INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

A

A Unified Approach to Structured Matrix Inversion and an Extension to Fast Solution of Trummer's Problem

by

Stephen V. Providence

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2000

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

UMI Number: 9959217

UMI®

UMI Microform 9959217

Copyright 2000 by Bell & Howell Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

January 31, 2000 ____________________ Prof. Victor Pan ____________________

Date **Chair of Examining Committee**

JANUARY 28, 2000 ____________________ Prof. Stathis Zachos ____________________

Date **Executive Officer**

STANLEY HABIB

VICTOR PAN

STATHIS ZACHOS

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Abstract

A Unified Approach to Structured Matrix Inversion and an Extension to Fast Solution to Trummer's Problem

by

Stephen V. Providence

Advisor: Professor Victor Y. Pan

We extend to any input, the Greengard and Rokhlin multipole algorithm for solution to Trummer's problem or the problem of multiplying a Cauchy matrix by a vector. We use matrix transformation equations (cf. [PACLS,a]) to transform Trummer's problem into one or more special Trummer's problems, where we choose one of the input vectors. Our choice of input vector bypasses the weakness in the multipole algorithm of having some difficult inputs and along with transformation via matrix equations, allows us to exploit the speed of the FFT and inverse the FFT. We present a superfast divide-and-conquer algorithm to invert strongly nonsingular, nonsingular, and singular structured matrices of the basic classes. Operators of displacement and scaling which produce generator matrices, together with discrete cosine or disrete sine transforms (cf. [GKO95]), transform structured matrices into Cauchy-like matrices. When applicable, symmetrization is used to ensure strong nonsingularity. Randomization may transform a given input structured matrix into one of generic rank profile (GRP). A strongly nonsingular Cauchy-like matrix or a Cauchy-like matrix with GRP may be inputs to our superfast divide-and-conquer algorithm. We use short matrix generator blocks as input to our algorithm.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Contents

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

List of Tables

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 1

Motivation and Preliminaries

1.1 Introduction

Structured matrices are omnipresent in computation for sciences, engineering and communication. Particularly important are Toeplitz, Hankel, Vandermonde and Cauchy matrices and matrices having related structure. We first study the basic operation of multiplication of a Cauchy matrix by a vector, called Trummer's problem, having important applications to solutions to integral equations, particle simulations, and conformal mapping, among others. We propose a matrix transformation that enables us to extend the application of the celebrated Fast Multipole algorithm (proposed by Greengard and Rokhlin) to the general input class of Cauchy matrices. Moreover, our techniques enables an extension to this effective algorithm to yield a fast and numerically stable solution of the classical problems of multipoint polynomial evaluation and multipoint polynomial interpolation.

Our second topic is recursive factorization of structured matrices which enables us to

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

compute the rank and the determinant of the input matrix, as well as its inverse (if the matrix is nonsingular) and a basis for its null space. This approach was previously developed for Toeplitz and Toeplitz-like matrices yielding nearly optimal (nearly linear time and space) computations. We first extend such a divide and conquer approach to the case of Cauchy-like input. Then we propose a unified approach covering all the four fundamental classes of structured matrices. We propose some special techniques to handle singularity by means of randomization, and we follow the ideas of [P90], [P2000] to improve the resulting algorithms further by using transformations among various classes of structured matrices.

The $n \times n$ dense structured matrices are defined by $O(n)$ parameters, and their structure enables dramatic acceleration of computations with such matrices. For example, an $n \times n$ Toeplitz matrix $T = [t_{i-j}]$ can be multiplied by a vector over any field of constants by using

$$T_{M_v}(n) = O((n \log n) \log\log n) \tag{1.1}$$

arithmetic operations [BP94] versus $2n^2 - n$ for general $n \times n$ matrices (hereafter, we refer to arithmetic operations as *ops*). Furthermore, the well-known divide-and-conquer algorithm of [M74], [M80], [BA80] (hereafter, we refer to it as the *MBA algorithm*) rapidly computes the recursive triangular factorization of T, as well as T^{-1}, $\det T$, and the solution $\vec{x} = T^{-1}\vec{b}$ to a linear system $T\vec{x} = \vec{b}$. The algorithm uses

$$T_{RF}(n) = O(T_{M_v}(n) \log n) = O((n \log^2 n) \log\log n)$$

ops over a wider class of Toeplitz-like matrices, having structure of Toeplitz type defined by associated *displacement operators* [KKM]. The algorithm is called *superfast* because it runs in almost linear time versus both cubic time of Gaussian elimination and quadratic

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

time of some other known fast algorithms such as Levinson's and Trench's [GL]. In [BP93], [PZ,a], [PACPS98], [OP98], the MBA algorithm was extended to other structured matrices, in particular, to Toeplitz-like + Hankel-like matrices [BP93] and to Cauchy-like and Vandermonde-like matrices [PZ,a], [PACPS98], [OP98], [P2000]. The reader is referred to [BP94], [KS99], [PACLS,a], [OP98], [OS99] and to the bibliography therein on various applications of structured matrices. In many cases, in particular in application to signal processing, Padé approximation and sparse multivariate polynomial interpolation, one needs to solve singular Toeplitz-like, Vandermonde-like or Cauchy-like linear systems of equations.

Kaltofen in [K95] extended the MBA approach to the solution of a singular Toeplitz-like linear system of equations. We will do the same thing in the Cauchy-like case and will also ameliorate slightly the algorithm of [K95] in the Toeplitz-like case. Our algorithms use

$$T_{sing}(n) = O(T_{M_v}(n) \log n) \tag{1.2}$$

ops in the Toeplitz-like case (as in [K95]) and

$$C_{sing}(n) = O(C_{M_v}(n) \log n) \tag{1.3}$$

ops in the Cauchy-like case. Here,

$$C_{M_v}(n) = O((n \log^2 n) \log \log n) \tag{1.4}$$

is the number of ops sufficient to solve *Trummer's problem* of multiplication of an $n \times n$ Cauchy matrix by a vector over any field of constants $\mathbf{F}$ (cf. [BP94], p. 130; [Ger87],

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[PACPS98]). If $\mathbf{F}$ supports FFT, then the factor $\log\log n$ in (1.1) and (1.4) can be removed. Our algorithms sample $4n$ random parameters from a fixed finite set S of cardinality $|S|$ in the Cauchy-like case. In Toeplitz-like case, they sample $2n-2$ parameters (versus order of $n\log n$ in [K95]). Our algorithms may fail with a probability at most $2\rho(\rho+1)/|S|$ where ρ is the rank of the input matrix (versus $4n\rho/|S|$ in [K95]), otherwise they produce correct output. Besides being linear solvers, our algorithms (like one of [K95]) can be immediately applied to compute (at the same asymptotic computational cost) the rank of a given matrix and a basis for its null-space and to verify the correctness of the output in all cases.

Trummer's problem, besides being the main basic block of our algorithms for Cauchy-like computations, is highly important in its own right; its solution having several scientific and engineering applications. This motivated our work on its numerical solution. Currently the best is the Multipole Algorithm by Greengard and Rokhlin. It has some difficulties for certain input classes. Our contribution is a novel method of the extension of the algorithm to any input (see chapter 2).

Our study of Toeplitz-like and Cauchy-like computations and, in particular, our asymptotic complexity estimates (1.1), (1.2) and (1.3), (1.4) can be easily extended to the study of Hankel-like and Vandermonde-like computations, respectively. The bound (1.3) is supported directly by our algorithms in the Cauchy-like and Vandermonde-like cases, but in both cases it can be improved to the bound

$$T_{sing}(n) + O(V_{M_v}(n)) = O((n\log^2 n)\log\log n) \tag{1.5}$$

(assuming that an $n \times n$ Vandermonde matrix can be multiplied by a vector in $V_{M_v}(n)$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

ops), by means of some general techniques proposed in [P90] for the transformation (at the cost $O(V_{M_v}(n)))$ of various computational problems for Cauchy-like and Vandermonde-like matrices to the same problems for Toeplitz/Hankel-like matrices. (In fact, [P90] defined such transformations among all the cited classes of structured matrices in all directions.) The difference by a logarithmic factor may be not decisive in practice, because of the role of other potential criteria (such as numerical stability and the decrease of the overhead constants hidden in the above "O" notation), so we show all estimates (1.1)-(1.5) and not just (1.1), (1.2), (1.5). Furthermore, we comment on further practical improvement in Remark 3.2.1.

On various applications of Cauchy-like algorithms, see bibliography in [OP98], [PACLS,a] and note that by using the cited transition to Toeplitz/Hankel computations we may save a logarithmic factor in the complexity estimates versus ones of [OP98]. More dramatic improvement of the known record complexity estimates was resulted from the application of our algorithms to the list decoding of Reed-Solomon and algebraic-geometric codes [OS99], [SW98], [Su97]. The bottleneck of the list decoding is the computation of a nonzero vector from the null space of a singular $l \times m$ Vandermonde-like matrix where $l + m = O(n)$, n being the input size of the decoding problem. Application of our algorithms enables immediate decrease of the record complexity of these computations by order of magnitude - from at best quadratic order of n^2 [OS99], [SW98], [Su97] to almost linear order of n (up to a polylogarithmic factor).

We organize this thesis as follows. In chapter 1, we recall some definitions and auxiliary facts. In chapter 2, we recall Trummer's problem and its solution by the multipole algorithm,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and then extend the algorithm to a more general class of inputs. In chapter 3, we recall the MBA algorithm and the results of [PZ,a], [PACPS98], [OP98] on its Cauchy-like extension. In chapter 4, section 1 we apply randomization to extend the latter results to the case of a nonsingular but not strongly nonsingular input matrix. In chapter 4, sections 2 and 3, we cover the extension of Cauchy-like and Toeplitz-like solvers to the singular case. In chapter 5, we extend our algorithms to the Hankel-like and Vandermonde-like cases (cf. [P2000]), effectively covering the four basic classes of structured matrix.

1.2 Structured Matrices

1.2.1 Basic Classes

Definition 1.2.1 *We define Toeplitz, Hankel, Vandermonde, Chebyshev-Vandermonde and Cauchy matrices* $T = (t_{i,j})$, $H = (h_{i,j})$, $V(\vec{x}) = (v_{i,j})$, $V_P(\vec{x})$ $C(\vec{s}, \vec{t}) = (c_{i,j})$, *respectively, where* $t_{i+1,j+1} = t_{i,j}$, $h_{i+1,j-1} = h_{i,j}$, $v_{i,j} = x_i^j$, $c_{i,j} = \frac{1}{s_i - t_j}$, $\vec{s} = (s_i)$, $\vec{t} = (t_i)$, $\vec{x} = (x_i)$, $s_i \neq t_j$, *for all i and j for which the above values are defined.*

$$T = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{-n+1} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}, \quad H = \begin{bmatrix} h_0 & \cdots & h_{n-2} & h_{n-1} \\ \vdots & \iddots & h_{n-1} & h_n \\ h_{n-2} & \iddots & \iddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 1.1: Definition of the Basic Classes of Structured Matrices

Toeplitz-like [KKM]	$F = Z_1$	$A = Z_{-1}$
Toeplitz-plus-Hankel-like [HJR], [GK], [SLAK]	$F = Y_{00}$	$A = Y_{11}$
Cauchy-like [HR]	$F = \mathrm{diag}(c_1, \ldots, c_n)$	$A = \mathrm{diag}(d_1, \ldots, d_n)$
Vandermonde-like [HR]	$F = \mathrm{diag}(1/x_1, \ldots, 1/x_n)$	$A = Z_1$
Chebyshev-Vandermonde-like [KO]	$F = \mathrm{diag}(1/x_1, \ldots, 1/x_n)$	$A = 2 \cdot \sum_{i=1}^{\lceil n/2 \rceil} (-1)^{i-1} (Z_0^T)^{2i-1}$

$$V(\bar{x}) = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^{n-1} \\ 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^{n-1} \end{bmatrix}, \quad C(\bar{s}, \bar{t}) = \begin{bmatrix} \frac{1}{s_0 - t_0} & \frac{1}{s_0 - t_1} & \cdots & \frac{1}{s_0 - t_{n-1}} \\ \frac{1}{s_1 - t_0} & \frac{1}{s_1 - t_1} & \cdots & \frac{1}{s_1 - t_{n-1}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{s_{n-1} - t_0} & \frac{1}{s_{n-1} - t_1} & \cdots & \frac{1}{s_{n-1} - t_{n-1}} \end{bmatrix}$$

$$V_P(\bar{x}) = \begin{bmatrix} P_0(x_0) & P_1(x_0) & \cdots & P_{n-1}(x_0) \\ P_0(x_1) & P_1(x_1) & \cdots & P_{n-1}(x_1) \\ \vdots & \vdots & & \vdots \\ P_0(x_{n-1}) & P_1(x_{n-1}) & \cdots & P_{n-1}(x_{n-1}) \end{bmatrix}$$

