^{2^i+1} = -C$$

and it is a nice surprise to find that $\|B_i\| = O(\nu^{2^i})$ so that $X_i = -\widehat{A}_i^{-1} C$ provides an approximation to the solution X with error $O((\nu\eta)^{2^i})$. This makes CR one of the fastest algorithms to solve this kind of matrix equations.

Besides this formulation given in terms of Toeplitz matrices, there is a more elegant formulation given in functional form which provides a generalization of the Graeffe iteration. More precisely, define $\varphi(z) = z^{-1}C_i + A_i + zB_i$ and find that $\varphi_{i+1}(z^2) = \varphi_i(z) A_i^{-1} \varphi_i(-z)$, that is a generalization to the case of matrix polynomials of the celebrated Graeffe-Lobachevsky-Dandelin iteration [61], [62].

Another nice interpretation of CR can be given in terms of the matrix functions $\psi_i(z) = \varphi_i(z)^{-1}$ defined for all the $z \in \mathbb{C}$ where $\varphi_i(z)$ is nonsingular. In fact, one can easily verify that

$$\begin{aligned} \psi_{i+1}(z^2) &= \frac{\psi_i(z) + \psi_i(-z)}{2}, \\ \psi_0(z) &= (z^{-1}C + A + zB)^{-1}. \end{aligned}$$

This formulation enables one to provide the proof of convergence properties just by using the analytic properties of the involved functions. Moreover, the same formulation allows to define the functions $\psi_i(z)$ in the cases where there is a break-down in the construction of the sequence $\varphi_i(z)$ due to the singularity of some A_i.

The solutions G and R of the matrix equations in (7.1) provide the Wiener-Hopf factorization of $\varphi(z)$:

$$\varphi(z) = (I - zR) W (I - z^{-1}G), \quad W = A_0 + A_1 G,$$

which in matrix form takes the following expression

$$\begin{bmatrix} A & B & & \\ C & A & B & \\ & \ddots & \ddots & \ddots \end{bmatrix} = \begin{bmatrix} I & -R & & \\ & I & -R & \\ & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} W & & & \\ & W & & \\ & & \ddots & \end{bmatrix} \begin{bmatrix} I & & & \\ -G & I & & \\ & -G & I & \\ & & \ddots & \ddots \end{bmatrix}.$$

A detailed treatment of this topic can be found in [18].

The same technique can be extended to matrix equations of the kind

$$\sum_{i=-1}^{\infty} A_i X = 0$$

and to the computation of the Wiener-Hopf factorization of the function $A(z) = \sum_{i=-1}^{\infty} z^i A_i$, that is, the block UL factorization of the infinite block Toeplitz matrix in block Hessenberg form associated with $A(z)$.

8. Computing the exponential of a block triangular Toeplitz matrix

In the Erlangian approximation of Markovian fluid queues, one has to compute

$$Y = e^X = \sum_{i=0}^{\infty} \frac{1}{i!} X^i$$

where X is an $(\ell+1)\times(\ell+1)$ block triangular Block Toeplitz matrix with $m \times m$ blocks having negative diagonal entries such that the sum of the entries in each row is nonpositive.

Clearly, since block triangular Toeplitz form a matrix algebra then Y is still block triangular Toeplitz. The question is: what is the most convenient way to compute Y given X in terms of CPU time and error? This question has been faced in [16].

Let $X_0, X_1, \dots X_\ell$ be the blocks defining X. Embed X into an infinite block triangular block Toeplitz matrix X_∞ obtained by completing the sequence X_i with zeros, and denote $Y_0, Y_1, \dots$ the blocks defining $Y_\infty = e^{X_\infty}$. Then Y is the $(\ell+1)\times(\ell+1)$ principal submatrix of Y_∞ we can prove the following decay property

$$\|Y_i\|_\infty \le e^{\alpha(\sigma^{\ell-1}-1)}\sigma^{-i}, \quad \forall \sigma > 1,$$

where $\alpha = \max_j(-(X_0)_{j,j})$, $\beta = \|[X_1, \dots, X_\ell]\|_\infty$.

This property is fundamental to prove error bounds of the following different algorithms

Using ϵ-circulant matrices

Embed X into an ϵ-circulant matrix $X^{(\epsilon)}$ and approximate Y with $Y^{(\epsilon)} = e^{X^{(\epsilon)}}$. We can prove that

$$\|Y - Y^{(\epsilon)}\|_\infty \le e^{|\epsilon|\beta} - 1 = |\epsilon|\beta + O(|\epsilon|^2),$$

and, if ϵ is purely imaginary then

$$\|Y - Y^{(\epsilon)}\|_\infty \le e^{|\epsilon|^2\beta} - 1 = |\epsilon|^2\beta + O(|\epsilon|^4).$$

Using circulant matrices

Embed X into a $K \times K$ block circulant matrix $X^{(K)}$ for $K > \ell$ large, and approximate Y with the $K \times K$ submatrix $Y^{(K)}$ of $e^{X^{(K)}}$.

We can prove the following bound

$$\|[Y_0 - Y_0^{(K)}, \dots, Y_\ell - Y_\ell^{(K)}]\|_\infty \le (e^\beta - 1)e^{\alpha(\sigma^{\ell-1}-1)}\frac{\sigma^{-K+\ell}}{1-\sigma^{-1}}, \quad \sigma > 1.$$

Method based on Taylor expansion

The matrix Y is approximated by truncating the series expansion to r terms

$$Y^{(r)} = \sum_{i=0}^{r} \frac{1}{i!} X^i.$$

Denoting μ the machine precision of the floating point computation, the round-off errors grow as $\epsilon^{-1}\mu$ for the algorithm based on ϵ-circulant and as $\xi K\mu$ for a suitable ξ depending on m, α and β for the algorithm based on circulant embedding.

An experimental analysis shows that the method based on ϵ-circulant is the fastest one and the precision of the approximation is acceptable.

In Figure 8 we report the norm-wise error and the maximum component-wise relative and absolute errors of the result obtained with the ϵ-circulant technique applied to a real problem where the size is $\ell = 512$ and the block size is 2. In Figure 8 we report the same errors for the embedding technique. In Figure 8 we report the norm-wise errors of the method based on ϵ-circulant matrices where the precision is improved with the technique of averaging k different computations with $k = 1, 2, 4$ described in Section 4.5. It is interesting to observe that for $k = 4$ the error keeps decreasing even for $\epsilon \ge 1$. This means that the approximation error is a polynomial of degree less than 8 so that the averaging technique completely removes it. In Figure 8 we report the CPU time.

There are some open issues. In particular, two questions are yet unanswered.

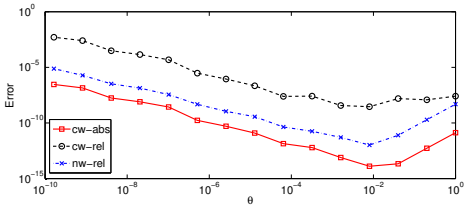

Figure 8.1: Norm-wise error, component-wise relative and absolute errors for the solution obtained with the algorithm based on ϵ-circulant matrices with $\epsilon = \underline{i}\theta$.

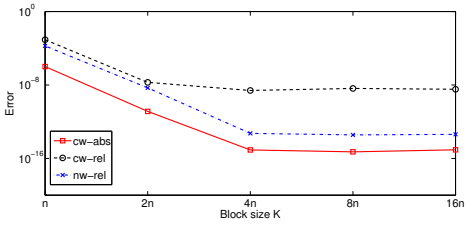

Figure 8.2: Norm-wise error, component-wise relative and absolute errors for the solution obtained with the algorithm based on circulant embedding for different values of the embedding size K.

