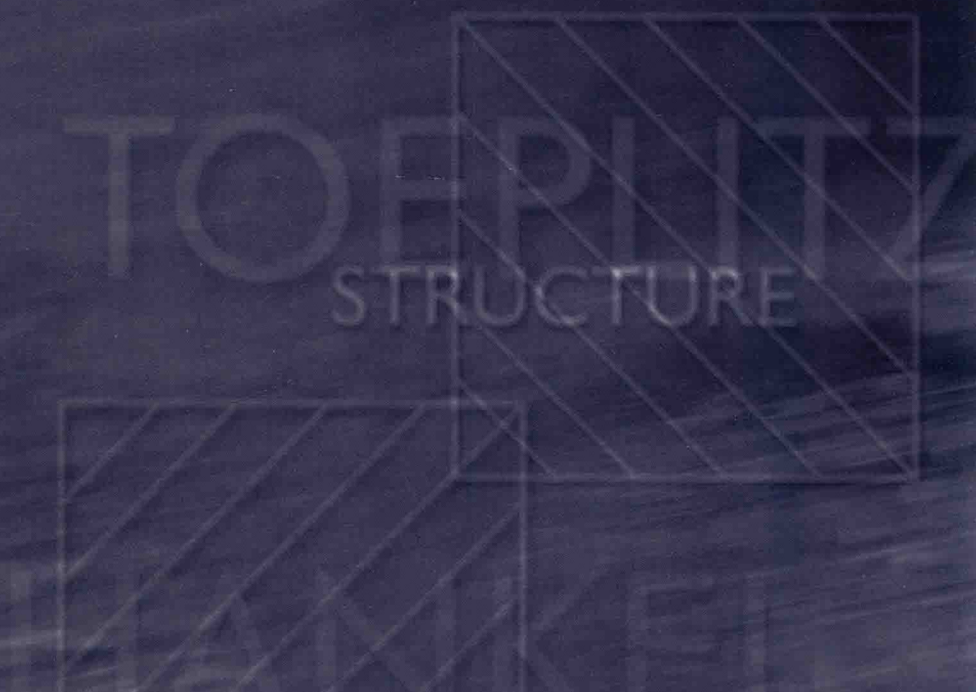
Victor Y. Pan

STRUCTURED MATRICES AND POLYNOMIALS

Unified Superfast Algorithms



BIRKHÄUSER SPRINGER

Victor Y. Pan

Structured Matrices and Polynomials

Unified Superfast Algorithms

BIRKHÄUSER • SPRINGER
BOSTON NEW YORK

Victor Y. Pan
Department of Mathematics
and Computer Science
Lehman College, CUNY
Bronx, NY 10468-1590

Library of Congress Cataloging-in-Publication Data

Pan, Victor.
Structured matrices and polynomials : unified superfast algorithms / Victor Y. Pan.
p. cm.
Includes bibliographical references and index.
ISBN 0-8176-4240-4 (alk. paper) – ISBN 3-7643-4240-4 (alk. paper)
1. Matrices–Data processing. 2. Polynomials–Data processing. I. Title.

QA188.P364 2001
512.9'434–dc21

2001035928
CIP

AMS Subject Classifications: Primary: 12Y05, 68Q40, 65F05, 65F10, 93B40, 47B35
Secondary: 65Y20, 68Q25,15A09, 47A57, 47A40, 93B25, 93B28

Printed on acid-free paper.

Birkhäuser

ISBN 0-8176-4240-4 SPIN 10835910
ISBN 3-7643-4240-4

Typeset by the author
Cover design by Mary Burgess, Cambridge, MA
Printed and bound by Hamilton Printing, Rensselaer, NY
Printed in the United States of America

9 8 7 6 5 4 3 2 1

Structured Matrices and Polynomials

Unified Superfast Algorithms

To My Parents

Preface

Research in mathematics can be viewed as the search for some hidden keys that open numerous scientific locks. In this book, we seek key techniques and ideas to unify various computations with matrices and polynomials, which are the oldest subjects in mathematics and computational mathematics and the backbone of modern computations in the sciences, engineering, and communication.

Four millennia ago Sumerians wrote the solutions of polynomial equations on clay tablets. Later the ancient Egyptians did so on papyrus scrolls [B40], [Bo68]. Likewise, the solution of linear systems of equations, the most popular matrix operation, can also be traced back to ancient times.