$O(n)$ parameters fully represent an $n \times n$ structured matrix.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.2.2 Matrix Multipliers

We write

$$Z_f = \begin{bmatrix} 0 & 0 & \cdots & 0 & f \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, Y_{\gamma,\delta} = \begin{bmatrix} \gamma & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & \delta \end{bmatrix}$$

In, particular Z_0 is the lower shift matrix.

$$Z_f(\vec{v}) = \sum_{i=0}^{n-1} v_i Z_f^i = \begin{bmatrix} v_0 & fv_{n-1} & fv_{n-2} & \cdots & \cdots & fv_2 & fv_1 \\ v_1 & v_0 & fv_{n-1} & fv_{n-2} & & & fv_2 \\ v_2 & v_1 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & v_2 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & fv_{n-2} \\ v_{n-2} & & & \ddots & \ddots & \ddots & fv_{n-1} \\ v_{n-1} & v_{n-2} & \cdots & \cdots & v_2 & v_1 & v_0 \end{bmatrix}$$

$$Z_f \cdot \vec{v} = \begin{bmatrix} 0 & 0 & \cdots & 0 & f \\ 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} v_0 \\ v_1 \\ \vdots \\ \vdots \\ v_{n-1} \end{bmatrix} = \begin{bmatrix} fv_{n-1} \\ v_0 \\ v_1 \\ \vdots \\ v_{n-2} \end{bmatrix}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Also, we define the diagonal matrix $D_{\bar{x}} = \text{diag}(\bar{x})$ for a vector $\bar{x} = (x_0, x_1, \ldots, x_{n-1})$:

$$D_{\bar{x}} = \begin{bmatrix} x_0 & 0 & \cdots & \cdots & 0 \\ 0 & x_1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & x_{n-1} \end{bmatrix}$$

1.2.3 Displacement Structure, Toeplitz and Toeplitz-like Matrices

We will next recall the fundamental concept of dislacement structure of matrices and some basic facts for their study. (cf. [KKM], [CKL-A87], [GO94], [BP94], [GKO95], [P2000]) Let matrices F, $A \in \mathbf{C}^{n\times n}$ be given. Let $R \in \mathbf{C}^{n\times n}$ be a matrix satisfying the Sylvester type equation

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A = G \cdot B, \tag{2.1}$$

with given rectangular matrices $G \in \mathbf{C}^{n\times\alpha}$, $B \in \mathbf{C}^{\alpha\times n}$, where number α is small in comparison with n. The pair of matrices in (2.1) is referred to as a $\{F, A\}$-*generator* of R of length α and the smallest possible length α among all $\{F, A\}$-generators of R is called an $\{F, A\}$-*displacement rank* of R. The concept of *displacement structure* was introduced in [KKM] using the linear operator $\nabla_{\{Z_0, Z_0^T\}}(\cdot) : \mathbf{C}^{n\times n} \to \mathbf{C}^{n\times n}$, which transforms each matrix R to its *displacement(in Stein format)*,

$$\nabla_{\{Z_0, Z_0^T\}}(R) = R - Z_0 \cdot R \cdot Z_0^T \tag{2.2}$$

where the superscript $(\cdot)^T$ denotes transposition of a matrix. The number

$$\alpha = \text{rank}\nabla_{\{Z_0, Z_0^T\}}(R) = \text{rank}(R - Z_0 \cdot R \cdot Z_0^T) \tag{2.3}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

is referred to as the $\{Z_0, Z_0^T\}$-displacement rank of R. Definitions based on (2.1) and (2.2) are closely related to each other. In particular we have

Theorem 1.2.1 *Under the definition of the displacement rank based on (2.1) with $F = Z_1$, $A = Z_{-1}$ as well as under the definition based on (2.3), the displacement rank of a Toeplitz matrix for Toeplitz matrix R does not exceed 2.*

Proof: Given a Toeplitz matrix $T = [t_{i-j}]_{1 \le i,j \le n}$,

$$T = \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & \cdots & t_{-n-1} \\ t_1 & t_0 & t_{-1} & \ddots & & \vdots \\ t_2 & t_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & t_{-2} \\ \vdots & & \ddots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & \cdots & t_2 & t_1 & t_0 \end{pmatrix}$$

and lower shift circulant matrices Z_1, Z_{-1}, defined in section (1.2.2),

$$Z_1 = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \ddots & & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix}, Z_{-1} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & 0 & \ddots & & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix},$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the Sylvester matrix equation $\nabla_{\{Z_1,Z_{-1}\}}(T) = Z_1 \cdot T - T \cdot Z_{-1} =$

$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & \ddots & & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & \cdots & t_{-n+1} \\ t_1 & t_0 & t_{-1} & \ddots & & \vdots \\ t_2 & t_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & t_{-2} \\ \vdots & & \ddots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & \cdots & t_2 & t_1 & t_0 \end{pmatrix}$$

$$- \begin{pmatrix} t_0 & t_{-1} & t_{-2} & \cdots & \cdots & t_{-n+1} \\ t_1 & t_0 & t_{-1} & \ddots & & \vdots \\ t_2 & t_1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & t_{-2} \\ \vdots & & \ddots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & \cdots & t_2 & t_1 & t_0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & 0 & \ddots & & 0 \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 & 0 \end{pmatrix}$$

reduces to $\nabla_{\{Z_1,Z_{-1}\}}(T) =$

$$\begin{pmatrix} t_{n-1} & t_{n-2} & \cdots & \cdots & t_1 & t_0 \\ t_0 & t_{-1} & \cdots & \cdots & t_{-n+2} & t_{-n+1} \\ \vdots & \vdots & & & \vdots & \vdots \\ \vdots & \vdots & & & \vdots & \vdots \\ t_{n-3} & t_{n-4} & \cdots & \cdots & t_{-1} & t_{-2} \\ t_{n-2} & t_{n-3} & \cdots & \cdots & t_0 & t_{-1} \end{pmatrix} - \begin{pmatrix} t_{-1} & t_{-2} & \cdots & \cdots & t_{-n+1} & -t_0 \\ t_0 & t_{-1} & \cdots & \cdots & t_{-n+2} & -t_1 \\ \vdots & \vdots & & & \vdots & \vdots \\ \vdots & \vdots & & & \vdots & \vdots \\ t_{n-3} & t_{n-4} & \cdots & \cdots & t_{-1} & -t_{n-2} \\ t_{n-2} & t_{n-3} & \cdots & \cdots & t_0 & -t_{n-1} \end{pmatrix}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$
= \begin{pmatrix}
(t_{n-1} - t_{-1}) & (t_{n-2} - t_{-2}) & \cdots & \cdots & (t_1 - t_{-n+1}) & 2t_0 \\
0 & 0 & \cdots & \cdots & 0 & (t_{-n+1} + t_1) \\
\vdots & \vdots & & & \vdots & \vdots \\
\vdots & \vdots & & & \vdots & \vdots \\
0 & 0 & \cdots & \cdots & 0 & (t_{-2} + t_{n-2}) \\
0 & 0 & \cdots & \cdots & 0 & (t_{-1} + t_{n-1})
\end{pmatrix}
$$

$$
= \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} (t_{n-1} - t_{-1}) \\ (t_{n-2} - t_{-2}) \\ \vdots \\ \vdots \\ (t_1 - t_{-n+1}) \\ t_0 \end{pmatrix}^T + \begin{pmatrix} t_0 \\ (t_{-n+1} + t_1) \\ \vdots \\ \vdots \\ (t_{-2} + t_{n-2}) \\ (t_{-1} + t_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{pmatrix}^T
$$

with the $\{Z_1, Z_{-1}\}$-displacement generator.

For the Stein type matrix equation, $\nabla_{\{Z_0, Z_0^T\}}(T) = T - Z_0 \cdot T \cdot Z_0^T$, we obtain a similar expansion

$$
\nabla_{\{Z_0, Z_0^T\}}(T) = \begin{pmatrix}
t_0 & t_{-1} & \cdots & \cdots & t_{-n+2} & t_{-n+1} \\
t_1 & 0 & \cdots & \cdots & 0 & 0 \\
\vdots & \vdots & & & \vdots & \vdots \\
\vdots & \vdots & & & \vdots & \vdots \\
t_{n-2} & 0 & \cdots & \cdots & 0 & 0 \\
t_{n-1} & 0 & \cdots & \cdots & 0 & 0
\end{pmatrix}
$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$= \begin{pmatrix} \frac{t_0}{2} \\ t_1 \\ \vdots \\ \vdots \\ t_{n-2} \\ t_{n-1} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix}^T + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{t_0}{2} \\ t_{-1} \\ \vdots \\ \vdots \\ t_{-n+2} \\ t_{-n+1} \end{pmatrix}^T$$

with the $\{Z_0, Z_0^T\}$-displacement Stein type generator. This proves that under both definitions, the displacement rank of any Toeplitz matrix does not exceed 2. □

In fact, Theorem 1.2.1 holds true even for any choice of both Sylvester type operator $\nabla_{\{Z_e, Z_f\}}$, $e \neq f$ and Stein type operator $\nabla_{\{Z_e, Z_f^{-1}\}}$, $f \neq e$, $f \neq 0$.

$$\nabla = \sum_{m=1}^{\alpha} a_m \cdot b_m^T \quad (a_m, b_m \in \mathbf{C}^n), \tag{2.4}$$

since any matrix ∇ of rank at most α can be nonuniquely represented as a sum of α products of vectors, (with small α). We will refer to any matrix having displacement rank α as a *Toeplitz-like* matrix (the designations of *Toeplitz type* matrix and *close to Toeplitz* matrix are also in use.) For the operator $\nabla_{\{Z_0, Z_0^T\}}(\cdot)$ of (2.2), it is shown in [KKM] that any matrix $R \in \mathbf{C}^{n \times n}$ is uniquely determined by its displacement, and that equality (2.4) holds if and only if

$$R = \sum_{m=1}^{\alpha} L(a_m) \cdot L(b_m)^T, \tag{2.5}$$

where $L(a)$ denotes a lower triangular Toeplitz matrix whose first column is a. The two basic properties of displacement rank are that it is preserved by the operations of inversion and Schur complementation.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.2.4 Chebyshev-Vandermonde and Chebyshev-Vandermonde-like matrices

Next we will study Chebyshev-Vandermonde and Chebyshev-Vandermonde-like matrices by following [KO]. Chebyshev polynomials of the first kind $T_0(\vec{x}), T_1(\vec{x}), \ldots, T_{n-1}(\vec{x})$ and of the second kind $U_0(\vec{x}), U_1(\vec{x}), \ldots, U_{n-1}(\vec{x})$ are the basis for the following Chebyshev-Vandermonde matrices

$$V_T(\vec{x}) = \begin{pmatrix} T_0(x_0) & T_1(x_0) & \cdots & T_{n-1}(x_0) \\ T_0(x_1) & T_1(x_1) & \cdots & T_{n-1}(x_1) \\ \vdots & \vdots & & \vdots \\ T_0(x_{n-1}) & T_1(x_{n-1}) & \cdots & T_{n-1}(x_{n-1}) \end{pmatrix}; \tag{2.1}$$

$$V_U(\vec{x}) = \begin{pmatrix} U_0(x_0) & U_1(x_0) & \cdots & U_{n-1}(x_0) \\ U_0(x_1) & U_1(x_1) & \cdots & U_{n-1}(x_1) \\ \vdots & \vdots & & \vdots \\ U_0(x_{n-1}) & U_1(x_{n-1}) & \cdots & U_{n-1}(x_{n-1}) \end{pmatrix}.$$