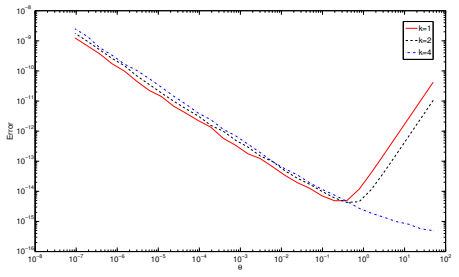

Figure 8.3: Norm-wise errors for the solution obtained with the algorithm based on ϵ-circulant matrices with averaging, for different values of ϵ and for a number $k = 1, 2, 4$ of averaging points.

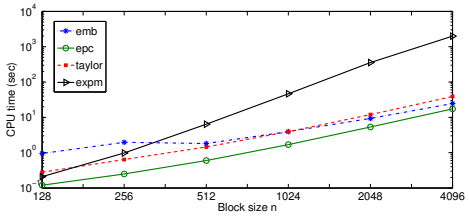

Figure 8.4: CPU time of the Matlab function `expm`, and of the algorithms based on ϵ-circulant, circulant embedding, power series expansion.

- Can we prove that the exponential of a general block Toeplitz matrix does not differ much from a block Toeplitz matrix? Numerical experiments, performed with matrices encountered in fluid queues, confirm this fact but a proof is missing.
- Can we design effective *ad hoc* algorithms for the case of general block Toeplitz matrices?

CHAPTER 2

Rank structures

1. Introduction

Informally speaking, a rank-structured matrix is a matrix where its submatrices located in some part of its support have low rank. An important example of rank structured matrices is the class of *quasi-separable matrices* characterized by the property that all the submatrices strictly contained in the lower triangular part as well as all the submatrices strictly contained in the upper triangular part have rank at most 1. A simple and elementary example of matrices satisfying this property is the set of tridiagonal matrices.

It is also interesting to observe that the inverse of an irreducible tridiagonal matrix A is still quasi-separable. Indeed, this fact is less elementary to prove. Moreover, the upper triangular part of A^{-1} coincides with the upper triangular part of a matrix of rank 1. The same property holds true for the lower triangular part of A^{-1}. That is, there exist vectors u, v, w, z such that $\texttt{tril}(A^{-1}) = \texttt{tril}(uv^T)$, $\texttt{triu}(A^{-1}) = \texttt{triu}(wz^T)$, where `tril` and `triu` denote the lower and the upper triangular part, respectively of a matrix. In this case we say that the quasi-separable matrix A^{-1} has a generator given by the vectors u, v, w, z. Observe that a tridiagonal matrix is quasi-separable but has no generator.

In general, given a pair of integers (h, k) we say that the $n \times n$ matrix A is (h, k)-quasi-separable, or quasi-separable of separability rank (h, k) if all its submatrices strictly contained in the lower triangular part have rank at most h, where at least one submatrix reaches this rank, while all the submatrices strictly contained in the upper triangular part have rank at most k and there exists at least a submatrix with this rank.

Band matrices are an example of (h, k) quasi-separable matrices and it can be proved that their inverses still share this property.

Rank structured matrices are investigated in different fields like integral equations, statistics, vibrational analysis.

There is a very wide literature on this subject. We refer to the paper [78] for a commented bibliography starting from the pioneering papers by Gantmaker and Krein in 1937, and to the recent books [79], [80], [41], [42] with the state of the art in this field. The research activity on rank structured matrices is very intense, there are several international groups which are very active and have produced interesting theoretical and algorithmic advances.

In this chapter, we limit ourselves to describe the problems related to linearizations of polynomials and of matrix polynomials which lead to quasi-separable matrices where recently there has been a growing interest.

2. Basic properties

First we recall, in terms of list, some of the main properties of quasi-separable matrices. For more details we refer to the books [79], [80], [41], [42] and to the current literature. For the sake of simplicity we assume $h = k$, in this case we call the (h, k) quasi-separable matrix A more simply k quasi-separable, or quasi-separable of rank k.

Let A be k quasi-separable. Then

(1) if A is invertible then A^{-1} is k quasi-separable;
(2) if $A = LU$ is the LU factorization of A then L and U are quasi-separable of rank $(k, 0)$ and $(0, k)$, respectively;
(3) if $A = QR$ is a QR factorization of A then Q is quasi-separable of rank k and and U is quasi-separable of rank $(0, 2k)$;

(4) the matrices L_i, U_i, A_i defined by the LR iteration $A_i =: L_iU_i$, $A_{i+1} = U_iL_i$ are quasi-separable of rank $(k,0)$, $(0,k)$, k, respectively.

Moreover, there are algorithms for

(1) computing A^{-1} in $O(nk^2)$ ops;
(2) solving the system $Ax = b$ in $O(nk^2)$ ops;
(3) computing the LU and the QR factorization of A in $O(nk^2)$ ops;

3. Companion matrices

Let $a(x) = \sum_{i=0}^{n} a_i x^i$ be a monic polynomial, i.e., such that $a_n = 1$, with real or complex coefficients. A companion matrix associated with $a(x)$ is a matrix A such that $\det(xI - A) = a(x)$.

Among the most popular companion matrices we recall the first and second Frobenius forms F_1 and F_2 given by ([36])

$$F_1 = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \dots & -a_0 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \quad F_2 = F_1^T,$$

respectively. Both matrices are quasi-separable, F_1 has a generator for the upper triangular part given by the vectors $(1,0,\dots,0)^T$ and $(-a_{n-1},-a_{n-2},\dots,-a_0)$, while F_2 has a generator concerning the lower triangular part. Both matrices can be written as an orthogonal (permutation) matrix, that is the unit circulant, plus a correction of rank 1, that is,

$$F_1 = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{bmatrix} - \begin{bmatrix} a_{n-1} & \dots & a_1 & 1+a_0 \\ 0 & \dots & 0 & 0 \\ \vdots & \dots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \end{bmatrix} =: C - uv^T.$$

The application of the QR iteration to F_1 generates a sequence of matrices A_i, $i = 0,1,2,\dots$, where $A_0 = F_1$, $A_{i+1} = Q_i^* A_i Q_i$, with $Q_i^* Q_i = I$, so that A_i can be still written as orthogonal plus rank 1. In fact, denoting $u_1 = u$, $v_1 = v$, $U_1 = C$, so that $F_1 = U_1 - u_1 v_1^T$, one has $A_i = U_i - u_i v_i^T$ where $U_{i+1} = Q_i^* U_i Q_i$ is still orthogonal and $u_{i+1} = Q_i u_i$, $v_{i+1} = Q_i v_i$. The same property holds true for F_2. This and other similar techniques have been used in [12], [29], [24], [77], [25], [83], [5], [44], [76], [4], for designing effective algorithms for computing the roots of a given polynomial. In fact with this strategy, it is possible to implement the QR step in $O(n)$ ops so that the overall execution of the QR iteration takes $O(n^2)$ ops and is competitive with the classical LAPACK implementation already for small values of the size n.

Another companion matrix is the *comrade matrix* introduced by Barnett [8] in 1975. Define the sequence of orthogonal polynomials satisfying the following three-term recurrence

$$\begin{aligned} &p_0(x) = 1, \quad p_1(x) = x - b_1, \\ &p_{i+1}(x) = (x - b_{i+1})p_i(x) - c_i p_{i-1}(x), \quad i = 1,2,\dots,n-1, \end{aligned} \tag{3.1}$$

where $c_i > 0$. Consider a monic polynomial $p(x)$ represented in this orthogonal basis as $p(x) = \sum_{i=0}^{n} d_i p_i(x)$, where $d_n = 1$. Then one can prove (see [35]) that $p(x) = \det(xI - A)$, where

$$A = \begin{bmatrix} b_1 & c_1 & & \\ 1 & b_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ & & 1 & b_n \end{bmatrix} - [0,\dots,0,1] \begin{bmatrix} d_0 \\ \vdots \\ d_{n-3} \\ \widehat{d}_{n-2} \\ \widehat{d}_{n-1} \end{bmatrix}$$

and $\widehat{d}_{n-2} = -d_{n-2} + c_{n-1}$, $\widehat{d}_{n-1} = -d_{n-1} + b_n$.