How frequently do these subjects enter our life? Much more frequently than one commonly suspects. For instance, every time we turn on our radio, TV, or computer, convolution vectors (polynomial products) are computed. Indeed, all modern communication relies on the computation of convolution and the related operations of discrete Fourier, sine, and cosine transforms. Experts also know that most frequently the practical solution of a scientific or engineering computational problem is achieved by reduction to matrix computations.

Why are our two subjects put together? What do matrices and polynomials have in common?

Mathematicians know that matrices and polynomials can be added and multiplied together. Moreover, matrices endowed with certain structures (for instance, matrices whose entries are invariant in their shifts in the diagonal or antidiagonal directions) share many more common features with polynomials. Some basic operations with polynomials have an equivalent interpretation in terms of operations with structured matrices. A number of examples of this duality is covered in the book.

What is the impact of observing the correlation between matrices and polynomials? Besides the pleasure of having a unified view of two seemingly distinct subject areas, we obtain substantial improvement in modern computations. Structured matrices often appear in computational applications, in many cases along with polynomial computations. As a demonstration, in Sections 3.7–3.9 we cover a simple application of structured matrices to loss-resilient encoding/decoding and their well-known correlation to some celebrated problems of rational interpolation and approximation.

Applied linear algebra versus computer algebra

What is amazing — in view of the close ties between polynomial and structured matrix computations — is that applied mathematicians largely study the two subjects quite independently. The community of applied linear algebra, which studies computations with matrices and benefits from these studies, interacts very little with the computer algebra community which studies computations with polynomials and benefits from those studies. They constitute two distinct groups of people with distinct traditions, principles, and methods of working.

Matrix computation (applied linear algebra) people rely on numerical computation with rounding to a fixed precision. This enables faster computation using a small amount of computer memory. One must, however, take special care about rounding errors. The propagation of such errors in computational processes should be restricted to keep the output meaningful. This requirement has led to the advancement of error and perturbation analysis, approximation theory, and various techniques of algorithm design that stabilize computations numerically.

Computer algebra people are successfully exploiting and exploring an alternative path: error-free symbolic computations. This path requires more computer time and memory and thus is more expensive than numerical computation. The main advantage of symbolic computations is having completely reliable output. Various advanced mathematical tools have been developed to support this direction. Typical important examples are the transition to computations in finite fields, the Chinese remainder algorithm, and the p-adic Newton–Hensel lifting algorithms. Computations with polynomials make up much of computer algebra.

Polynomial and structured matrix computations combined

One of our goals is to reveal the correlation between computations with polynomials and structured matrices, continuing the line of the survey paper [P92a] and the book [BP94]. The expected impact includes better insight into both subjects and the unification of successful techniques and algorithms developed separately for each.

We study this correlation and its impact quite systematically in Chapters 2 and 3. This enables us to cover a substantial part of computer algebra using structured matrices. We observe close ties between structured matrices and the Nevanlinna–Pick and Nehari problems of rational interpolation and approximation. These celebrated algebraic problems allow numerical solution via reduction to matrix computations. Thus they may serve as a natural bridge between polynomial and matrix computations.

The displacement rank approach

Apart from unifying the study of matrices and polynomials, we focus on the design of effective algorithms unified over various classes of structured matrices. Our basic tool is the *displacement rank approach* to computations with structured matrices. The idea is to represent these matrices by their displacements, that is, the images of special displacement operators applied to the matrices. The displacements are defined by only small number of parameters, and the matrices can be recovered easily from their displacements.

The displacement rank approach consists of compression and decompression stages (the back and forth transition between matrices and their displacements) with dramatically simplified computations in-between (memory and time intensive computations with matrices are replaced with dramatically simplified operations with their displacements).

Some history and the displacement transformation approach to unification

The displacement rank approach to computations with structured matrices became celebrated after the publication of the seminal paper [KKM79], where some basic underlying results were presented for the class of Toeplitz-like matrices. Subsequent extensions to other classes of structured matrices followed [HR84], [GKK86], [GKKL87]. This was first done separately for each type of matrix structure, but in the paper [P90] we pointed out that the four most important classes of structured matrices, having structures of Toeplitz, Hankel, Vandermonde, and Cauchy types, are closely related to each other via their associated displacement operators.