It was shown in [GO2] and [FHR93] that if $x_0, x_1, \ldots, x_{n-1}$ are n distinct points, then $V_T(\vec{x})$ and $V_U(\vec{x})$ are invertible and

$$V_T(\vec{x})^{-1} = D_0 \cdot H(\vec{d}) \cdot D_0 \cdot V_T(\vec{x})^T \cdot \mathrm{diag}(\vec{c}), \tag{2.2}$$

$$V_U(\vec{x})^{-1} = H(\vec{e}) \cdot D_0 \cdot V_U(\vec{x})^T \cdot \mathrm{diag}(\vec{c}), \tag{2.3}$$

$$V_T(\vec{x})^{-1} = 2 \cdot D_0 \cdot H(\vec{a}) \cdot D_0 \cdot V_U(\vec{x})^T \cdot \mathrm{diag}(\vec{c}), \tag{2.4}$$

$$V_U(\vec{x})^{-1} = 2 \cdot H(\vec{a}) \cdot D_0 \cdot V_T(\vec{x})^T \cdot \mathrm{diag}(\vec{c}), \tag{2.5}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where $H(\vec{f})$ stand for the Hankel matrix

$$H(\vec{f}) = \begin{pmatrix} f_0 & f_1 & \cdots & f_{n-2} & f_{n-1} \\ f_1 & & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & 0 \\ \vdots & f_{n-2} & \cdot^{\cdot^{\cdot}} & \cdot^{\cdot^{\cdot}} & \vdots \\ f_{n-2} & f_{n-1} & \cdot^{\cdot^{\cdot}} & & \vdots \\ f_{n-1} & 0 & \cdots & \cdots & 0 \end{pmatrix}, \quad \vec{f} = (f_k)_{k=0}^{n-1},$$

$D_0 = \mathrm{diag}(\frac{1}{2}, 1, \ldots, 1)$ and $a, c, d, e \in \mathbf{C}^n$ are arbitrary vectors. Using the concept of displacement structure, we can generalize the above matrices to Chebyshev-Vandermonde-like matrices.

For Chebyshev-Vandermonde-like matrices, we extend the definition of basic classes of structured matrices where we choose

$$F = \mathrm{diag}(\frac{1}{x_0}, \frac{1}{x_1}, \cdots, \frac{1}{x_{n-1}}), A = 2 \cdot \sum_{i=1}^{[\frac{n}{2}]} (-1)^{i-1} \cdot (Z_0^T)^{2i-1}. \tag{2.6}$$

Here Chebyshev-Vandermonde matrices R are transformed by the displacement operator

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A = G \cdot B^T \tag{2.7}$$

to the new class of Chebyshev-Vandermonde-like matrices.

Allow $V_T(\vec{x})$ and $V_U(\vec{x})$ to be Chebyshev-Vandermonde matrices as in (2.1), where $x_1, x_2, \ldots, x_n$ are nonzero. From the following recurence relations

$$T_0(\vec{x}) = 1, T_1(\vec{x}) = x, T_n(\vec{x}) = 2x \cdot T_{n-1}(\vec{x}) - T_{n-2}(\vec{x}), \tag{2.8}$$

$$U_0(\vec{x}) = 1, U_1(\vec{x}) = x, U_n(\vec{x}) = 2x \cdot U_{n-1}(\vec{x}) - U_{n-2}(\vec{x}), \tag{2.9}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

it becomes obvious that Chebyshev-Vandermonde matrices possess displacement structure with respect to the displacement operator

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A, \tag{2.10}$$

where

$$F = D_{\frac{1}{x}} = \mathrm{diag}(\frac{1}{x_0}, \frac{1}{x_1}, \ldots, \frac{1}{x_{n-1}})$$

and

$$A = W = \begin{pmatrix} 0 & 2 & 0 & -2 & 0 & \cdots & & & \\ 0 & 0 & 2 & 0 & -2 & \cdots & & & \\ & & \ddots & \ddots & \ddots & \ddots & & & \\ & & & \ddots & \ddots & \ddots & \ddots & \vdots & \\ \vdots & & & & \ddots & \ddots & \ddots & -2 \\ & & & & & \ddots & \ddots & 0 \\ & & & & & & \ddots & 2 \\ 0 & & & \cdots & & & & 0 \end{pmatrix} = 2 \cdot \sum_{i=1}^{[\frac{n}{2}]} (-1)^{i-1} \cdot (Z_0^T)^{2i-1},$$

where, as before, Z_0 is the lower shift circulant matrix.

Lemma 1.2.1 *Let* $\nabla_{\{D_{\frac{1}{x}}\}}(\cdot) : \mathbf{C}^{n \times n} \to \mathbf{C}^{n \times n}$ *be the displacement operator given by (2.9) and* $D_0 = \mathrm{diag}(\frac{1}{2}, 1, \ldots, 1)$. *Then the Chebyshev-Vandermonde matrices satisfy*

$$\nabla_{\{D_{\frac{1}{x}}, W\}}(V_T(\bar{x}) \cdot D_0) = \begin{pmatrix} \frac{1}{x_0} \\ \frac{1}{x_1} \\ \vdots \\ \frac{1}{x_{n-1}} \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{2} & 0 & -1 & 0 & 1 & 0 & -1 \cdots \end{pmatrix}, \tag{2.11}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\nabla_{\{D_{\frac{1}{x}},W\}}(V_U(\vec{x})\cdot D_0)=\begin{pmatrix}\frac{1}{x_0}\\ \frac{1}{x_1}\\ \vdots\\ \frac{1}{x_{n-1}}\end{pmatrix}\cdot(1\quad 0\quad -1\quad 0\quad 1\quad 0\quad -1\cdots). \tag{2.12}$$

Proof. *Substituting in the last equality in (2.8), the same expression isolating $T_{n-2}(\vec{x})$ and proceeding similarly, we obtain the equalities*

$$\frac{1}{\vec{x}}T_n(\vec{x})-2\sum_{k=0}^{\frac{n}{2}-1}(-1)^kT_{n-1-2k}(\vec{x})=\frac{(-1)^{\frac{n}{2}}}{\vec{x}}(n=2,4,6,\ldots),$$

$$\frac{1}{\vec{x}}T_n(\vec{x})-2\sum_{k=0}^{\frac{n-1}{2}-1}(-1)^kT_{n-1-2k}(\vec{x})-(-1)^{\frac{n-1}{2}}=0(n=1,3,5,\ldots).$$

These two equalities are equivalent to (2.11). Formula (2.12) is similarly deduced from recurrence relations (2.9). □

By analogy with (2.11) and (2.12) we shall refer to a matrix R of the

$$\nabla_{\{D_{\frac{1}{x}},W\}}(R)=G\cdot H^T, G,B\in\mathbf{C}^{n\times\alpha}$$

with small α as a Chebyshev-Vandermonde-like matrix. The matrices $\{G,B\}$ are called a $\{D_{\frac{1}{x}},W\}$*-generator* of R, and the smallest number α of columns over all possible generators is called a $\{D_{\frac{1}{x}},W\}$*-dislplacement rank* of R. The following theorem demonstrates that any square matrix can be recovered from its $\{D_{\frac{1}{x}},W\}$-generator. $U(\vec{a})$ denotes an upper triangular Toeplitz matrix with first row $a\in\mathbf{C}$.

Theorem 1.2.2 *Let $\nabla_{\{D_{\frac{1}{x}},W\}}(\cdot)\mathbf{C}^{n\times n}\to\mathbf{C}^{n\times n}$ stand for the displacement operator in (2.10), and let $\vec{g}_m=[g_{m,k}]_{k=1}^{\alpha}\in\mathbf{C}^{1\times\alpha}$, $\vec{h}_m=[h_{m,k}]_{k=1}^{\alpha}\in\mathbf{C}^{\alpha\times 1}(m=1,\ldots,n)$. Then the*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

unique solution R of the equation

$$\nabla_{\{D_{\frac{1}{2}},W\}} = D_{\frac{1}{2}} \cdot R - R \cdot W = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} \cdot \begin{pmatrix} h_1 & h_2 & \cdots & h_n \end{pmatrix} \tag{2.13}$$

is given by

$$R = \sum_{m=1}^{\alpha} \operatorname{diag}(\vec{c}_m) \cdot V_T(\vec{x}) \cdot D_0 \cdot U(\vec{d}_m), \tag{2.14}$$

where $D_0 = \operatorname{diag}(\frac{1}{2}, 1, \cdots, 1)$ *and*

$$c_m = [x_m \cdot g_{k,m}]_{1 \le k \le n}, d_m = \left[2h_{k,m} + 4 \sum_{s=1}^{[\frac{k-1}{2}]} h_{k-2s,m}\right]_{1 \le k \le n}.$$

The matrix R also can be represented as

$$R = \sum_{m=1}^{\alpha} \operatorname{diag}(\vec{c}_m) \cdot V_U(\vec{x}) \cdot U(\vec{a}_m), \tag{2.15}$$

where $a_m = [a_{k,m}]_{k=1}^{n}$ *with* $a_{1,m} = h_{1,m}$, $a_{2,m} = h_{2,m}$ *and* $a_{k,m} = h_{k,m} + h_{k-2,m}$ *for* $(k = 3, 4, \ldots, n)$.

Proof. *Since the spectra of matrices* $D_{\frac{1}{2}}$ *and* W *in (2.10) have no intersection, there is a unique matrix, satisfying (2.13). Let* R *be given by (2.14). Since* $D_{\frac{1}{2}}$ *and* $\operatorname{diag}(\vec{c}_m)$ *obviously commute, and* W *and* $U(\vec{d}_m)$ *commute, being upper triangular Toeplitz matrices, we can write* $\nabla_{\{D_{\frac{1}{2}},W\}}(R) =$

$$\sum_{m=1}^{\alpha} \operatorname{diag}(c_m) \cdot \nabla_{\{D_{\frac{1}{2}},W\}}(V_T(\vec{x}) \cdot D_0)) \cdot U(d_m) =$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\sum_{i=1}^{\alpha} \operatorname{diag}(g_{1,m}x_1, g_{2,m}x_2, \ldots, g_{n,m}x_n) \cdot \begin{pmatrix} \frac{1}{x_0} \\ \frac{1}{x_1} \\ \vdots \\ \frac{1}{x_{n-1}} \end{pmatrix}$$

$$\cdots \begin{pmatrix} \frac{1}{2} & 0 & -1 & 0 & 1 & \cdots \end{pmatrix} \cdot U(d_{1,m}, d_{2,m}, \ldots, d_{n,m}) =$$

$$\sum_{i=1}^{\alpha} \begin{pmatrix} g_{1,m} \\ g_{2,m} \\ \vdots \\ g_{n,m} \end{pmatrix} \cdot \begin{pmatrix} h_{1,m} & h_{2,m} & \cdots & h_{n,m} \end{pmatrix} = \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vdots \\ \vec{g}_n \end{pmatrix} \cdot \begin{pmatrix} \vec{h}_1 & \vec{h}_2 & \cdots & \vec{h}_n \end{pmatrix},$$

and (2.14) follows. Formula (2.15) can be verified similarly. □

1.2.5 Alternative Displacement Operators for Chebyshev-Vandermonde-like matrices

We seek alternate displacement operators for Chebyshev-Vandermonde matrices so that we can reduce them to Cauchy-like matrices. Vandermonde-like matrices are a subclass of Chebyshev-Vandermonde-like matrices. Recall that Chebyshev-Vandermonde-like matrices, as described in the previous section 1.2.4 via the displacement operator of the form

$$\nabla_{\{F,A\}}(R) = F \cdot R - R \cdot A \tag{2.16}$$

where

$$F = D_{\frac{1}{x}}, A = W$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

are diagonal and upper triangular matrices, respectively (see (2.10)). Here we use

$$F = 2D_{\bar{x}} = \mathrm{diag}(2x_0, 2x_1, \dots, 2x_{n-1}),\ A = Y_{\gamma,\delta} = \begin{pmatrix} \gamma & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & \delta \end{pmatrix}, \tag{2.17}$$

$$A = Y_{\gamma,\delta} = Z_0 + Z_0^T + \gamma \vec{e}_0 \vec{e}_0^T + \delta \vec{e}_{n-1} \vec{e}_{n-1}^T$$

or

$$F = 2D_{\bar{x}} = \mathrm{diag}(2x_0, 2x_1, \dots, 2x_{n-1}),\ A = Z_1 + Z_1^T \tag{2.18}$$

where Z_ϕ stands for the lower-shift circulant matrix

$$Z_\phi = \begin{pmatrix} 0 & \cdots & \cdots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}.$$