Also this matrix A has a quasi-separable structure with rank $(1,2)$. This fact can be exploited to implement a polynomial root-finder for polynomials represented in an orthogonal basis which relies on the QR iteration.

Another companion matrix is the *colleague matrix* (Good 1961 [51], Werner 1983 [82])

$$C = \begin{bmatrix} x_1 & & & & -a_0 \\ 1 & x_2 & & & -a_1 \\ & 1 & \ddots & & \vdots \\ & & \ddots & x_{n-1} & -a_{n-2} \\ & & & 1 & x_n - a_{n-1} \end{bmatrix}.$$

This matrix provides the representation of a polynomial $p(x)$ in the Newton basis. More precisely, one can prove that

$$\begin{aligned} \det(xI - C) =& a_0 + a_1(x - x_1) + a_2(x - x_1)(x - x_2) + \cdots \\ &+ a_{n-1} \prod_{i=1}^{n-1} (x - x_i) + \prod_{i=1}^{n} (x - x_i). \end{aligned}$$

Similarly, given a monic polynomial $p(x)$ of degree n, we may choose n pairwise different values $x_0, x_1, \ldots, x_{n-1}$ and consider the arrowhead companion pencil of size $n+1$ defined by $xC_1 - C_0$ where

$$C_0 = \begin{bmatrix} x_0 & & & & p_0 \\ & x_1 & & & p_1 \\ & & \ddots & \vdots & \\ & & & x_{n-1} & p_{n-1} \\ -\ell_0 & -\ell_1 & \ldots & -\ell_{n-1} & 0 \end{bmatrix}, \quad C_1 = \mathrm{diag}(1, 1, \ldots, 1, 0),$$

$\ell_i = 1/\prod_{j=1,\, j\neq i}^{n-1}(x_i - x_j)$ and $p_i = p(x_i)$.

Computing $\det(xC_1 - C_0)$ by means of the Laplace rule along the last column provides the following expression

$$\det(xC_1 - C_0) = \sum_{i=0}^{n} p_i L_i(x), \quad L_i(x) = \prod_{j=1,\, j\neq i}^{n-1} (x - x_j),$$

that is, the Lagrange representation of the polynomial $p(x)$. Also the pencil $xC_1 - C_0$ is quasi-separable of rank 1.

The Smith companion form given by Smith [71] in 1970 with the goal of locating polynomial roots, and considered by Golub [50] in 1973, has the following form

$$S = \mathrm{diag}(b_1, \ldots, b_n) - ew^T, \quad e = (1, \ldots, 1)^T, \quad w = (w_i),\ w_i = \frac{p(b_i)}{\prod_{j=1,\, j\neq i}^{n}(b_i - b_j)},$$

where $p(x)$ is a monic polynomial of degree n, and $b_1, \ldots, b_n$ are pairwise different numbers.

It is easy to show that $\det(xI - S) = p(x)$, that is, S is a companion matrix for $p(x)$. Also in this case, S is a quasi-separable matrix given in terms of a generator. In fact S is expressed as a diagonal plus a rank 1 matrix.

In [71] the Gerschgorin theorem has been applied to A in order to locate the roots of the polynomial $p(x)$. This property was used in the package MPSolve of [14] to provide guaranteed approximations to the zeros of a polynomial. In [21] this companion form is used for providing a more efficient implementation of MPSolve and to prove that the condition number of the eigenvalues of the Smith companion form converge to zero as the knots b_i converge to the roots of the polynomial, under suitable condition on the type of convergence in case of multiple roots. In fact, from the Smith companion form one can rewrite the condition $p(x) = 0$ in terms of the secular equation

$$\sum_{i=1}^{n} \frac{w_i}{x - b_i} - 1 = 0.$$

The closer the knots $b_1, \ldots, b_n$ to the roots of the polynomial, the better conditioned the roots of $p(x)$ as functions of $w_1, \ldots, w_n$.

The above companion matrices enable one to reduce the polynomial root-finding problem to a (generalized) eigenvalue problem for which matrix based techniques can be applied. We recall that the QR iteration is a reliable numerical technique for computing the eigenvalue of a matrix

A once A has been reduced to upper Hessenberg form $H = (h_{i,j})$ by means of an orthogonal transformation. We recall that an upper Hessenberg matrix H is such that $h_{i,j} = 0$ if $i > j + 1$. Now, some of the companion matrices that we have described (Frobenius and comrade), are already in Hessenberg form so that one can apply directly the QR iteration to the matrix itself. However, the cost of the iteration step is $O(n^2)$ so that the overall cost of the QR iteration is $O(n^3)$. This makes the matrix approach for computing the roots of a polynomial not convenient for moderately large values of the degree n.

However, by exploiting the rank structure of the companion matrices, one can provide implementation of the QR, or the QZ step in $O(n)$ ops. This would provide an algorithm for polynomial root-finding with the cost of $O(n^2)$ which in principle could be competitive with the fixed point iteration techniques, like the Ehrlich-Aberth iteration, which have the same asymptotic cost $O(n^2)$. It must be said that the fastest polynomial root-finding software currently available is MPSolve which relies on the Ehrlich-Aberth iteration as the main engine for refining approximation to the roots. As an example, this package enables one to compute roots of polynomials with degree of the order of millions with a desktop computer in a reasonable time.

In principle, the Ehrlich-Aberth iteration could be replaced by a shifted QR iteration based on the quasi-separable technology. At the moment, the available algorithms for the QR computation do not seem to be still sufficiently effective to make structured QR competitive with Ehrlich-Aberth iteration.

4. Extension to matrix polynomials

In the last decade, much interest has been focused on the properties of matrix polynomials and on their linearizations. The motivations are given by the nice and interesting theoretical properties and by the fact that linearizations of matrix polynomials are the main tool for the numerical solution of the polynomial eigenvalue problems encountered in many applications.

Given $m \times m$ matrices A_i, $i = 0, \ldots, A_n$, with $A_n \neq 0$, we call $A(x) = \sum_{i=0}^{n} x^i A_i$ a matrix polynomial of degree n. The polynomial eigenvalue problem consists in computing the solutions of the polynomial equation $\det A(x) = 0$, given the matrix coefficients of $A(x)$. Throughout, we assume that $A(x)$ is regular, that is $\det A(x)$ is not constant.

In the previous section we have given examples of companion matrices which reduce the polynomial root-finding problem to a matrix eigenvalue problem. Similarly, here we show that we may associate with an $m \times m$ matrix polynomial $A(x)$ of degree n a linear matrix pencil $\mathcal{A}(x) = x\mathcal{A}_1 - \mathcal{A}_0$ of size $mn \times mn$ whose eigenvalues are the solutions of the polynomial eigenvalue problem, that is $\det A(x) = \det \mathcal{A}(x)$.

In fact, it is easy to show [48] that the generalization of the Frobenius matrix is given by the pencil

$$\mathcal{A}(x) = x \begin{bmatrix} I & & & \\ & \ddots & & \\ & & I & \\ & & & A_n \end{bmatrix} - \begin{bmatrix} -A_{n-1} & -A_{n-2} & \dots & -A_0 \\ I & 0 & & \\ & \ddots & \ddots & \\ & & I & 0 \end{bmatrix}.$$

Observe that if A_n is singular then $\det A(x)$ has degree less than mn and there are less than mn eigenvalues counted with their multiplicity. For this reason we prefer to say that even in the case where $\det A_n = 0$, the matrix polynomial, as well as the matrix pencil, still has mn eigenvalues by adding some eigenvalues to the infinity.