For each of these four classes of matrices, we showed sample transformations into the three other classes by transforming the associated displacement operators and displacements. We proposed using such techniques systematically as a means of extending any efficient algorithm, developed for one of these four classes, to the other three classes. Later the approach proved to be effective for practical computations, such as the direct and iterative solution of Toeplitz/Hankel-like linear systems of equations (via the transformation of the displacement operators associated with their coefficients matrices to the ones of Cauchy type) [H95], [GKO95], [H96], [KO96], [HB97], [G97], [HB98], [Oa], numerical polynomial interpolation and multipoint evaluation [PLST93], [PZHY97], [PACLS98], [PACPS98], and algebraic decoding [OS99/OS00, Section 6]. In Chapters 4 and 5, we show applications of this approach for avoiding singularities in matrix computations and for accelerating numerical rational interpolation (Nevanlinna–Pick and Nehari problems). The list of applications of our approach is far from exhausted. The value of the method was widely recognized but its best known historical account in [G98] omits the source paper [P90]. Recent rediscovery of the method and techniques in [OS99/00], Section 6, caused even more confusion.

Symbolic operator approach to unification

An alternative and complementary method for unifying matrix structure is the *symbolic operator approach*. This approach also unifies numerical and algebraic algorithms. The matrix is associated with a largely unspecified displacement operator, and the high level description of the algorithms covers operations with displacements assuming their cited basic properties. This is the main framework for our presentation of some highly effective algorithms. The algorithms are completly specifiedwhen a structured input matrix, an associated displacement operator, and the rules of algebraic and numerical implementation are fixed. The specification is not our primary goal in this book, but we give some comments. We frequently combine the symbolic operator approach with displacement transformation techniques; we specify the results on the matrix structures. We also comment on the impact of our algorithms on the problems of computer algebra and numerical rational interpolation and approximation.

Unified superfast algorithms

How large is the gain of the displacement rank approach, based on shifting from matrices to their displacements?

An $n \times n$ structured matrix M has n^2 entries, all of which may be distinct, but its displacement may have small rank r, say equal to $1, 2$ or 3, even where n is very large. Then the matrix can be represented with many fewer from rn to $2rn$ parameters.

Similarly, the solution of a structured linear system of n equations, $M\mathbf{x} = \mathbf{b}$, can also be accelerated dramatically. *Classical* Gaussian elimination uses the order of n^3 operations but has *fast* versions for structured matrices, running in time of the order of n^2, where numerical stabilization with pivoting is also incorporated [H95], [GKO95], [G97]. In Chapters 5–7, we cover *unified superfast algorithms*, which support the running time bound of the order of $r^2 n \log^2 n$ and which we present in a unified way for various matrix structures. The latter time bound is *optimal* up to the (small) factor of $r \log^2 n$.

Two technically distinct classes of superfast algorithms are shown to complement each other. Divide-and-conquer algorithms are covered in Chapter 5, and Newton's iteration in Chapters 6 and 7. In both cases the algorithms can be applied numerically, with rounding to a fixed finite precision, and symbolically (algebraically), with infinite precision. We do not develop these implementations but give some comments and refer the reader to http://comet.lehman.cuny.edu/vpan/newton for a numerical version of Newton's iteration for structured matrices.

The presented unified superfast algorithms are immediately extended to many important computational problems having ties to structured matrices. We cover some applications to computer algebra and the Nevanlinna–Pick and Nehari problems of numerical rational computations. The correlation to structured matrices enables better insight into both areas and an improvement of numerical implementation of known solution algorithms. For the Nevanlinna–Pick and Nehari problems, their reduction to computations with structured matrices is the only known way to yield a superfast and therefore nearly optimal (rather than just fast) solution, which is unified for several variations of these problems.

Our presentation

Our primary intended readership consists of researchers and algorithm designers in the fields of computations with structured matrices, computer algebra, and numerical rational interpolation, as well as advanced graduate students who study these fields. On the other hand, the presentation in the book is elementary (except for several results that we reproduced with pointers to source papers or books), and we only assume superficial knowledge of some fundamentals from linear algebra. This should make the book accessible to a wide readership, including graduate students and new researchers who wish to enter the main disciplines of our study: computations with structured matrices and polynomials. Examples, tables, figures, exercises of various levels of difficulty, and sample pseudocodes in Section 2.15 should facilitate their efforts.