Lemma 1.2.2 *Let $\nabla_{\{2D_{\bar{x}}, Y_{\gamma,\delta}\}}(\cdot) : \mathbf{C}^{n\times n} \to \mathbf{C}^{n\times n}$ be the displacement operator given by (2.16), (2.17), and $\nabla_{\{2D_{\bar{x}}, Z_1+Z_1^T\}}(\cdot) : \mathbf{C}^{n\times n} \to \mathbf{C}^{n\times n}$ be the displacement operator given by (2.16), (2.18). Then Chebyshev-Vandermonde matrices satisfy*

$$\nabla_{\{2D_{\bar{x}} Y_{\gamma,\delta}\}}(V_T(\bar{x})) = \begin{pmatrix} x_0 - \gamma \\ x_1 - \gamma \\ \vdots \\ x_{n-1} - \gamma \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} T_n(x_0) - \delta T_{n-1}(x_0) \\ T_n(x_1) - \delta T_{n-1}(x_1) \\ \vdots \\ T_n(x_{n-1}) - \delta T_{n-1}(x_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}, \tag{2.19}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$\nabla_{\{2D_{\bar{x}}Y_{\gamma,\delta}\}}(V_U(\bar{x})) = \begin{pmatrix} -\gamma \\ -\gamma \\ \vdots \\ -\gamma \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} U_n(x_0) - \delta U_{n-1}(x_0) \\ U_n(x_1) - \delta U_{n-1}(x_1) \\ \vdots \\ U_n(x_{n-1}) - \delta U_{n-1}(x_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}, \quad (2.20)$$

$$\nabla_{\{2D_{\bar{x}},Z_1+Z_1^T\}}(V_T(\bar{x})) = \begin{pmatrix} x_0 - T_{n-1}(x_0) \\ x_1 - T_{n-1}(x_1) \\ \vdots \\ x_{n-1} - T_{n-1}(x_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} T_n(x_0) - 1 \\ T_n(x_1) - 1 \\ \vdots \\ T_n(x_{n-1}) - 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}, \quad (2.21)$$

$$\nabla_{\{2D_{\bar{x}},Z_1+Z_1^T\}}(V_U(\bar{x})) = \begin{pmatrix} -U_{n-1}(x_0) \\ -U_{n-1}(x_1) \\ \vdots \\ -U_{n-1}(x_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} + \begin{pmatrix} U_n(x_0) - 1 \\ U_n(x_1) - 1 \\ \vdots \\ U_n(x_{n-1}) - 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} \quad (2.22)$$

Proof. From the recurrence relations (2.8) and (2.9) it immediately follows that only the entries in the first and last columns of the matrices on the left hand sides of (2.19)-(2.22) may differ from zero. Calculating these entries, one obtains the assertions of the lemma. □

1.2.6 Discrete Cosine and Discrete Sine Transforms matrices

The following shows that matrices $Y_{\gamma,\delta}$ where $\gamma, \delta \in \{-1, 1\}$ or $\gamma = \delta = 0$, can be diagonalized by Fast Trigonometric Transform matrices.

Lemma 1.2.3 *[KO] Let $Y_{\gamma,\delta}$ be defined as in (2.17). Then*

$$Y_{11} = C \cdot D_C \cdot C^T, Y_{00} = S \cdot D_S \cdot S, \quad (2.23)$$

where

$$C = [\sqrt{\frac{2}{n}}(q_j \cos\frac{(2k-1)\cdot(j-1)\pi}{2n})]_{1\le k,j\le n}, (q_1 = \frac{1}{\sqrt{2}}, q_2 = \ldots = q_n = 1),$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

is the (normalized) Discrete Cosine Transform-II matrix,

$$S = [\sqrt{\frac{2}{n+1}} \sin(\frac{kj\pi}{n+1})]_{1 \leq k,j \leq n}$$

is the (normalized) Discrete Sine Transform-I matrix, and

$$D_C = \mathrm{diag}(2, 2\cos(\frac{\pi}{n}), 2\cos(\frac{2\pi}{n}), \ldots, 2\cos(\frac{(n-1)\pi}{n})),$$

$$D_S = \mathrm{diag}(2, 2\cos(\frac{\pi}{n+1}), 2\cos(\frac{2\pi}{n+1}), \ldots, 2\cos(\frac{n\pi}{n+1})).$$

The next theorem show how to recover any matrix from its $\{2D_{\bar{x}}, Y_{00}\}$-generator.

Theorem 1.2.3 *[KO] Let $\nabla_{\{2D_{\bar{x}}, Y_{00}\}}(\cdot) : \mathbf{C}^{n \times n} \to \mathbf{C}^{n \times n}$ denote the displacement operator in (2.16), (2.17), and assume that*

$$x_0, x_1, \ldots, x_{n-1} \bigcap \{\cos(\frac{\pi}{n+1}), \cos(\frac{2\pi}{n+1}), \ldots, \cos(\frac{n\pi}{n+1})\} = 0. \tag{2.24}$$

Then the unique solution R of the equation

$$\nabla_{\{2D_{\bar{x}}, Y_{00}\}}(R) = \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vdots \\ \vec{g}_n \end{pmatrix} \cdot \begin{pmatrix} \vec{h}_1 & \vec{h}_2 & \cdots & \vec{h}_n \end{pmatrix}, \tag{2.25}$$

with rows $\vec{g}_m = [g_{m,k}]_{k=1}^{\alpha} \in \mathbf{C}^{1 \times \alpha}$, and columns $\vec{h}_m = [h_{m,k}]_{k=1}^{\alpha} \in \mathbf{C}^{\alpha \times 1}$, is given by

$$R = \sum_{m=1}^{\alpha} \mathrm{diag}(\vec{c}_m) \cdot V_U(\vec{x}) \cdot S \cdot D(\vec{d}_m) \cdot S, \tag{2.26}$$

where, as in (2.23), S stand for the DST-I matrix,

$$\vec{c}_k = [\frac{h_{mk}}{U_n x_m}]_{m=1}^{n}, \vec{d}_k = [\frac{\sqrt{\frac{n+1}{2}} \omega_{mk}}{\sin(\frac{mn}{n+1}\pi)}]_{m=1}^{n},$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and where $\vec{\omega}_m = [\omega_{m,k}]_{m=1}^{n} \in \mathbf{C}^{1\times\alpha}$ *are determined from*

$$\begin{pmatrix} \vec{\omega}_1 \\ \vec{\omega}_2 \\ \vdots \\ \vec{\omega}_n \end{pmatrix} = S \cdot \begin{pmatrix} \vec{h}_1^T \\ \vec{h}_2^T \\ \vdots \\ \vec{h}_n^T \end{pmatrix}.$$

Proof. *In view of the assumption in (2.24), the spectra of the matrices* $2D_{\bar{x}}$ *and* Y_{00} *have no intersection (see Lemma 1.2.3). Therefore there is a unique solution* R *of (2.25). Let* R *be given by (2.26). Note that since the points of the second set in (2.24) are zeros of* $U_n(\vec{x})$*, therefore* $U_n(x_m) \neq 0 (i = 0, 1, \ldots, n-1)$ *and* R *is well-defined by (2.26). Also,* $2D_{\bar{x}}$ *and* $D(\vec{c}_m)$ *commute, and that* Y_{00} *and* $S \cdot D(\vec{d}_k) \cdot S$ *commute, since they are diagonizable matrices. Therefore, using (2.21) with* $\gamma = 0$*, we have*

$$\nabla_{\{2D_{\bar{x}}, Y_{00}\}}(R) = \sum_{m=0}^{\alpha} \text{diag}(\vec{c}_m) \cdot \nabla_{\{2D_{\bar{x}}, Y_{00}\}}(V_U(\vec{x})) \cdot S \cdot D(\vec{d}_m) \cdot S =$$

$$= \sum_{k=0}^{\alpha} \text{diag}(\vec{c}_m) \cdot \begin{pmatrix} U_n(x_1) \\ U_n(x_2) \\ \vdots \\ U_n(x_n) \end{pmatrix} \cdot \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} \cdot S \cdot D(\vec{d}_m) \cdot S =$$

$$\sum_{m=1}^{\alpha} \begin{pmatrix} g_{1,m} \\ g_{2,m} \\ \vdots \\ g_{n,m} \end{pmatrix} \cdot \begin{pmatrix} h_{1,m} & h_{2,m} & \cdots & h_{n,m} \end{pmatrix} = \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \vdots \\ \vec{g}_n \end{pmatrix} \cdot \begin{pmatrix} \vec{h}_1 & \vec{h}_2 & \cdots & \vec{h}_n \end{pmatrix},$$

and (2.25) follows. □

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

1.2.7 Chebyshev-Vandermonde matrix Reduction to a Cauchy-like matrix

As observed in [HR], a Cauchy matrix $C(\vec{x},\vec{y}) = [\frac{1}{x_i - y_j}]$ satisfies the equation

$$\nabla_{\{D_{\vec{x}},D_{\vec{y}}\}}(C(\vec{x},\vec{y})) = D_{vecx} \cdot C(\vec{x},\vec{y}) - C(\vec{x},\vec{y}) \cdot = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix},$$

where $D_{\vec{x}} = \text{diag}(x_0, x_1, \ldots, x_{n-1})$ and $D_{\vec{y}} = \text{diag}(y_0, y_1, \ldots, y_{n-1})$ then Cauchy-like matrices were introduced in [HR] as matrices with low $\{D_{\vec{x}}, D_{\vec{y}}\}$-displacemnet rank. In this section we show that Chebyshev-Vandermonde-like matrices become Cauchy-like matrices after multiplication by Discrete Transform matrices. (cf. [H95], [GKO95]).