This block companion pencil has the quasi-separable structure of rank m and some methods for executing the QR iteration applied to the linear pencil in $O(n^2m^3)$ ops have been designed by relying on the quasi-separable structure.

Similarly, we can extend to matrix polynomials the colleague and the comrade companion [35]. In fact the pencil

$$\mathcal{A}(x) = \begin{bmatrix} (x-x_1)I & & & & A_0 \\ -I & (x-x_2)I & & & A_1 \\ & -I & \ddots & & \vdots \\ & & \ddots & (x-x_{n-1})I & A_{n-2} \\ & & & -I & (x-x_n)A_n + A_{n-1} \end{bmatrix}$$

is such that $\det \mathcal{A}(x) = \det A(x)$, thus provides an extension of the colleague pencil to matrix polynomials.

Similarly, representing $A(x)$ in the basis formed by the orthogonal monic polynomials $p_i(x)$, $i = 0, \ldots, n$ defined in (3.1), such that $A(x) = \sum_{i=0}^{n} D_i p_i(x)$, then the extension of the comrade pencil is

$$\mathcal{A}(x) = x\mathrm{diag}(I, \ldots, I, D_n) - \begin{bmatrix} b_1 I & c_1 I & & \\ I & b_2 I & \ddots & \\ & \ddots & \ddots & c_{n-1} I \\ & & I & b_n I \end{bmatrix} + [0, \ldots, 0, I] \begin{bmatrix} D_0 \\ \vdots \\ D_{n-3} \\ \widehat{D}_{n-2} \\ \widehat{D}_{n-1} \end{bmatrix},$$

where $\widehat{D}_{n-1} = -D_{n-1} + b_n D_n$ and $\widehat{D}_{n-2} = -D_{n-2} + c_{n-2} D_n$ [35]. That is, one can prove that $\det A(x) = \det \mathcal{A}(x)$.

Concerning the Smith companion, we report a recent result obtained in [22].

A first generalization of the Smith companion form, valid for polynomials with scalar coefficients, relies on replacing the linear terms $x - b_i$ with higher degree polynomials. More precisely, let $b_i(x)$ be polynomials of degree d_i for $i = 1, \ldots, k$ such that $\sum_{i=1}^{k} d_i = n$ and $\gcd(b_i(x), b_j(x)) = 1$ for $i \neq j$. Define

$$b(x) = \prod_{i=1}^{k} b_i(x), \quad c_i(x) = \prod_{j=1,\, j\neq i} b_j(x)$$

so that, in view of the Chinese remainder theorem, there exists unique the decomposition

$$\begin{aligned} p(x) &= b(x) + \sum_{i=1}^{k} w_i(x) c_i(x), \\ w_i(x) &= p(x)/c_i(x) \mod b_i(x). \end{aligned}$$

Consequently,

$$p(x) = \det P(x), \quad P(x) = \begin{bmatrix} b_1(x) & & \\ & \ddots & \\ & & b_k(x) \end{bmatrix} + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} [w_1(x), \ldots, w_k(x)].$$

This way, the original problem of finding the roots of a polynomial has been reduced to solving a polynomial eigenvalue problem for a $k \times k$ matrix polynomial. Moreover, the coefficients of this polynomial are quasi-separable matrices.

Observe that if $k = n$, then $d_i = 1$, $b_i(x) = x - b_i$ and we get the Smith companion form. It is possible to provide an explicit expression of the left and the right eigenvectors of this matrix polynomial.

A further extension concern the case where $A(x) = \sum_{i=0}^{n} x^i A_i$ is a nondegenerate matrix polynomial of degree n. In this case, let us consider once again the pairwise prime scalar polynomials $b_i(x)$, for $i = 1, \ldots, k$, which we assume monic, such that their degrees d_i sum up to n. Define $B_i(x) = b_i(x)I$, for $i = 1, \ldots, k-1$, and set $B_k(x) = b_k(x)I + sI$ where $s \in \mathbb{C}$ is such that $\det B_k(x) \neq 0$ when x is zero of any $b_i(x)$.

Then there exists, unique, the decomposition

$$A(x) = B(x) + \sum_{i=1}^{k} W_i(x) C_i(x),$$

where $B(x) = \prod_{i=1}^{k} B_i(x)$, $\quad C_i(x) = \prod_{j=1,\, j\neq i}^{k} B_j(x)$ and

$$W_i(x) = \frac{A(x)}{\prod_{j=1,\, j\neq i}^{k-1} b_j(x)} B_k(x)^{-1} \mod b_i(x), \quad i = 1, \ldots, k-1,$$

$$W_k(x) = \frac{A(x)}{\prod_{j=1}^{k-1} b_j(x)} - sI - s \sum_{j=1}^{k-1} \frac{W_j(x)}{b_j(x)} \mod b_k(x).$$

Moreover,

$$\det A(x) = \det \mathcal{A}(x), \quad \mathcal{A}(x) = D(x) + \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} [W_1(x), \ldots, W_k(x)],$$

where

$$D(x) = \begin{bmatrix} b_1(x)I & & & \\ & \ddots & & \\ & & b_{k-1}(x)I & \\ & & & b_k(x)A_n + sI \end{bmatrix}.$$

This expression extends to the case of matrix polynomials the Smith companion given in the generalized form in terms of the pairwise prime polynomials $b_i(x)$.

Let us comment on some features of this reduction. Assume for simplicity $A_n = I$ and $k = n$, and choose $b_i(x) = x - \omega_n^i$, where $\omega_n = \cos\frac{2\pi}{n} + \underline{i}\sin\frac{2\pi}{n}$ is a primitive nth root of 1. Then, using the Fourier matrix $F_n = \frac{1}{\sqrt{n}}(\omega_n^{i,j})_{i,j=1,n}$, one can verify that

$$(F_n^* \otimes I)\mathcal{A}(x)(F_n \otimes I) = xI - C$$

where C is the block Frobenius matrix associated with $A(x)$. On the other hand, by choosing $b_i(x) = x - \alpha\omega_n^i$, then we get a pencil which is unitarily similar to the block companion matrix scaled with the diagonal matrix $\mathrm{diag}(1, \alpha, \ldots, \alpha^{n-1}) \otimes I$. This means that the condition number of the eigenvalues of the extended Smith companion matrix is not worse than that of the Frobenius, or that of the scaled Frobenius matrix.

The choice of suitable values of b_i provides linearizations of the matrix polynomial where the eigenvalues are much better conditioned. We will give numerical evidence of this property in the next section where we report on the results of some numerical experiments.

This kind of reduction of an $m \times m$ polynomial eigenvalue problem of degree n to a lower degree matrix polynomial, belongs to the so called class of ℓ-ifications introduced in [37]. That is, one can prove that there exist unimodular $mk \times mk$ matrix polynomial $E(x)$, $F(x)$ such that

$$E(x)\mathcal{A}(x)F(x) = I_{mk-k} \oplus A(x).$$

Moreover, if $d_1 = \cdots = d_k$ then there exist unimodular $mk \times mk$ matrix polynomial $\widehat{E}(x)$, $\widehat{F}(x)$ such that

$$\widehat{E}(x)\mathcal{A}^{\#}(x)\widehat{F}(x) = I_{mk-k} \oplus A^{\#}(x)$$

where $A^{\#}(x) = \sum_{i=0}^{n} A_{n-i}x^i$ denotes the "reversed polynomial". That is, if the $b_i(x)$ have the same degree then $\mathcal{A}(x)$ is a strong ℓ-ification of $A(x)$ in the sense of [37].