Actually, the author was quite attracted by a chance to present advanced hot topics and even new results in a structured form. Most of the material is covered with proofs, derivations, and technical details, but we completely avoid long proofs. Wherever we omit them, we supply the relevant bibliography in the Notes at the end of each chapter. In particular, this applies to Newton's numerical iteration (Chapter 6) and the reduction of Nevanlinna–Pick and Nehari problems to matrix computations (Sections 3.8 and 3.9). Other topics are much more self-contained. We hope that the inclusion of details and some well-known definitions and basic results will not turn off more advanced readers who may just skip the elementary introductory parts of the book. Similarly, many readers may focus on the description of the main algorithms and skip some details of their analysis and the complexity estimates.

In the Notes, we cite the most relevant related works, reflecting also the earlier non-unified versions of the presented algorithms. The reader should be able to trace the most imporant related works from the cited bibliography. We apologize for any omission, which is inavoidable because of the huge and rapidly growing number of publications on computations with structured matrices.

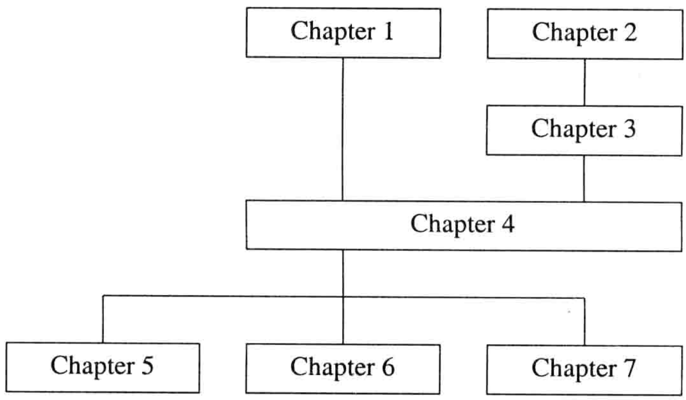
To keep our presentation unified, we omit the eigenproblem for structured matrices and do not study some important matrix classes (such as multilevel matrices, banded matrices, infinite but finitely generated structured matrices, and block banded block Toeplitz matrices with Toeplitz blocks), whose treatment relies on distinct techniques. We give only brief comments on the important issues of numerical stability and parallel implementation of the presented algorithms, sending the reader to the relevant bibliography, and we leave the topic of data structures to more introductory texts.

Chapters 2 and 3 and the first sections of Chapter 7 overlap with some expository material on polynomial and Toeplitz matrix computations in the book [BP94], but we extend this material substantially, present it more clearly and more systematically, and show the correlation between computations with polynomials and structured matrices more extensively. Otherwise, most of our presentation is from journals and proceedings articles. Furthermore, several new unpublished results are included. Chapters 4–7 unify and extend scanty preceding works mostly devoted to some specified classes of structured matrices. Chapter 5 has a very preliminary exposition in the Proceedings paper [OP98] and a more developed Proceedings version in [P00]. The Proceedings papers [PR01], [PRW00], and [P01] are the predecessors of the first five sections of Chapter 6; they have no substantial overlap with the last six sections of the chapter. Chapter 7 overlaps with [P92] and [P00b]. In the last three chapters and also in Sections 3.6, 3.7, and 4.4, we present several novel techniques and algorithms and yield new record estimates for the arithmetic and Boolean complexity of some fundamental computations with structured matrices as well as for the cited problems of rational interpolation and approximation.

Selective reading

Despite our goal of a unified study, we structure the book to encourage selective reading of chapters and even sections. This makes the book accessible to graduate student, yet less boring for experts with specialized interests. Guiding graphs, titles, and figures help the selection. In particular, we give the following guiding graph for selective reading of chapters.

Guiding Graph for Selective Reading of Chapters



Let us also give some guiding comments (also see Section 1.10). In Chapters 2, 3, and 7, we assume an interest on the part of the readers in computer algebra problems. Passing through Chapters 2 and 3, these readers can be motivated to study the superfast algorithms of Chapter 5 but may decide to skip Chapter 6. On the contrary, readers who come from applied linear algebra and have no particular interest in computer algebra may focus on Chapters 4–6 and skip Chapters 2 and 3, except for some basic facts, the estimates collected in Table 1.2, and the definitions reproduced in Section 4.1. Furthermore, either Chapters 5 or 6 can be read independently of one another. In the beginning of Chapters 3–7 we display graphs for selective reading within each of these chapters. We propose that all sections of Chapters 1 and 2 be read in succession.