Theorem 1.2.4 *Let R be a Chebyshev-Vandermonde-like matrix. Then the following statements hold.*

(i)*Let R be given by $\{2D_{\vec{x}}, Y_{11}\}$-generator $\{G, B\}$:*

$$\nabla_{\{2D_{\vec{x}},Y_{11}\}}(R) = 2D_{\vec{x}} \cdot R - R \cdot Y_{11} = G \cdot B^T \quad G, B \in \mathbf{C}^{n \times \alpha} \tag{2.27}$$

Then $R \cdot C$ is a Cauchy-like matrix:

$$\nabla_{\{2D_{\vec{x}},D_C\}}(R \cdot C) = 2D_{\vec{x}} \cdot (R \cdot C) - (R \cdot C) \cdot D_C = G \cdot \hat{B}^T, \tag{2.28}$$

where C and D_C are as in Lemmas 1.2.3 and

$$\hat{B} = C^T \cdot B. \tag{2.29}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(ii) *Let R be given by $\{2D_{\bar{x}}, Y_{00}\}$-generator:*

$$\nabla_{\{2D_{\bar{x}},Y_{00}\}}(R) = 2D_{\bar{x}} \cdot R - R \cdot Y_{00} = G \cdot B^T \quad G, B \in \mathbf{C}^{n\times\alpha}$$

Then $R \cdot S$ is a Cauchy-like matrix:

$$\nabla_{\{2D_{\bar{x}},D_S\}}(R \cdot S) = 2D_{\bar{x}} \cdot (R \cdot S) - (R \cdot S) \cdot D_S = G \cdot \hat{B}^T, \tag{2.30}$$

where S and D_C are as in Lemmas 1.2.3 and

$$\hat{B} = S^T \cdot B.$$

(iii) *Let R be given by $\{2D_{\bar{x}}, Z_1 + Z_1^T\}$-generator:*

$$\nabla_{\{2D_{\bar{x}},Z_1+Z_1^T\}}(R) = 2D_{\bar{x}} \cdot R - R \cdot (Z_1 + Z_1^T) = G \cdot B^T \quad G, B \in \mathbf{C}^{n\times\alpha}$$

Then $R \cdot F^*$ is a Cauchy-like matrix:

$$\nabla_{\{2D_{\bar{x}},D_C\}}(R \cdot F^*) = 2D_{\bar{x}} \cdot (R \cdot F^*) - (R \cdot F^*) \cdot D_C = G \cdot \hat{B^*}^T, \tag{2.31}$$

where $F = \frac{1}{\sqrt{n}}[\exp \frac{2\pi i}{n}(k-1)(j-1)]_{1\le k,j\le n}$ stands for the (normalized) Discrete Fourier matrix, and $D_C = \mathrm{diag}(2, 2\cos(\frac{\pi}{n}), 2\cos(\frac{2\pi}{n}), \ldots, 2\cos(\frac{(n-1)\pi}{n}))$ and

$$\hat{B} = F \cdot B.$$

Proof. Recall that the matrix Y_{11} is diagonized by the DCT-II matrix: $Y_{11} = C \cdot D_C \cdot C^T$, see Lemma 1.2.3. Substituting the latter expression into (2.27) and then multiplying it by C from the right, we get (2.28). Formula (2.30) is deduced from Lemma 1.2.3 by similar arguments. Formula (2.29) follows from the knwon identities

$$Z_1 = F^* \cdot D_1 F, Z_1^T = F^* \cdot D_1^* \cdot F,$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where F is the (*normalized*) DFT matrix and $D_1 = \mathrm{diag}(1, \exp(\frac{2\pi i}{n})), \ldots, \exp(\frac{2\pi i}{n})(n-1)$.

□

Formulas (2.30) and (2.31) suggest two efficient algorithms for solving linear systems with Chebyshev-Vandermonde matrices via transformation into Cauchy-like matrices, by exploiting the FST and the FFT, respectively.

1.2.8 Structured Matrices and Their Generators

Definition 1.2.2 *$T = (t_{i,j})_{i,j=0}^{n-1} \in \mathbf{F}^{n\times n}$ is a* Toeplitz matrix *if $t_{i+1,j+1} = t_{i,j}, i, j = 0, \ldots n-2$. $\overline{H} = (\overline{h}_{i,j=0}^{n-1}) \in \mathbf{F}^{n\times n}$ is a* Hankel matrix *if $\overline{h}_{i+1,j-1} = \overline{h}_{i,j}$, $i = 0, \ldots, n-2$; $j = 1, \ldots, n-1$. $V(\vec{t}) = (t_i^j)_{i,j=0}^{n-1} \in \mathbf{F}^{n\times n}$ is a* Vandermonde matrix, *and $C(\vec{s}, \vec{t}) = (\frac{1}{s_i - t_j})_{i,j=0}^{n-1} \in \mathbf{F}^{n\times n}$ is a* Cauchy matrix, *for any pair of vectors $\vec{s} = (s_i)_{i=0}^{n-1}$, $\vec{t} = (t_j)_{j=0}^{n-1}$ where $s_i \neq t_j$ for every pair (i,j). (Many authors use the name "Vandermonde matrix" for $V^T(\vec{t})$.)*

Definition 1.2.3 *$\vec{e} \in \mathbf{F}^{n\times \ell}$ is the i-th* coordinate vector *having 1 as its i-th coordinate and 0 as all its other coordinates. 0 is the null vector of appropriate dimension. $J = (\vec{e}_i)_{i=n-1}^{0}$ is the reflection matrix, $J\vec{v} = (v_i)_{i=n-1}^{0}$ for any vector $\vec{v} = (v_i)_{i=0}^{n-1}$. $D_{\vec{v}}$ or $D(\vec{v}) = \mathrm{diag}(v_0, \ldots, v_{n-1}) = (v_i \vec{e}_i)_{i=0}^{n-1} \in \mathbf{F}^{n\times n}$ is the* diagonal matrix *with diagonal entries $v_0, \ldots, v_{n-1}$. $\vec{v}^k = (v_i^k)_{i=0}^{n-1}$, for $\vec{v} = (v_i)_{i=0}^{n-1}$ and an integer k.*

Definition 1.2.4 *$Z_f = (\vec{e}_l, \ldots, \vec{e}_{n-1}, f\vec{e}_0) \in \mathbf{F}^{n\times n}$ (for a scalar f) is the* unit f-circulant matrix, *called the* unit circulant *if $f = 1$. ($Z_f\vec{v} = (fv_{n-1}, v_0, \ldots, v_{n-2})$ for $\vec{v} = (v_i)_{i=0}^{n-l}$.) $Z_f(\vec{v}) = \sum_{i=0}^{n-1} v_i Z_f^i$ is an* f-circulant matrix. *$Z_1(\vec{v})$ is a* circulant matrix, *and $Z_0(\vec{v})$ is a* lower triangular Toeplitz matrix. *(cf. section 1.2.2)*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark 1.2.1 *To show more options in choosing the operators associated with structured matrices, we allow any f when we define the matrices Z_f and $Z_f(\vec{v})$. In, our presentation, in this thesis, however, we could have always stayed with the triangular Toeplitz case $f = 0$ or with $f = 1$ (the circulant case) and later on (in (2.33) and in Remark 1.2.3 and 5.4.1, and Proposition 5.4.3) $e = -1$ for the matrices $C_e(\vec{v})$ (the skew circulant case).*

Fact 1.2.1 $J^2 = I$, $Z_0^n = 0$, $Z_{1/f}^T$ *for* $f \neq 0$, $JZ_fJ = Z_f^T$ *for any* f . $JD(\vec{v})JD(J\vec{v})$ *for any* $\vec{v}$.

Fact 1.2.2 *TJ and JT are Hankel matrices if T is a Toeplitz matrix, and vice versa, $\overline{H}J$ and $J\overline{H}$ are Toeplitz matrices if $\overline{H}$ is a Hankel matrix.*

Next, we will show some generalizations of the classes of the structured matrices of Definition 3.2.1 in the form of (scaled or shifted) bilinear or trilinear combinations X of the structured matrices of Definition 1.2.2 and diagonal matrices. Equivalently, such matrices are the solutions to Sylvester's basic matrix equation:

$$XK + LX = GH^T \tag{2.32}$$

Here K, $L \in \mathbf{F}^{n\times n}$ and G, $H \in \mathbf{F}^{n\times \ell}$ are fixed matrices, and ℓ is small relative to n. We will write $T(G,H)$, $V(\vec{t}, G, H)$ (with some subscripts) and $C(\vec{s}, \vec{t}, G, H)$ for matrices whose structure generalizes one of T, $V(\vec{t})$ and $C(\vec{s}, \vec{t})$, respectively.

Definition 1.2.5 *Given two scalars e and f, $ef \neq 1$, two vectors $\vec{s}$, $\vec{t} \in \mathbf{F}^{n\times \ell}$ with $s_i \neq t_j$ for all pairs (i,j), and a pair of $n \times \ell$ matrices $G = (g_i)_{i=1}^{\ell}$, $H = (h_i)_{i=1}^{\ell} \in \mathbf{F}^{n\times \ell}$,*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

we write

$$T_f = T_f(G,H) = (fZ_f(T_f\vec{e}_0) + \frac{1}{1-ef}\sum_{k=1}^{\ell} Z_f(\vec{g}_k)Z_e^T(\vec{h}_k))Z_{1/f}^T, \quad for\ f \neq 0, \tag{2.33}$$

$$T_0 = T_0(G,H) = Z_0(T_0\vec{e}_0) + \sum_{k=1}^{\ell} Z_0(\vec{g}_k)Z_0^T(Z_0\vec{h}_k), \tag{2.34}$$

$$V_f(\vec{t},G,H) = D((\vec{t} - f\vec{t}^{n+1})^{-1})\sum_{k=1}^{\ell} D(\vec{g}_k)V(\vec{t})Z_f^T(\vec{h}_k), \quad ft_i^{n+1} \neq t_i, i = 0,\dots,n-1, \tag{2.35}$$

$$C(\vec{s},\vec{t},G,H) = \sum_{k=1}^{\ell} D(\vec{g}_k)C(\vec{s},\vec{t})D(\vec{h}_k). \tag{2.36}$$

The matrix pair (G,H) is called a (K,L)-generator (or just a generator) of length ℓ for a matrix X where $K = -L = Z_f$ for $X = T_f$ and for any scalar f; $K = -Z_f^T$, $L = D^{-1}(\vec{t})$ for $X = V_f(\vec{t},G,H)$, and $K = -D(\vec{t})$, $L = D(\vec{s})$ for $X = C(\vec{s},\vec{t},G,H)$. For a fixed (K,L), the minimum ℓ in all (K,L)-generators of X is called the (K,L)-rank or a generator rank of X and is denoted by $r_{K,L}(X)$. The operator $X \to XK + LX$ is called a basic operator for X, and the pair (K,L) is called a basic matrix pair for X.

The latter definition is motivated by the following results, defining matrix structure in terms of the associated linear operators of scaling and displacement (shift) (cf. Remark 1.2.5 at the end of this section).

Theorem 1.2.5 *A matrix X satisfies Sylvester's matrix equation (2.32) if (G,H) is a (K,L)-generator of X for the triples X, K, L defined above, that is, if*

a) $T_fZ_f - Z_fT_f = GH^T$ under (2.33), (2.34),

b) $(JT_f)Z_f - Z_f^T(JT_f) = (JG)H^T$, $(T_fJ)Z_f^T - Z_f(T_fJ) = GH^TJ$, under (2.33), (2.34),

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

c) $D^{-1}(\vec{t})V_f(\vec{t},G,H) - V_f(\vec{t},G,H)Z_f^T = GH^T$ *under (2.35), and*

d) $D(\vec{s})C(\vec{s},\vec{t},G,H) - C(\vec{s},\vec{t},G,H)D(\vec{t}) = GH^T$ *under (2.36).*

Furthermore, $r_{K,L}(X) = \text{rank } (XK + LX)$ *in all these cases.*

Proof. Parts a), c), d) of Theorem 1.2.5 easily follow from Theorems 1.1, 2.1 and 3.1 of [GO94] (cf. also [GO92]) and Theorem 2.11.3a) of [BP94]. Part b) follows from part a) and Fact 1.2.1. □

We immediately obtain short (K, L)-generators for the matrices T, H, $V(\vec{t})$ and $C(\vec{s},\vec{t})$ of Definition 1.2.2, and for all scalars f we deduce that $r_{Z_f,-Z_f}(T) \le 2$, $r_{Z_f,-Z_f^T}(JT) \le 2$, $r_{Z_f^T,-Z_f}(TJ) \le 2$, $r_{-Z_f^T,D^{-1}(\vec{t})}(V(\vec{t})) \le 1$ and $r_{-D(\vec{t}),D(\vec{s})}(C(\vec{s},\vec{t})) \le 1$. By generalizing f these observations, we will say that for all scalars f the matrices T_f of (2.33), (2.34) are *Toeplitz-like*, JT_f and T_fJ are *Hankel-like*, $V_f(\vec{t},G,H)$ of (2.35) are *Vandermonde-like*, and $C(\vec{s},\vec{t},G,H)$ of (2.36) are *Cauchy-like* if ℓ is small relatively to n, say if ℓ is bounded by a small fixed constant.