The eigenvectors of the matrix polynomial $\mathcal{A}(x)$ can be explicitly given. In fact, if $A(\lambda)v = 0$, $v \neq 0$, then

$$\begin{bmatrix} \prod_{j\neq 1} B_j(\lambda)v \\ \vdots \\ \prod_{j\neq k} B_j(\lambda)v \end{bmatrix}$$

is a right eigenvector of $\mathcal{A}(x)$ corresponding to λ. Moreover, If $u^T A(\lambda) = 0$ then

$$\left[u^T W_1 \prod_{j\neq 1} B_j(\lambda), \ldots, u^T W_k \prod_{j\neq k} B_j(\lambda) \right]$$

is a left eigenvector of $\mathcal{A}(x)$ corresponding to λ.

It is interesting to observe that if $L = \begin{bmatrix} I & & & \\ -I & \ddots & & \\ & \ddots & \ddots & \\ & & -I & I \end{bmatrix}$ then the matrix $L\mathcal{A}(x)$ takes the block Hessenberg form

$$L\mathcal{A}(x) = \begin{bmatrix} B_1(x)+W_1(x) & W_2(x) & \ldots & W_{k-1}(x) & W_k(x) \\ -B_1(x) & B_2(x) & & & \\ & -B_2(x) & \ddots & & \\ & & \ddots & B_{k-1}(x) & \\ & & & -B_{k-1}(x) & B_k(x) \end{bmatrix}.$$

5. Numerical properties

Here, we present some numerical experiments to show that in many interesting cases a careful choice of the $B_i(x)$ can lead to linearizations where the eigenvalues are much better conditioned than in the original problem.

5.1. **Scalar polynomials.** As a first example, consider a monic scalar polynomial $p(x) = \sum_{i=0}^{n} p_i x^i$ where the coefficients p_i have unbalanced moduli. In this case, we generate p_i using the MATLAB command `p = exp(12 * randn(1,n+1)); p(n+1)=1;`

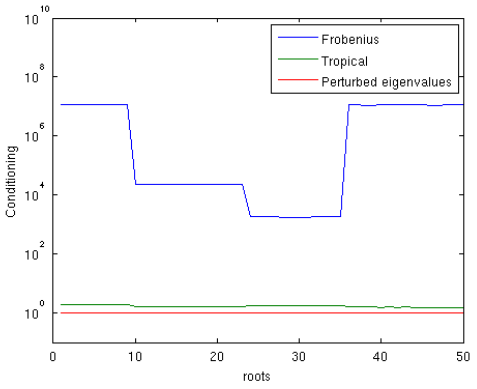

Figure 5.1: Conditioning of different linearizations of a scalar polynomial of degree 50 having random unbalanced coefficients.

Then from $p(x)$ we build the companion forms and compute the condition number of the eigenvalues by means of the Matlab function `condeig()`.

We have considered three different linearizations:

- The Frobenius linearization obtained by `compan(p)`;
- the Smith companion matrix obtained by taking as b_i some perturbed values of the roots;
- the Smith companion with nodes given by the tropical roots of the polynomial multiplied by unit complex numbers.

The results are displayed in Figure 5.1. On can see that in the first case the condition numbers of the eigenvalues are much different from each other and can be as large as 10^{13} for the worst conditioned eigenvalue. In the second case the condition number of all the eigenvalues is close to 1, while in the third linearization the condition numbers are much smaller than those of the Frobenius linearization and have an almost uniform distribution.

These experimental results are a direct verification of a conditioning result of [21, Sect. 5.2] that is at the basis of the MPSolve algorithm presented in that paper.

5.2. **The matrix case.** Consider now a matrix polynomial $P(x) = \sum_{i=0}^{n} P_i x^i$ where for simplicity we assume $P_n = I$. As a first example, consider the case where the coefficients P_i have unbalanced norms.

We can give reasonable estimates to the modulus of the eigenvalues using the Pellet theorem or the tropical roots (see [19, 46, 64], for some insight on these tools).

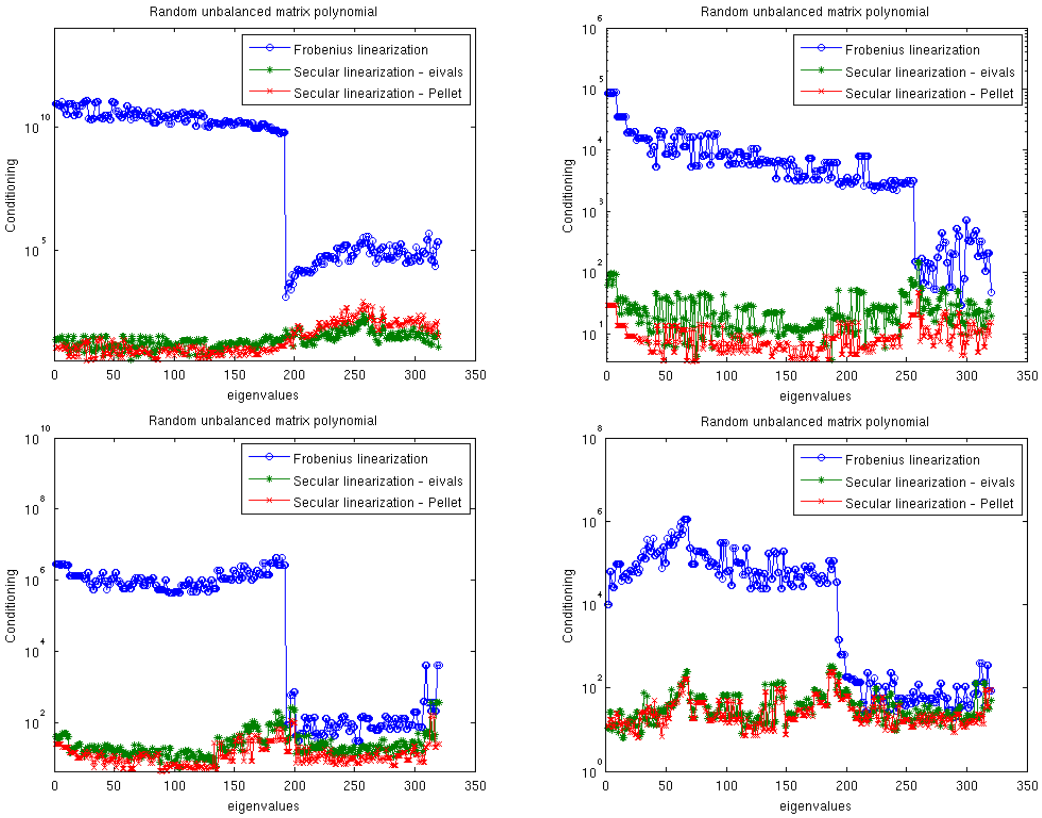

Figure 5.2: Conditioning of different linearizations for some matrix polynomials with random coefficients having unbalanced norms.

We have considered three linearizations: the standard Frobenius companion matrix, and two versions of the extended Smith companion form. In the first version the nodes b_i are the mean of the moduli of set of eigenvalues with close moduli multiplied by unitary complex numbers. In the second, the values of b_i are obtained by the Pellet estimates delivered by the tropical roots.

In Figure 5.2 we report the conditioning of the eigenvalues, measured with Matlab's `condeig` for all these linearizations.

It is interesting to note that the conditioning of the eigenvalues of the Smith companion is, in every case, not exceeding 10^2. Moreover it can be observed that no improvement is obtained on the conditioning of the eigenvalues that are already well-conditioned. In contrast, there is a clear improvement on the ill-conditioned ones. In this particular case, this class of linearizations seems to give an almost uniform bound to the condition number of all the eigenvalues.