Chapter 1 outlines our main subjects and should serve as an introduction to the book. Technically, however, it can be skipped, except for some basic results, examples, and definitions in Sections 1.3–1.5, for which we give pointers whenever we use them.

Those readers whose interests are restricted to the tangential Nevanlinna–Pick and matrix Nehari problems of rational interpolation and approximation or to the loss-resilient encoding/decoding may start their reading with the respective sections of Chapter 3 and then if necessary may follow pointers and cross references to other parts of the book.

Summary of the book

Let us briefly summarize what we cover:

1. Unification of studies in the areas of:
    (a) computations with structured matrices and polynomials,
    (b) computer algebra and numerical linear algebra,
    (c) matrix structures of various classes,
    (d) structured matrices and Nevanlinna–Pick and Nehari problems.
2. Fundamental techniques:
    (a) displacement rank approach,
    (b) algorithmic transformation techniques,
    (c) divide-and-conquer method and recursive matrix factorization,
    (d) Newton's iteration in numerical and algebraic versions.
3. Superfast and memory efficient algorithms for several fundamental problems, including new algorithms and new derivations and analysis of some known algorithms for
    (a) structured matrix computations,
    (b) numerical rational interpolation and approximation,
    (c) loss-resilient encoding/decoding,
    (d) other areas of computer algebra.

Several unpublished results are included, in particular in Chapters 5–7 and in Sections 3.6, 4.4, and 4.6.2. (See more details in the Notes following each chapter).

The unification approach reflects the personal taste of the author whose primary interests include numerical linear algebra, polynomial computations, and algorithm analysis. He tried to focus on the key ideas and techniques of these areas. He believes, however, that the unification approach is beneficial for the subjects covered, since it provides for a deeper understanding of the power and the deficiencies of the solution algorithms.

This book should invite researchers (and perhaps some novices) to explore some subjects further. In particular, the following areas seem to be widely open to new research and/or implementation of recent algorithms:

- applications of the displacement transformation approach, including randomized transformations,
- analysis and implementation of the Newton-Structured Iteration and its generalizations and variations, including scaled versions, extensions to other residual correction methods, and particularly heuristic variations,

- complexity analysis and numerical analysis of the algorithms of Chapters 4 and 5 and their implementation and applications to specific classes of structured matrices,
- elaboration and implementation of the presented superfast algorithms for the solution of specific problems of encoding/decoding, computer algebra, and rational interpolation and approximation,
- parallel implementation of the presented algorithms (see [BP94, Chapter 4] and [P96], [P00b]).

Taking into account scientific and applied interest in topics covered, this book should fill a substantial void in the market. On the other hand, we hope that the elementary presentation will attract new people to the subjects and will help unify the efforts of researchers in several covered areas.

Acknowledgements

My involvement in the subjects of this book was motivated by my joint research with M. Abutabanjeh, A. Atinkpahoun, D. A. Bini, D. Bondyfalat, S. Branham, H. Brönnimann, H. Cebecioğlu, Z. Chen, J. W. Demmel, O. Dias (Tiga), I. Z. Emiris, Z. Galil, L. Gemignani, Y. Han, X. Huang, E. Kaltofen, M.-H. Kim, M. Kunin, J. Laderman, E. I. Landowne, K. Li, Y. Lin-Kriz, E. Linzer, W. L. Miranker, B. Mourrain, B. Murphy, V. Olshevsky, S. Pion, F. P. Preparata, S. Providence, Y. Rami, J. H. Reif, R. E. Rosholt, O. Ruatta, A. Sadikou, R. Schreiber, H. X. Sha, D. Shallcross, I. Sobze, C. Stewart, S. R. Tate, X. Wang, Y. Yu, and A. Zheng. I am grateful to all of them and to many other senior and junior colleagues, for their stimulating interest in my work, (p)reprints, support, discussions, and/or helpful comments on the subjects of this book, in particular to A. B. Borodin, B. Codenotti, S. A. Cook, R. M. Corless, G. Cybenko, D. Dobkin, W. Eberly, C. M. Fiduccia, P. Flajolet, A. Galligo, J. von zur Gathen, G. Heinig, T. Kailath, I. Kaporin, D. E. Knuth, Y. N. Lakshman, L. A. Levin, R. J. Lipton, L. Lovász, R. C. Melville, A. M. Odlyzko, J. Renegar, M.-F. Roy, B. Salvy, I. Shparlinski, M. Shub, S. Smale, H. J. Stetter, J. F. Traub, E. E. Tyrtyshnikov, M. Van Barel, P. Van Dooren, G. Villard, S. M. Watt, and S. Winograd. I specially thank Dario A. Bini for long time collaboration, Ioannis Z. Emiris and Bernard Mourrain for productive years of joint research on algebraic computation, Vadim Olshevsky for his enthusiasm about the papers [PZ00] and [PZHY97], suggestion of their speedy submission, his most valuable part in our joint effort on their extension, and advising me on the Nevanlinna–Pick and Nehari problems, both Georg Heinig and Vadim Olshevsky for their early appreciation of the displacement transformation method and successful demonstration of its power, and Erich Kaltofen and Joachim von zur Gathen for many informative discussions on algebraic complexity.