Remark 1.2.2 *[BP94], [GO94]. The solution to Sylvester's matrix equation (2.32) is unique for a fixed generator* (G, H) *and each basic pair* (K, L) *specified above. In the Vandermonde-like and Cauchy-like cases of (2.35), (2.36), there exists such a (unique) solution for any fixed pair of matrices* G, $H \in \mathbf{F}^{n\times r}$. *In the Toeplitz/ Hankel-like case of (2.33), (2.34), there exists a (unique) solution if and only if* $\sum_{k=1}^{\ell} Z_f(\vec{g}_k)Z_{1/f}(\vec{h}_k)^T = 0$ *for* $f \ne 0$ *or* $\sum_{k=1}^{\ell} Z_f^T(\vec{g}_k)\vec{h}_k = \sum_{k=1}^{\ell} Z_f(J\vec{h}_k)\vec{g}_k = 0$ *for* $f = 0$. *Thus, one may take the dual point of view, that is, define the classes of structured*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

matrices by equation (2.32) and then apply Theorem 1.2.5 as a recovery theorem, defining some explicit expression for the solution matrix X satisfying this equation.

Structured matrices X of (2.33)-(2.36) can be completely represented by the $2\ell n$ entries of their (K, L)-generators (G, H), rather than by their own n^2 entries (in the Toeplitz/ Hankel-like case, we need in addition the n entries of the first or last column or row of the matrix). Such a *compressed representation* of a matrix is not unique. Moreover, ℓ may exceed $r_{K,L}(X)$, but this can be repaired by using our generator compression techniques:

Fact 1.2.3 *Let X stand for T_f, $V_f(\vec{t}, G, H)$ or $C(\vec{s}, \vec{t}, G, H)$ of (2.33)-(2.36). Let a (K, L)-generator (G, H) of a length ℓ for a matrix X and the (K, L)-rank of X, $r = r_{K,L}(X)$, be given as an input, together with f and $T_f\vec{e}_0$, f and $\vec{t}$, or $\vec{s}$ and $\vec{t}$, respectively. Then it is sufficient to use $O(\ell^2 n)$ ops in order to compute a (K, L)-generator of length r for X.*

Proof. See Proposition A.6 of [P92] or Problem 2.2.llb of [BP94]. □

Remark 1.2.3 *Theorem 1.2.5 can be used as a springboard for the definition of the same or closely related classes of structured matrices based on the Sylvester equation (2.32) with distinct basic pairs (K, L). For instance, the matrices $Z_f - Z_e$, have rank 1 for $e \neq f$, and this immediately implies close correlation between the classes of $\{T_f(G, H)\}$ and $\{T_e, (G, H)\}$ of Toeplitz-like matrices as well as between $\{V_f(\vec{t}, G, H)\}$ and $\{V_e, (\vec{t}, G, H)\}$ for distinct e and f, for a fixed ℓ, and for varying G, $H \in \mathbf{F}^{n \times \ell}$ Parts a) and b) of Theorem 1.2.5 can be immediately extended to define a $(Z_e^T, -Z_f)$-generator for T_f; a $(Z_e, -Z_f^T)$-generator for JT_f, and a $(Z_e^T, -Z_f)$-generator for $T_f J$.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Further variations of the matrix equation (2.32) can be obtained by the transposition of the matrices on both its sides, by its multiplication by a scalar, e.g. by -1, and/or (cf. [GKO95]) by its pre- and post-multiplications by some selected matrices. In this way, we obtain from (2.32) that

$$X^T L^T + K^T X^T = HG^T, \tag{2.37}$$

$$\hat{X}\hat{K} + \hat{L}\hat{X} = \hat{G}\hat{H}^T, \tag{2.38}$$

where $\hat{X} = UXW$, $\hat{K} = W^{-1}KW$, $\hat{L} = ULU^{-1}$, $\hat{G} = UG$, $\hat{H} = H^T W$, and U and W is any fixed pair of nonsingular matrices. (Observe that the spectra of the matrices K and L are preserved in the transition to K and L.) In particular, for $U = W = J$, we obtain

$$(JXJ)(JKJ) + (JLJ)(JXJ) = JGH^T J. \tag{2.39}$$

We may also combine (2.37) and (2.38) together.

In all these variations, the representations (2.33)-(2.36) of a matrix X are immediately extended to the matrices X^T, $\hat{X}$ and $\hat{X}^T$, in particular, *we immediately represent compactly the matrices UT_fW, $UV_f(\vec{t}, G, H)W$, $UV_f^T(\vec{t}, G, H)W$, $UC(\vec{s}, \vec{t}, G, H)W$ for any fixed pair of nonsingular matrices U and W.*

Tables 2-4 summarize some particular, classes of structured matrices together with the associated basic matrix pairs (K, L), which we obtained based on Theorem 1.2.5, equations (2.37) and (2.39) and Fact 1.2.2. Some other variations of K, L of (2.32) and the respective extensions of Theorem 1.2.5 can be found in [BP94], pp. 187-188, [KO96], [KO97], [KS99].

Definition 1.2.6 *The basic matrix pairs (K, L) and (L, K) are called dual and if, in addition, $K = -L$, then self-dual.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Table 1.2: Toeplitz/Hankel-like Matrices and Their Basic Matrix Pairs

X	T_f	T_f^T	JT_f	T_fJ
K	Z_f	$-Z_f^T$	Z_f	Z_f^T
L	$-Z_f$	Z_f^T	$-Z_f^T$	$-Z_f$

Table 1.3: Vandermonde-like Matrices and Their Basic Matrix Pairs

X	$V_f(\vec{t}, G, H)$	$V_f^T(\vec{t}, G, H)J$	$JV_f(\vec{t}, G, H)J$	$JV_f^T(\vec{t}, G, H)J$
K	$-Z_f^T$	$D^{-1}(\vec{t})$	$-Z_f$	$D^{-1}(J\vec{t})$
L	$D^{-1}(\vec{t})$	$-Z_f$	$D-1(J\vec{t})$	$-Z_f^T$

Table 1.4: Cauchy-like Matrices and Their Basic Matrix Pairs

X	$C(\vec{s}, \vec{t}, G, H)$	$C(\vec{s}, \vec{t}, G, H)$
K	$-D(\vec{t})$	$D(\vec{s})$
L	$D(\vec{s})$	$-D(\vec{t})$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark 1.2.4 *All basic matrix pairs (K, L) appear in Tables 2-4 together with their dual pairs (L, K) (up to scaling by -1). The basic matrix pairs (K, L) of Table 2 are self-dual.*

Remark 1.2.5 *For $K = -L = Z_f$ and $K = -L = Z_f^T$, the basic operator $X \to XK + LX$ maps the matrix X to the difference between the two displacements (shifts) of X (into two directions orthogonal to each other). $r_{K,L}(X)$, the rank of such a difference, is commonly called the displacement rank of X. The nomenclature is due to the seminal pioneering paper) [KKM], where the Toeplitz type structure was first formally introduced (based on the Stein type equations $X + LXM = GH^T$ with $K = -M^T = Z_0^T$ and $K = -M^T = Z_0$).*

1.3 Correlation of Polynomial Computations to Computations with Structured Matrices

The entries of the matrices are related to each other via some operators of displacement and/or scaling (e.g. $t_{i,j}$ for a Toeplitz matrix $T = (t_{i,j})$ and $h_{i,j}$ for a Hankel matrix $H = (h_{i,j})$ are invariant in their displacement along the diagonal or antidiagonal directions, respectively). All entries of such an $n \times n$ structured matrix can be expressed via a few parameters (from n to $2n$, versus n^2 for a general $n \times n$ matrix). Such matrices can be multiplied by vectors fast as this task reduces to some basic operations with polynomials. For instance, polynomial product

$$(\sum_i u_i x^i)(\sum_j v_j x^j) = \sum_k w_k x^k,$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Toeplitz-by-vector product $T\vec{v} = \vec{w}$, and Hankel-by-vector product $H\vec{v} = \vec{w}$ can immediately be made equivalent by matching properly the entries of the matrices T, H and the vector $\vec{u} = (u_i)$ (see [BP94], pp. 132-133). Likewise the product

$$V(\vec{x})\vec{p} = \vec{v} \tag{3.1}$$

represents the vector $\vec{v} = (v_i)$ of the values of the polynomial $p(x) = \sum_j p_j x^j$ on a node set $\{x_i\}$. Consequently, the known fast algorithms for the relevant operations with polynomials also apply to the associated matrix operations and vice/versa. In particular (cf. [BP94]), $O(n \log n)$ ops suffice for polynomial and Toeplitz(Hankel)-by-vector multiplication, and $O(n \log^2 n)$ ops suffice for multipoint polynomial evaluation (p.e.) as well as for the equivalent operations with the matrix $V(\vec{x})$ and the vectors $\vec{p}$ and $\vec{v}$ of (3.1), assuming the input size $O(n)$ in all cases. Here and hereafter, *"ops"* stand for "arithmetic operations", *"p.e."* for interpolation and multipoint polynomial evaluation and *"p.i."* for polynomial interpolation. An important special case of (3.1), $\vec{x} = \vec{w} = (w_n^i)_{i=0}^{n-1}$ is the vector of the n-th roots of 1, $w_n = \exp(2\pi\sqrt{-1}/n)$, $w_n^n = 1$. In this case, we write $F = V(\vec{w})/\sqrt{n}$, and p.e. turns into discrete Fourier transform, takes $O(n \log n)$ ops (due to FFT), and allows numerically stable implementation, in sharp contrast with the case of general $V(\vec{x})$. The numerical stability requirement is practically crucial and motivated the design of fast *approximation algorithms* for the p.e. and p.i. [PLST93], [PZHY97], which rely on expressing $V(\vec{x})$ of (3.1) via Cauchy matrices, e.g. as follows:

$$V(\vec{x}) = \frac{1}{\sqrt{n}} \operatorname{diag}(1 - x_i^n)_{i=0}^{n-1} C(\vec{x}, \vec{w}) \operatorname{diag}(w_i)_{i=0}^{n-1} F. \tag{3.2}$$

Here for any vector $\vec{y} \in \mathbf{C}^n$ and hereafter $\operatorname{diag}(h_i)_{i=0}^{n-1} = D(\vec{h})$ for $\vec{h} = (h_i)_{i=0}^{n-1}$ denotes

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

the $n \times n$ diagonal matrix with diagonal entries $h_0, \ldots, h_{n-1}$. (3.2) reduces the operations of (3.1) with $V(\vec{x})$ to ones with $C(\vec{x}, \vec{w})$, which brings us to Trummer's problem (that is, the problem of multiplication of an $n \times n$ Cauchy matrix by a vector), our next topic. Its solution by Multipole Algorithm leads to p.e. and p.i. algorithms based on (3.2), which are both fast in terms of ops used and (according to experimental tests of [PZHY97]) numerically stable even on the inputs where the known $O(n \log^2 n)$ algorithms fail numerically, due to roundoff errors. Reducing p.e. and p.i. to Trummer's problem for $C(\vec{x}, \vec{y})$, we have the power of choosing $\vec{y} = c\vec{w}$, for the vector $\vec{w}$ of roots of 1 defined above and for any scalar $c \neq 0$; furthermore, we may vary vector $\vec{x}$, that is, we may linearly map $\vec{x}$ into

$$\vec{y} = a\vec{x} + b\vec{e}, \quad \vec{e} = (1)_{i=0}^{n-1}, \tag{3.3}$$

where we may choose any scalars $a \neq 0$ and b. With such a choice of $\vec{y}$, (3.2) reduces p.e. and p.i. to FFT and the solution of Trummer's problem. Furthermore, we have substantial control over the input vectors $\vec{x}$ and $\vec{y}$ of such a Trummer's problem, which will enable us to facilitate a solution.

The resulting improvement of p.e. and p.i. is but one of several known examples where transformations among various classes of structured matrices facilitate substantially the design of efficient algorithms. (The idea and first major examples of using such transformations for the algorithm design are due to [P90] (cf. also [BP94], [GKO95])).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 2

Trummer's Problem

2.1 The Problem Stated

Trummer's problem is the problem of multiplication of an $n \times n$ Cauchy matrix by a vector.