Further examples come from the NLEVP collection of [9] where we have selected some problems that exhibit bad conditioning.

As a first example we consider the problem `orr_sommerfeld`. Using the tropical roots we can find some values inside the unique annulus that is identified by the Pellet theorem. In this example the values obtained only give a partial picture of the eigenvalues distribution. The Pellet theorem gives about `1.65e-4` and `5.34` as lower and upper bound to the moduli of the eigenvalues, but the

tropical roots are rather small and near to the lower bound. More precisely, the tropical roots are `1.4e-3` and `1.7e-4` with multiplicities 3 and 1, respectively.

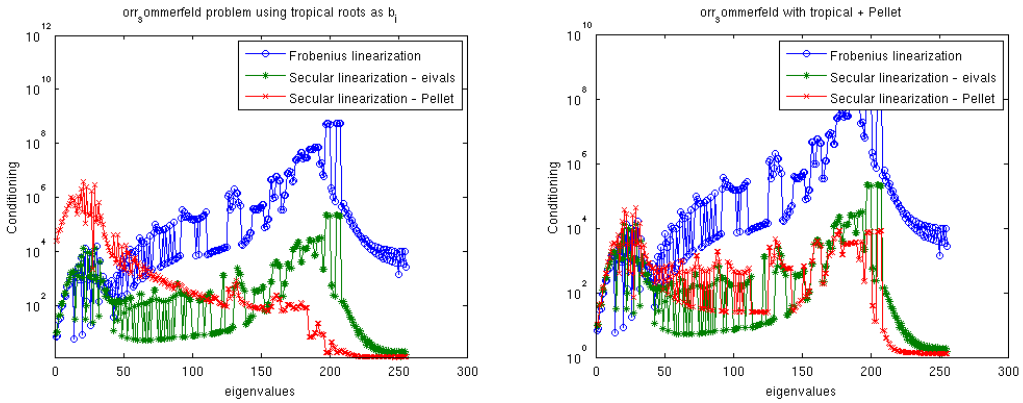

Figure 5.3: On the left we report the conditioning of the Frobenius and of the Smith companion matrices with the choices of b_i as block mean of eigenvalues and as the estimates given by the tropical roots. On the right the tropical roots are coupled with estimates given by the Pellet theorem.

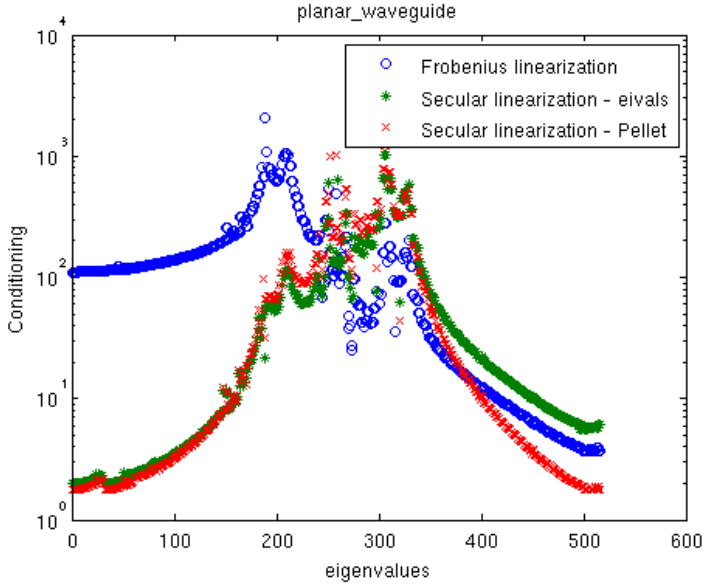

Figure 5.4: Conditioning of the eigenvalue problem for three different linearizations on the `planar_waveguide` problem.

This leads to a linearization $\mathcal{A}(x)$ that is well-conditioned for the smaller eigenvalues but with a higher conditioning on the eigenvalues of bigger modulus as can be seen in Figure 5.3 on the left (the eigenvalues are ordered in nonincreasing order with respect to their modulus). It can be seen, though, that coupling the tropical roots with the standard Pellet theorem and altering the b_i by adding a value slightly smaller than the upper bound (in this example we have chosen 5 but the result is not very sensitive to this choice) leads to a much better result that is reported in Figure 5.3 on the right. In the right figure we have used `b = [ 1.7e-4, 1.4e-3, -1.4e-3, 5 ]`. This seems to justify that there exists a link between the quality of the approximations obtained through the tropical roots and the conditioning properties of the Smith companion.

We analyzed another example problem from the NLEVP collection called `planar_waveguide`. The results are shown in Figure 5.2. This problem is a PEP of degree 4 with two tropical roots

approximately equal to 127.9 and 1.24. Again, it can be seen that for the eigenvalues of smaller modulus (that will be near the tropical root 1.24) the Frobenius and the Smith companion behave in the same way, whilst for the bigger ones the Smith companion has some advantage in the conditioning. This may be justified by the fact that the Frobenius linearization is similar to the Smith companion on the roots of the unity.

Note that in this case the information obtained by the tropical roots seems more accurate than in the `orr_sommerfeld` case, so the Smith companion built using the tropical roots and the one built using the block-mean of the eigenvalues behave approximately in the same way.

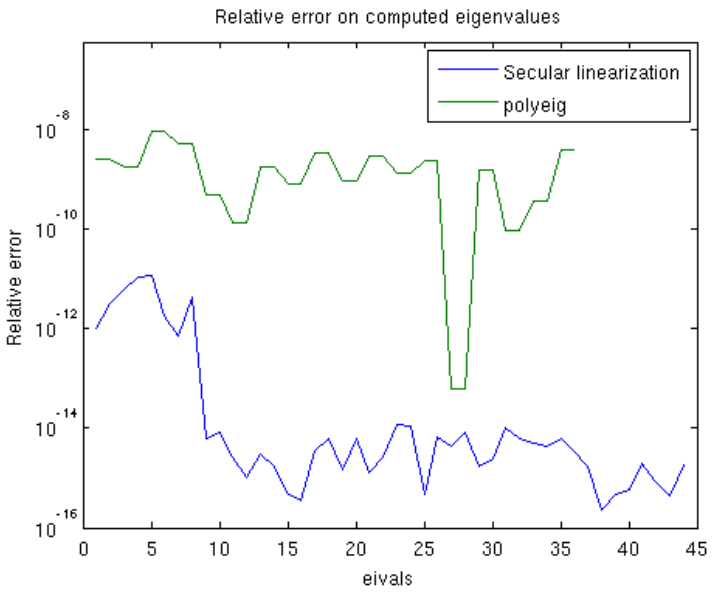

Figure 5.5: The accuracy of the computed eigenvalues using polyeig and the extended Smith companion matrix with the b_i obtained through the computation of the tropical roots.

As a last example, we have tried to find the eigenvalues of a matrix polynomial defined by integer coefficients. We have used polyeig and the Smith companion matrix (using the tropical roots as b_i) and the QZ method. We have chosen the polynomial $P(x) = P_{11}x^{11} + P_9x^9 + P_2x^2 + P_0$ where

$$P_{11} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 \\ & & 1 & 1 \\ & & & 1 \end{bmatrix}, \; P_9 = 10^8 \begin{bmatrix} 3 & 1 & & \\ 1 & 3 & 1 & \\ & 1 & 3 & 1 \\ & & 1 & 3 \end{bmatrix}, \; P_2 = 10^8 P_{11}^T, \; P_0 = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 3 & \\ & & & 4 \end{bmatrix}$$

In this case the tropical roots are good estimates of the blocks of eigenvalues of the matrix polynomial. We obtain the tropical roots $1.2664 \cdot 10^4$, 0.9347 and $1.1786 \cdot 10^{-4}$ with multiplicities 2, 7 and 2, respectively. We have computed the eigenvalues with a higher precision and we have compared them with the results of `polyeig` and of `eig` applied to the Smith companion matrix. Here, the Smith companion has been computed with the standard floating point arithmetic. As shown in Figure 5.5 we have achieved much better accuracy with the latter choice. The generalized Smith companion matrix has achieved a relative error of the order of the machine precision on all the eigenvalues except the smaller block (with modulus about 10^{-4}). In that case the relative error is about 10^{-12} but the absolute error is, again, of the order of the machine precision.