Very encouraging for me were the invitations to present the material of Chapters 4-6 at the Joint 1999 AMS/IMS/SIAM Summer Research Conference "Structured matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing," organized by V. Olshevsky, in Boulder, Colorado, June 27–July 1, 1999; at the structured matrix session of G. Heinig and S. Serra Capizzano of the Second Conference on Numerical Analysis and Applications, organized by P. Yalamov in Rousse, Bulgaria, in June 2000; of Chapters 4, 5, and 7 at the Annual IMACS/ACA Conference, organized by N. Vassiliev in St. Petersburg, Russia, in June 2000; and of Chapters 4 and 6 at the P. Dewilde's session at the 14th International Symposium on Mathematical Theory of Networks and Systems in Perpignan, France, in June 2000; acceptance of my papers [P00] on Chapters 4 and 5 and [P00c] on Section 3.7 by the Program Committees of the ACM-SIAM SODA 2000 and the ACM-SIGSAM ISSAC 2000, respectively; the reception of my talks by the audiences at all these meetings; and the invitations to speak at the minisymposium on fast matrix algorithms (organized by S. Chandrasekaran and M. Gu) at the SIAM Annual Meeting in San Diego, California, in July 2001; at the Joint 2001 AMS/IMS/SIAM Summer Research Conference on Fast Algorithms, organized by V. Olshevsky in S. Hadley, Massachusetts, in August 2001; and at the special session on structured matrices organized by D. Bini and T. Kailath at the Joint AMS-UMI Conference in Pisa, Italy, in June 2002. T. Kailath's invitation to participate in the volume [KS99] revived my interest in Newton's iteration.

Besides this ample encouragement, writing of this book was supported by NSF Grant CCR 9732206, PSC CUNY Awards 61393-0030 and 62435-0031, and George N. Shuster Fellowship. Xinmao Wang and Hülya Cebecioğlu provided superb assistance with preparation of the book. I am grateful to Ioannis Z. Emiris and Robert C. Melville for prompt but careful review of the draft of the book and most valuable comments; to Anh Nguyen, Samer Salame, and particularly Elliott I. Landowne, Hülya Cebecioğlu, and Xinmao Wang for pointing out several typos in the manuscript; to Brian Murphy and Rhys Rosholt for many helpful comments on the style of the presentation in the original drafts of the Preface and Chapter 1; to Mikhail Kunin and Rhys Rosholt for carrying out the experiments on Newton's iteration, and to Brian Murphy, Rhys Rosholt, Samer Salame, Alexandre Ushakov, and Xinmao Wang for their assistance with writing pseudocodes. I greatly appreciate full cooperation of Birkhäuser/Springer, in particular most helpful collaboration with Ann Kostant, the timely liaison service of Tom Grasso and Nick Burns, the beautiful design of the book cover by Mary Burgess, and the excellent job of the copyeditor. It was both most vital and pleasant for me to have ample unconditional support of my research at Lehman College and the Graduate Center of CUNY as well as my home.