2.1.1 Some Subjects Related to Trummer's Problem

- p.e. and p.i.
- integral equations of potential theory (solution to the Laplace equation) (cf. [Rok85])
- conformal mappings (cf. [T86])
- Riemann Zeta function (cf. [OS88])
- particle simulations using Multipole algorithm (cf. [GR87])

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.1.2 Trummer's Problem, Known Algorithms

- $O(n^2)$ arithmetic operations (ops); straightforward method

$$C(\vec{s},\vec{t})\vec{v} = \left(\sum_{j=0}^{n-1} \frac{v_j}{s_i - t_j}\right)_{i=0}^{n-1}.$$

- $O(n \log^2 n)$ ops; Gerasoulis (1987)(cf. [Ger87] and [GGS]),

  reduces Trummer's problem to polynomial multiplication, p.e. and p.i.

  Difficultly: numerical instability due to relying on fast p.e. and p.i.

2.2 Trummer's Problem: Fast Unstable Solution

The solution of Trummer's problem is required in many areas of scientific and engineering computing (see bibliography in [BP94], p.260; [PACLS,a]). The straightforward algorithm solves Trummer's problem in $O(n^2)$ ops. Let us next show $O(n\log^2 n)$ algorithms.

Definition 2.2.1 *We repeat the definitions for clarity,* $H(\vec{t}) = (h_{i,j})_{i,j=0}^{n-1}$, $h_{i,j} = t_{i+j}$ *for* $i + j \leq n-1$,$h_{i,j} = 0$ *for* $i+j \geq n$. W^{-1}, W^T*and* W^{-T} *denote the inverse, the transpose, and the transpose of the inverse of a matrix (or vector)* W, *respectively.*

Definition 2.2.2 $p_{\vec{t}}(x) = \prod_{j=0}^{n-1}(x - t_j)$, $p'_{\vec{t}}(x) = \sum_{i=0}^{n-1} \prod_{\substack{j=0 \\ (j\neq i)}}^{n-1} (x - t_j)$.

Definition 2.2.3 $D(\vec{s},\vec{t}) = \mathrm{diag}(p_{\vec{t}}(s_i))_{i=0}^{n-1} = \mathrm{diag}(\prod_{j=0}^{n-1}(s_i - t_j))_{i=0}^{n-1}$.

$D'(\vec{t}) = \mathrm{diag}(p'_{\vec{t}}(t_i))_{i=0}^{n-1} = \mathrm{diag}(\prod_{\substack{j=0 \\ (j\neq i)}}^{n-1} (t_i - t_j))_{i=0}^{n-1}$.

Theorem 2.2.1 [FHR93](cf. also [Ger87]) *Let* $s_i \neq t_j$, $i, j = 0, 1, \ldots, n-1$. *Then*

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1} V(\vec{s}) H(\vec{t}) V(\vec{t})^T, \tag{2.1}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}V(\vec{s})V(\vec{t})^{-1}D'(\vec{t}). \tag{2.2}$$

Proof: Use the following inversion formulae:

$$C(\vec{c},\vec{d})^{-1} = -D_1 \cdot C(\vec{c},\vec{d})^T \cdot D_2, \tag{2.3}$$

$$V(\vec{c})^{-1} = H(\vec{s}) \cdot V_n(\vec{c})^T \cdot D^{-1}, \tag{2.4}$$

Let $\vec{c}$, $\vec{d}$, $\vec{s}$ be arbitrary vectors of length n, D, D_1, D_2 be arbitrary diagonal matrices of size $n \times n$ and $\vec{p}$ be the coefficient vector for some given polynomial.

If $C(\vec{c},\vec{d})\vec{x} = \vec{y}$, let

$$V_n(\vec{c})\vec{p} = D_1\vec{y}, \text{ and } V_n(\vec{d})\vec{p} = D_2\vec{x}$$

which implies

$$D_1^{-1}V_n(\vec{c})\vec{p} = \vec{y}, \text{ and } \vec{p} = V_n(\vec{d})^{-1}D_2\vec{x}.$$

By combining the two latter expressions with $C(\vec{c},\vec{d})\vec{x} = \vec{y}$, we get

$$D_1^{-1}V_n(\vec{c})V_n(\vec{d})^{-1}D_2 = C(\vec{c},\vec{d}),$$

which proves (2.2) for $\vec{c} = \vec{s}$ and $\vec{d} = \vec{t}$ □

Substitute (2.4) into the previous result to get

$$D_1^{-1}V_n(\vec{c})H(\vec{s})V_n(\vec{c})^T D^{-1}D = C(\vec{c},\vec{d}),$$

which implies

$$D_1^{-1}V_n(\vec{c})H(\vec{s})V_n(\vec{c})^T = C(\vec{c},\vec{d})$$

and this proves (2.1) for $\vec{c} = \vec{s}$ and $\vec{d} = \vec{t}$ □

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Theorem 2.2.1 reduces Trummer's problem essentially to the evaluation of the coefficients of $p_{\bar{t}}(x)$ and then the values of $p_{\bar{t}}(s_i)$ and $p'_{\bar{t}}(t_j)$ followed by multiplication of structured matrices $V(\bar{s})$, $V^T(\bar{t})$, $H(\bar{t})$ and $V(\bar{t})^{-1}$ by vectors. Both of (2.1) and (2.2) lead to $O(n \log^2 n)$ algorithms [Ger87]. As numerically unstable, however, they are nonpractical for numerical computations for larger n.

2.3 Fast and Numerically Stable Approximate Solution, Its Limitations

Presently the algorithm of choice for practical solution of Trummer's problem is the celebrated Multipole Algorithm, which belongs to the class of hierarchical methods (for bibliography see [PACLS,a], [BP94], pp. 261–262). The algorithm approximates the solution in $O(n)$ ops in terms of n, and works efficiently for a large class of inputs but also has some "difficult" inputs. The basis for the algorithm is the following expansions.

$$\frac{1}{s_i - t_j} = \frac{1}{s_i}\sum_{k=0}^{\infty}(t_j/s_i)^k = -\frac{1}{t_j}\sum_{k=0}^{\infty}(s_i/t_j)^k. \tag{3.1}$$

Proof of basis for Multipole algorithm.

Consider an infinite geometric series and the corresponding partial sum:

$$\sum_{n=1}^{\infty} ar^{n-1} = \frac{a}{1-r}, \quad \sum_{n=1}^{N} ar^{n-1} = \frac{a(1-r)^n}{1-r} \text{ for } |r| < 1$$

We take the limit of the partial sum as $n \to \infty$.

$$\lim_{n\to\infty} \frac{a(1-r)^n}{1-r} = \frac{a}{1-r} - \lim_{n\to\infty} \frac{ar^n}{1-r} = \frac{a}{1-r}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

If we equate our Cauchy matrix $C(\vec{s},\vec{t})$ with the geometric series expansion, we get

$$\frac{1}{s_i - t_j} = \frac{\frac{1}{s_i}\cdot(s_i)}{(1-\frac{t_j}{s_i})\cdot(s_i)} = \frac{\frac{1}{s_i}}{1-\frac{t_j}{s_i}} = \frac{a}{1-r} \text{ where } a = \frac{1}{s_i} \text{ and } r = \frac{t_j}{s_i}.$$

This implies that

$$\sum_{n=1}^{\infty} ar^{n-1} = \sum_{n=1}^{\infty} \frac{1}{s_i}(\frac{t_j}{s_i})^{n-1} = \frac{1}{s_i}\sum_{n=1}^{\infty}(\frac{t_j}{s_i})^{n-1} \quad if \quad |t_j/s_i| < 1$$

Similarly, we have

$$\frac{1}{s_i - t_j} = -\frac{1}{t_j}\sum_{n=1}^{\infty}(\frac{s_i}{t_j})^{n-1} \quad if \quad |s_i/t_j| < 1$$

□

The prefix of the former (latter) geometric series (3.1) up to the term $(t_j/s_i)^\kappa$ (*resp.*$(s_i/t_j))^\kappa$ for a fixed moderately large κ already approximates $\frac{1}{s_i - t_j}$ well if, say, $|t_j/s_i| < 1/2$ (resp. $|s_i/t_j| < 1/2$). Substitute such a prefix into the expressions $C(\vec{s},\vec{t})\vec{v} = (\sum_{j=0}^{n-1}\frac{v_j}{s_i - t_j})_{i=0}^{n-1}$ and obtain their approximations, say, by $-\sum_{j=0}^{n-1}(v_j/t_j)\sum_{k=0}^{\kappa}(s_i/t_j)^k = \sum_{k=0}^{\kappa}A_k s_i^k$, where $A_k = -\sum_{j=0}^{n-1} v_j/t_j^{k+1}$. For $n \times n$ matrix $C(\vec{s},\vec{t})$, the computation of such approximations for all i requires $O(n\kappa)$ ops and is stable numerically. The approximation errors are small already for moderate κ if one of the dual ratios $|t_j/s_i|$ and $|s_i/t_j|$ is small. "Difficult" inputs have these dual ratios close to 1 for some i, j. In such irregular cases, some tedious hierarchical techniques give partial remedy. We, however, will treat the irregularity by general regularization techniques of transformation of the input (Cauchy) matrix.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

2.4 New Transformations of a Cauchy Matrix and Trummer's Problem

To extend the domain of inputs where the Fast Multipole Algorithm can be applied, to include any Cauchy matrix $C(\vec{s},\vec{t})$ we will reduce Trummer's problem for $C(\vec{s},\vec{t})$ to ones for $C(\vec{s},\vec{q})$ and/or $C(\vec{q},\vec{t})$ where $\vec{q}$ is a vector of our choice. We will rely on the next theorem (cf. also Remark 2.5.2).

Theorem 2.4.1 *[PACLS,a]. For a triple of n-dimensional vectors $\vec{q}=(q_i)_{i=0}^{n-1}$, $\vec{s}=(s_j)_{j=0}^{n-1}$, $\vec{t}=(t_k)_{k=0}^{n-1}$, where $q_i \neq s_j$, $s_j \neq t_k$, $t_k \neq q_i$ for $i,j,k=0,\ldots,n-1$, we have the following matrix equations:*

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}V(\vec{s})V(\vec{q})^{-1}D(\vec{q},\vec{t})C(\vec{q},\vec{t}), \tag{4.1}$$

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}D(\vec{s},\vec{q})C(\vec{s},\vec{q})D'(\vec{q})^{-1}D(\vec{q},\vec{t})C(\vec{q},\vec{t}), \tag{4.2}$$

$$C(\vec{s},\vec{t}) = C(\vec{s},\vec{q})D(\vec{q},\vec{s})V(\vec{q})^{-T}V(\vec{t})^{T}D(\vec{t},\vec{s})^{-1}, \tag{4.3}$$

$$C(\vec{s},\vec{t}) = -C(\vec{s},\vec{q})D(\vec{q},\vec{s})D'(\vec{q})^{-1}C(\vec{q},\vec{t})D(\vec{t},\vec{q})D(\vec{t},\vec{s})^{-1}. \tag{4.4}$$

Proof of equation (4.1): From (2.1) we take the inverse to get

$$C(\vec{b},\vec{t})^{-1} = V(\vec{t})^{-T}H(\vec{t})^{-1}V(\vec{b})^{-1}D(\vec{b},\vec{t}) \text{ for } \vec{s}=\vec{b}.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Substitute this inverse Cauchy matrix equation and (2.1) into

$$C(\vec{s},\vec{t}) = C(\vec{s},\vec{t})C(\vec{b},\vec{t})^{-1}C(\vec{b},\vec{t})$$

to obtain

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}V(\vec{s})H(\vec{t})V(\vec{t})^{T}V(\vec{t})^{-T}H(\vec{t})^{-1}V(\vec{b})^{-1}D(\vec{b},\vec{t})^{-1}C(\vec{b},\vec{t})$$

which reduces to

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}V(\vec{s})V(\vec{q})^{-1}D(\vec{q},\vec{t})^{-1}C(\vec{q},\vec{t}) \text{ for } \vec{b} = \vec{q}.$$