References

[1] A. H. Al-Mohy and N. J. Higham. The complex step approximation to the Fréchet derivative of a matrix function. *Numer. Algorithms*, 53:113–148 (2010).

[2] A. Amiraslani, R. M. Corless, and P. Lancaster. Linearization of matrix polynomials expressed in polynomial bases. *IMA J. Numer. Anal.*, 29(1):141–157, 2009.

[3] G.S. Ammar and W.B. Gragg, Superfast solution of real positive definite Toeplitz systems, SIAM J. Matrix Anal. Appl. 9, 61–76 (1988).

[4] J.L. Aurentz, T. Mach, R. Vandebril, D.S. Watkins, Fast and backward stable computation of roots of polynomials, TW654, 36 pages, Department of Computer Science, KU Leuven, Leuven, Belgium, August 2014

[5] J.L. Aurentz, R. Vandebril, D.S. Watkins, Fast computation of eigenvalues of companion, comrade, and related matrices, BIT Numer Math 54, 7–30 (2014).

[6] F. Avram, On bilinear forms on Gaussian random variables and Toeplitz matrices, Probab. Theory Related Fields 79, 37–45 (1988).

[7] O. Axelsson and G. Lindskög, On the rate of convergence of the preconditioned conjugate gradient method, Numerische Mathematik, 48, 499–523, 1986.

[8] S. Barnett, A companion matrix analogue for orthogonal polynomials, Linear Algebra Appl., 12, 197–202 (1975).

[9] T. Betcke, N.J. Higham, V. Mehrmann, C. Schröder, and F. Tisseur. NLEVP: A collection of nonlinear eigenvalue problems. `http://www.mims.manchester.ac.uk/research/numerical-analysis/nlevp.html`.

[10] D. Bini. Parallel solution of certain Toeplitz linear systems. *SIAM J. Comput.*, 13, 268–276 (1984).

[11] D. Bini, M. Capovani, Spectral and computational properties of band symmetric Toeplitz matrices, Linear Algebra Appl. 52/53, 99–126 (1983).

[12] D.A. Bini, Y. Eidelman, L. Gemignani, I. Gohberg, Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices SIAM J. Matrix Anal. Appl., 29, 566–585 (2007).

[13] D. Bini, P. Favati, On a matrix algebra related to the discrete Hartley transform, SIAM J. Matrix Anal. Appl., 14, 500–507 (1993).

[14] D.A. Bini, G. Fiorentino, Design, Analysis and Implementation of a Multiprecision Polynomial Rootfinder, Numer. Algorithms, 23, 127–219 (2000).

[15] D.A. Bini, B. Iannazzo, B. Meini, Numerical Solution of Algebraic Riccati Equations, Fundamentals of Algorithms, SIAM, Philadelphia 2012.

[16] D. A. Bini, S. Dendievel, G. Latouche, and B. Meini. Computing the Exponential of Large Block-Triangular Block-Toeplitz Matrices Encountered in Fluid Queues, Linear Algebra Appl., in Press, doi:10.1016/j.laa.2015.03.035.

[17] D. A. Bini, G. Latouche, and B. Meini. *Numerical methods for structured Markov chains*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005. Oxford Science Publications.

[18] D.A. Bini, B. Meini, The cyclic reduction algorithm: from Poisson equation to stochastic processes and beyond, Numerical Algorithms, 51, 23–60 (2009).

[19] D. A. Bini, V. Noferini, and M. Sharify. Locating the eigenvalues of matrix polynomials. SIAM J. Matrix Anal. Appl., 34, 1708–1727 (2013).

[20] D. Bini and V. Pan. *Polynomial and Matrix Computations*. Birkhäuser, 1994.

[21] D.A. Bini, L. Robol, Solving secular and polynomial equations: A multiprecision algorithm. J. Comput. Appl. Math., 272, 276–292 (2014).

[22] D.A. Bini, L. Robol, On a class of matrix pencils and ℓ-ifications equivalent to a given matrix polynomial. Linear Algebra Appl., in Press, doi:10.1016/j.laa.2015.07.017, arXiv:1406.1025 (2014).

[23] R.R. Bitmead, B.D.O. Anderson, Asymptotically fast solution of Toeplitz and related systems of linear equations. Linear Algebra Appl., 34, 103–116 (1980).

[24] P. Boito, Y. Eidelman, L. Gemignani, Implicit QR for Rank-Structured Matrix Pencils, BIT Numerical Mathematics 54, 85–111 (2014).

[25] P. Boito, Y. Eidelman, L. Gemignani, I. Gohberg, Implicit QR with Compression, Indag. Math. 23, 733–761 (2012).

[26] A. Böttcher, S.M. Grudsky, Toeplitz Matrices, Asymptotic Linear Algebra and Functional Analysis. Text and Readings in Mathematics 18, Hindustan Book Agency, New Delhi 2000.

[27] A. Böttcher, B. Silbermann, *Introduction to Large Truncated Toeplitz Matrices*, Springer, New York 1999.

[28] A. Böttcher, I. Spitkovsky, The factorization problem: Some known results and open questions, Operator Theory: Advances and App., 229, 101–122 (013).

[29] S. Chandrasekaran, M. Gu, J. Xia, J. Zhu, A fast QR algorithm for companion matrices, Recent Advances in Matrix and Operator Theory, Oper. Theory Adv. Appl., vol. 179, Birkhüser, Basel (2008).