Glossary of mathematical notation

Notation	Explanation
$\mathbb{F}$	an arbitary field
$\mathbb{C}$	the field of complex numbers
$\mathbb{R}$	the field of real numbers
$M \in \mathbb{F}^{m \times n}$	an $m \times n$ matrix with entries from field $\mathbb{F}$
$M = (m_{i,j})_{i,j=0}^{m-1,n-1}$	$m \times n$ matrix with the (i, j)-th entry $m_{i,j}$
$\mathbf{v} \in \mathbb{F}^{k \times 1}$	a k-dimensional column vector with coordinates from filed $\mathbb{F}$
$\mathbf{v} = (v_i)_{i=0}^{k-1}$	k-dimensional vector with the i-th coordinate v_i, $i = 0, \ldots, k-1$
$\mathbf{S} = \{s_1, \ldots, s_k\}$	the set of k elements $s_1, \ldots, s_k$ (not necessarily distinct)
$(W_1, \ldots, W_k)$	$1 \times k$ block matrix with blocks $W_1, \ldots, W_k$
$(\mathbf{w}_1, \ldots, \mathbf{w}_k)$	$n \times k$ matrix with the columns $\mathbf{w}_1, \ldots, \mathbf{w}_k$, where n is the dimension of the column vectors
$W^T, \mathbf{v}^T$	the transpose of a matrix W, the transpose of a vector $\mathbf{v}$
$W^*, \mathbf{v}^*$	the Hermitian (conjugate) transpose of a matrix, the Hermitian (conjugate) transpose of a vector
$\mathbf{e}_i$	the i-th coordinate vector
$I = I_n$	the $n \times n$ identity matrix
$0, O$	null matrices
$D(\mathbf{v}), \mathrm{diag}(\mathbf{v})$	the diagonal matrix with a vector $\mathbf{v}$ defining its diagonal
$T = (t_{i-j})_{i,j=0}^{n-1}$	an $n \times n$ Toeplitz matrix

Z, Z_0	an $n \times n$ matrix with the (i, j)-th entry 1 for $i = j+1$ and 0 for all other pairs i, j, the unit lower triangular Toeplitz matrix
Y_{00}	$Z + Z^T$
Y_{11}	$Z + Z^T + \mathbf{e}_0\mathbf{e}_0^T + \mathbf{e}_{n-1}\mathbf{e}_{n-1}^T$
Z_f	$Z + f\mathbf{e}_0\mathbf{e}_{n-1}^T$, the unit f-circulant matrix
$Z(\mathbf{v}), Z_0(\mathbf{v})$	$\sum_{i=0}^{n-1} v_i Z^i$, the lower triangular Toeplitz matrix with the first column $\mathbf{v}$
$Z_f(\mathbf{v})$	$\sum_{i=0}^{n-1} v_i Z_f^i$, the f-circulant matrix with the first column $\mathbf{v}$
$H = (h_{i+j})_{i,j=0}^{n-1}$	an $n \times n$ Hankel matrix
J	$(\mathbf{e}_{n-1}, \dots, \mathbf{e}_0)$, the $n \times n$ unit Hankel matrix, the reflection matrix
$V(\mathbf{t}) = (t_i^j)_{i,j=0}^{n-1}$	the $n \times n$ Vandermonde matrix with the second column $\mathbf{t}$
$C(\mathbf{s}, \mathbf{t}) = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$	the $n \times n$ Cauchy matrix defined by two vectors $\mathbf{s}$ and $\mathbf{t}$
$K(M, \mathbf{v}, k)$	$(M^i\mathbf{v})_{i=0}^{k-1}$, the $n \times k$ Krylov matrix with columns $M^i\mathbf{v}$, $i = 0, \dots, k-1$
L	linear operator
A, B	operator matrices
G, H	generator matrices, $GH^T = L(M)$
$\nabla_{A,B}$	a Sylvester type displacement operator
$\nabla_{A,B}(M)$	$AM - MB$, Sylvester type displacement of M
$\Delta_{A,B}$	a Stein type displacement operator
$\Delta_{A,B}(M)$	$M - AMB$, Stein type displacement of M
ω, ω_n	a primitive n-th root of 1
$\mathbf{w}$	the vector $(\omega^i)_{i=0}^{n-1}$ of the n-th roots of 1
Ω	$(\omega^{ij})_{i,j=0}^{n-1}$, the scaled matrix of the DFT
$\mathrm{DFT}(\mathbf{p})$	$(\sum_{i=0}^{n-1} p_i \omega_n^{ik})_{k=0}^{n-1}$, the discrete Fourier transform of a vector $\mathbf{p}$
DFT_n	computation of $\mathrm{DFT}(\mathbf{p})$
$p(x) \bmod u(x)$	$p(x)$ modulo $u(x)$, the remainder of the division of univariate polynomials $p(x)$ by $u(x)$