[30] R. Chan, Circulant preconditioners for Hermitian Toeplitz systems, SIAM J. Matrix Anal. Appl. 10, 542–550 (1989).
[31] R. H.-F. Chan and X.-Q. Jin. *An introduction to iterative Toeplitz solvers*, volume 5 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.
[32] T. F. Chan. An optimal circulant preconditioner for Toeplitz systems. *SIAM J. Sci. Statist. Comput.*, 9(4):766–771, 1988.
[33] R. Chan, G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner, SIAM J. Sci. Statist. Comput. 10, 104–119 (1989).
[34] R.M. Corless. Generalized companion matrices in the Lagrange basis. In L. Gonzalez-Vega and T. Recio, editors, *Proceedings EACA*, June 2004.
[35] R.M. Corless and G. Litt, Generalized Companion Matrices for Polynomials not expressed in Monomial Bases, Ontario Research Centre for Computer Algebra, Department of Applied Mathematics University of Western Ontario `http://www.apmaths.uwo.ca/~rcorless/frames/PAPERS/PABV/CMP.pdf`.
[36] F. De Terán, F.M. Dopico, J. Pérez, Backward stability of polynomial root-finding using Fiedler companion matrices, IMA J. Numer. Anal. (2014).
[37] F. De Terán, F.M. Dopico, D.S. Mackey, Fiedler companion linearizations, for rectangular matrix polynomials. Linear Algebra Appl., 437, 957–991 (2012).
[38] S. Delvaux, M. Van Barel, A Hessenberg reduction algorithm for rank structured matrices, SIAM J. on Matrix Analysis and App., 29, 895–926 (2007).
[39] F. Di Benedetto, Gram matrices of fast algebras have a rank structure, SIAM J. Matrix Anal. Appl. 31, 526–545 (2009).
[40] Y. Eidelman, L. Gemignani, and I. Gohberg. Efficient eigenvalue computation for quasiseparable Hermitian matrices under low rank perturbations. Numer. Algorithms, 47, 253–273 (2008).
[41] Y. Eidelman, I. Gohberg, and I. Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 1*, volume 234 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer, Basel, 2014. Basics. Completion problems. Multiplication and inversion algorithms.
[42] Y. Eidelman, I. Gohberg, and I. Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 2*, volume 235 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer Basel AG, Basel, 2014. Eigenvalue method.
[43] M. Fiedler, A note on companion matrices. *Linear Algebra and Its Applications*, 372:325–331, 2003.
[44] K. Frederix, S. Delvaux, M. Van Barel, An algorithm for computing the eigenvalues of block companion matrices, Numerical Algorithms, volume 62, 261–287 (2013).
[45] F.R. Gantmacher, M.G. Krein, Sur les matrices complètement non négatives et oscillatoires. Compositio Mathematica 4, 445–476 (1937).
[46] S. Gaubert and M. Sharify, Tropical scaling of polynomial matrices. In *Positive systems*, volume 389 of *Lecture Notes in Control and Inform. Sci.*, 291–303. Springer, Berlin, 2009.
[47] I. Gohberg, T. Kailath, V. Olshevsky (1995). "Fast Gaussian elimination with partial pivoting for matrices with displacement structure". Mathematics of Computation 64 (212): 1557–1576.
[48] I. Gohberg, P. Lancaster, and L. Rodman. Matrix polynomials. Academic Press Inc., New York, 1982. Computer Science and Applied Mathematics.
[49] I. Gohberg and A. Semencul, On the inversion of finite Toeplitz matrices and their continuous analogs (in Russian), Mat. Issled 7, 201–223 (1972).
[50] G.H. Golub, Some modified matrix eigenvalue problems, SIAM Review, 15, 318–334 (1973).
[51] I.J. Good, The colleague matrix, a Chebyshev analogue of the companion matrix Quart. J. Math., Oxf. Ser., 12, 61–68 (1961).
[52] T.NT Goodman, C. A. Micchelli, G. Rodriguez, S. Seatzu, Spectral factorization of Laurent polynomials, Advances in Computational Mathematics, 7 429–454 (1997).
[53] U. Grenander, G. Szegő. Toeplitz Forms and Their Applications. Second Edition, Chelsea, New York, 1984.
[54] N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.
[55] T. Kailath, A H. Sayed, Displacement Structure: theory and applications, SIAM Review, 297–386, 1995.
[56] T. Kailath and A. H. Sayed, eds., Fast Reliable Algorithms for Matrices with Structure, SIAM Press, 1999.
[57] T. Kailath, S. Y. Kung and M. Morf, Displacement ranks of matrices and linear equations, J. Math. Appl. 68, 395-407 (1979).
[58] F.-R. Lin, W.-K. Ching, M.K. Ng, Fast inversion of triangular Toeplitz matrices, Theoretical Computer Science, 315, 511–523 (2004).
[59] M.F. Neuts. *Structured stochastic matrices of $M/G/1$ type and their applications*, volume 5 of *Probability: Pure and Applied*. Marcel Dekker, Inc., New York, 1989.
[60] M.F. Neuts. *Matrix-geometric solutions in stochastic models*. Dover Publications, Inc., New York, 1994. An algorithmic approach, Corrected reprint of the 1981 original.
[61] A. Ostrowski, Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent. Chapitres III et IV. Acta Math. 72, 157–257 (1940)
[62] A. Ostrowski, Addition à notre mémoire: "Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent." Acta Math. 75, 183–186 (1943)
[63] S.V. Parter, Extreme eigenvalues of Toeplitz forms and applications to elliptic difference equations, Trans. Amer. Math. Soc., 99 (1966), pp. 153–192.

[64] M.A. Pellet. Sur un mode de séparation des racines des équations et la formule de Lagrange. *Bull. Sci. Math.*, 5:393–395, 1881.
[65] F. Poloni, A note on the $O(n)$-storage implementation of the GKO algorithm and its adaptation to Trummer-like matrices, Numer. Algorithms, 55, 115–139 (2010).
[66] Y. Saad, Iterative methods for sparse linear systems, second Edition, SIAM, Philadelphia 2003.
[67] S. Serra Capizzano, Toeplitz matrices: spectral properties and preconditioning in the CG method, NLA courses at FMB in Uppsala, TR 23 (2014), Information Technology Dept, Uppsala University.
[68] S. Serra Capizzano, Preconditioning strategies for Hermitian Toeplitz systems with nondefinite generating functions, SIAM J. Matrix Anal. Appl., 17, 1007–1019 (1996).
[69] S. Serra Capizzano. Optimal, quasi–optimal and superlinear preconditioners for asymptotically ill–conditioned positive definite Toeplitz systems, Math. Comput. 66, 651–665 (1997).
[70] S. Serra Capizzano, E. Tyrtyshnikov, How to prove that a preconditioner cannot be superlinear, Math. Comput. 1305-1316, (2003).
[71] B.T. Smith, Error bounds for zeros of a polynomial based upon Gerschgorin's theorems, J. Assoc. Comput. Math. 17, 661–674 (1970).
[72] G. Strang, A proposal for Toeplitz matrix calculation, Stnd. Appl. Math. 74, 171–176 (1986).
[73] W.F. Trench, An algorithm for the inversion of finite Toeplitz matrices, SIAM J. Appl. Math. 12, 515–522 (1964).
[74] E.E. Tyrtyshnikov, A Unifying Approach to Some Old and New Theorems on Distribution and Clustering, Linear Algebra Appl., 232 1–43 (1996).
[75] E.E. Tyrtyshnikov, N. Zamarashkin. Spectra of multilevel Toeplitz matrices: advanced theory via simple matrix relationships, Linear Algebra Appl., 270, 15–27 (1998).
[76] M. Van Barel, R. Vandebril, P. Van Dooren, K. Frederix, Implicit double shift QR-algorithm for companion matrices, Numerische Mathematik, 116, 177–212 (2010).
[77] R. Vandebril and G.M. Del Corso. An implicit multishift qr-algorithm for hermitian plus low rank matrices. SIAM Journal on Scientific Computing, 32, 2190–2212 (2010).
[78] R. Vandebril, M. Van Barel, G. Golub, N. Mastronardi, A bibliography on semiseparable matrices, Calcolo 42, 249–270 (2005)
[79] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices, Vol. 1 Linear Systems.* Johns Hopkins, Baltimore, Maryland, 2008.
[80] R. Vandebril, M. Van Barel, N. Mastronardi, *Matrix Computations and Semiseparable Matrices, Vol. 2 Eigenvalue and Singular Value Methods.* Johns Hopkins, Baltimore, Maryland, 2008.
[81] J. Xia, Y. Xi, M. Gu, A superfast structured solver for toeplitz linear systems via randomized sampling, SIAM J. Matrix Anal. Appl., 33, 837–858 (2012).
[82] W. Werner,A generalized companion matrix of a polynomial and some applications, Linear Algebra Appl., 55, 19–36 (1983).
[83] P. Zhlobich, Differential qd algorithm with shifts for rank-structured matrices. SIAM J. Matrix Anal. Appl. 33, 1153–1171 (2012).
[84] S. Zohar, Toeplitz matrix inversion: The algorithm of W.F. Trench, J. Assoc. Comput. Mach. 16, 592–601 (1967).

Dipartimento di Matematica, Università di Pisa, Italy