This proves (4.1) □

Proof of equation (4.2): From (2.2) we get

$$C(\vec{s},\vec{b}) = D(\vec{s},\vec{b})^{-1}V(\vec{s})V(\vec{b})^{-1}D'(\vec{b}) \text{ for } \vec{t} = \vec{b}.$$

Isolate Vandermonde matrices to get

$$V(\vec{s})V(\vec{b})^{-1} = D(\vec{s},\vec{b})C(\vec{s},\vec{b})D'(\vec{b})^{-1}.$$

Substitute the Vandermonde pair into (4.1) to get

$$C(\vec{s},\vec{t}) = D(\vec{s},\vec{t})^{-1}D(\vec{s},\vec{q})C(\vec{s},\vec{q})D'(\vec{q})^{-1}D(\vec{q},\vec{t})C(\vec{q},\vec{t}) \text{ for } \vec{b} = \vec{q}.$$

This proves (4.2) □

Proof of equation (4.3): Since $C(\vec{s},\vec{t}) = -C(\vec{t},\vec{s})^{T}$, rewrite (4.1) as

$$-C(\vec{s},\vec{t}) = C(\vec{t},\vec{s})^{T} = (D(\vec{s},\vec{t})^{-1}V(\vec{t})V(\vec{q})^{-1}D(\vec{q},\vec{s})C(\vec{q},\vec{s}))^{T},$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

rewrite as

$$-C(\vec{s},\vec{t}) = C(\vec{q},\vec{s})^T D(\vec{q},\vec{s}) V(\vec{q})^{-T} V(\vec{t})^T (D(\vec{t},\vec{s})^{-1} \text{ and substitute } C(\vec{s},\vec{q}) = -C(\vec{q},\vec{s})^T,$$

rewrite again as

$$-C(\vec{s},\vec{t}) = -C(\vec{q},\vec{s})^T D(\vec{q},\vec{s}) V(\vec{q})^{-T} V(\vec{t})^T (D(\vec{t},\vec{s})^{-1},$$

and finally

$$C(\vec{s},\vec{t}) = C(\vec{q},\vec{s})^T D(\vec{q},\vec{s}) V(\vec{q})^{-T} V(\vec{t})^T (D(\vec{t},\vec{s})^{-1}.$$

This proves (4.3) □

Proof of equation (4.4): Rewrite (2.2) by isolating the Vandermonde pair as before to get

$$V(\vec{t})V(\vec{b})^{-1} = D(\vec{t},\vec{b})C(\vec{t},\vec{b})D'(\vec{b})^{-1},$$

rewrite to obtain

$$V(\vec{b})^{-T}V(\vec{t})^T = D'(\vec{b})^{-1}C(\vec{t},\vec{b})^T D(\vec{t},\vec{b}),$$

and substitute this last matrix equation and $C(\vec{t},\vec{b})^T = -C(\vec{b},\vec{t})$ into (4.3) to get

$$C(\vec{s},\vec{t}) = -C(\vec{s},\vec{q})D(\vec{q},\vec{s})D'(\vec{q})^{-1}C(\vec{t},\vec{q})^T D(\vec{t},\vec{q})D(\vec{t},\vec{s})^{-1}.$$

This proves (4.4) □

2.5 Some Algorithmic Aspects

The expressions (4.2) and (4.4) for $C(\vec{s},\vec{t})$ are Vandermonde-free and Hankel-free, but they enable us to transform the basis vectors $\vec{s}$ and $\vec{t}$ for $C(\vec{s},\vec{t})$ into the two pairs of basis vectors

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$\vec{s}$, $\vec{q}$ and $\vec{q}$, $\vec{t}$ for any choice of the vector $\vec{q} = (q_j)$, $q_j \neq s_i$, $q_j \neq t_k$, $i, j, k = 0, \ldots, n-1$. The associated Trummer's problem is reduced to

a) the evaluation of the diagonal matrices $D'(\vec{q})^{-1}$, $D(\vec{f}, \vec{g})$ and/or $D(\vec{f}, \vec{g})^{-1}$, for $(\vec{f}, \vec{g})$ denoting $(\vec{s}, \vec{t})$, $(\vec{q}, \vec{t})$, $(\vec{s}, \vec{q})$, $(\vec{q}, \vec{s})$, $(\vec{t}, \vec{q})$ and/or $(\vec{t}, \vec{s})$,

b) recursive multiplication of these matrices and the matrices $C(\vec{q}, \vec{t})$ and $C(\vec{s}, \vec{q})$ by vectors.

Let us next specify parts a) and b) in the next two paragraphs. To compute the matrices $D'(\vec{g})$, $D(\vec{f}, \vec{g})$ and $D(\vec{f}, \vec{g})^{-1}$ for given $(\vec{f}, \vec{g})$, we first compute the coefficients of the polynomial $p_{\vec{g}}(x) = \prod_{j=0}^{n-1}(x - g_j)$ and then $p_{\vec{g}}(f_i)$, and $p'_{\vec{g}}(g_i)$, at the points $i = 0, \ldots, n-1$. We compute the coefficients by the fan-in method, that is, we pairwise multiply at first the linear factors $x - g_j$ and then, recursively, the computed products (cf. [BP94], p. 25). The computation is numerically stable and uses $O(n \log^2 n)$ ops. Multipoint polynomial evaluation in $O(n \log^2 n)$ ops ([BP94], p. 26) is not stable numerically but the fast and numerically stable approximation techniques of [R88], [P95], [PLST93], [PZHY97] can be used instead. We simplify greatly the evaluation of the matrices $D(\vec{f}, \vec{g})$, $D(\vec{f}, \vec{g})^{-1}$ and $D'(\vec{q})$, where $\vec{f} = \vec{q}$ or $\vec{g} = \vec{q}$ if we may choose any vector $\vec{q} = (q_i)_{i=0}^{n-1}$. For instance, let us fill this vector with the scaled n-th roots of 1, so that

$$q_i = a w_n^i, \quad i = 0, 1, \ldots, n-1, \tag{5.1}$$

for a scalar a and $w_n = \exp(2\pi\sqrt{-1}/n)$. Then $p_{\vec{q}}(x) = \prod_{i=0}^{n-1}(x - a w_n^i) = x^n - a^n$, $p'_{\vec{q}}(x) = n x^{n-1}$, and the matrices $D(\vec{f}, \vec{q})$ and$D(\vec{q})$ can be immediately evaluated in $O(n \log n)$ flops. Furthermore, any polynomial $p(x)$ of degree n can be evaluated at the scaled n-th roots

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

of 1 in $O(n \log n)$ ops, by means of FFT. The multiplication of $C(\vec{q}, \vec{t})$ or $C(\vec{s}, \vec{q})$ by a vector is Trummer's problem, its solution can be simplified under an appropriate choice of the vector $\vec{q}$. In particular, even if we restrict $\vec{q}$ by (5.1), the scaling parameter a still controls fast convergence of the power series of the Multipole Algorithm. The above study can be extended to the expressions (4.1) and (4.3) for $C(\vec{s}, \vec{t})$. Each of them involves two Vandermonde matrices, but one of these matrices in each expression is defined by a vector $\vec{q}$ of our choice, and this enables us to yield simplification. In particular, for two given vectors $\vec{u} = (u_i)_{i=0}^{n-1}$ and $\vec{q} = (q_i)_{i=0}^{n-1}$, the vector $\vec{v} = V(\vec{q})^{-1}\vec{u}$ is the coefficient vector of the polynomial $v(x)$ that takes on the values u_k at the points q_k, $k = 0, \ldots, n-1$. For q_k being a scaled n-th roots of 1, as in (5.1), the computation of $\vec{v}$ takes $O(n \log n)$ ops due to the inverse FFT. Similar comments apply to the multiplication of the matrix $V(\vec{q})^{-T}$ by a vector. Effective parallelization is immediate at all steps of the computation.

Remark 2.5.1 *Trummer's problem frequently arises for Cauchy degenerate matrices* $C(\vec{s}) = (c_{i,j})$, $c_{i,i} = 0$, $c_{i,j} = \frac{1}{s_i - s_j}$ *for all pairs of distinct* i *and* j. *We have* $C(\vec{s}) = \frac{1}{h}\sum_{g=0}^{h-1} C(\vec{s}, \vec{s} + \epsilon w_h^g \vec{e}) + O(\epsilon^h)$ *as* $\epsilon \to 0$, *where* $\vec{e} = (1)_{j=0}^{n-1}$, $\vec{s} = (s_i)$,ϵ *is a scalar parameter. Indeed,* $\sum_{g=0}^{h-1} \frac{1}{s_i - s_j - \epsilon w_h^g} = \frac{1}{s_i - s_j} \sum_{l=0}^{\infty} \sum_{g=0}^{h-1} (\frac{\epsilon w_h^g}{s_i - j})^l = \frac{h}{s_i - s_j}(1 + O(\epsilon^h))$ *because* $\sum_{l=0}^{n-1} w_n^{gl} = 0$ *for* $g = 1, \ldots, n-1$.

Remark 2.5.2 *A distinct transformation of Trummer's problem may rely on substitution of (3.2) into (2.1) or (2.2). Here again, we may use map (3.3).*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 3

Divide-and-Conquer Algorithm

3.1 Basic Facts and a Cauchy-like Linear Solver

The transformations among the listed classes of structured matrices as a general means of improving the known algorithms for computations with such matrices were first proposed in [P90]. Along this line, the known practical Toeplitz and Toeplitz-like linear solvers were improved substantially in [GKO95] by reduction to Cauchy-like solvers, which enhanced the importance of the latter ones. A known explicit formula for the inverse of a Cauchy matrix (cf. e.g. [BP94], p.131) produces a good Cauchy solver but this is not enough in the application to Toeplitz linear solvers. We will follow [PZ,a] and [OP98] to show a distinct Cauchy-like linear solver, which extends the well known divide-and-conquer MBA algorithm, proposed in [M74], [M80], [BA80] as a Toeplitz-like linear solver. Every recursive divide-and-conquer step of the algorithm reduces to Trummer's problem, which relates this algorithm to the previous sections. We will start with definitions and basic facts [GO94a],

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[H95], [PZ,a], [OP98].

Definition 3.1.1 *[BP94], [C841], [GO94a], [H95]. For a field* $\mathbf{F}$ *and for vectors* $\vec{q} = (q_i)$, $\vec{t} = (t_j)$, $q_i \neq t_j$, $i, j = 0, \cdots, n-1$, *a matrix* $A \in \mathbf{F}^{n \times n}$ *is a* Cauchy *matrix (denoted by* $C(\mathbf{q}, \mathbf{t})$*) if* $A = (\frac{1}{q_i - t_j})_{i,j=0}^{n-1}$. *A is a* Cauchy-like *matrix if*

$$F_{[D(\vec{q}), D(\vec{t})]}(A) = D(\vec{q})A - AD(\vec{t}) = GH^T, \tag{1.1}$$

G, $H \in \mathbf{F}^{n \times r}$, *and* $r = O(1)$. *(Clearly,* $r = 1$ *for* $C(\mathbf{q}, \mathbf{t})$*.) The pair of matrices* (G, H^T) *of (1.1) is a* $[D(\vec{q}), D(\vec{t})]$-generator (or a scaling generator) *of length r for A and is denoted by* $s.g._r(A)$. *The minimum r allowing representation (1.1) is equal to* rank $(F_{[D(\vec{q})), D(\vec{t})]}(A))$ *and is called the* $[D(\vec{q}), D(\vec{t})]$-rank *(or the* scaling rank*) of A.*

Lemma 3.1.1 *Let* A, $\vec{q}$, $\vec{t}$, $G = [\vec{g}_1, \ldots, \vec{g}_r] = (\vec{u}_i^T)_{i=0}^{n-1}$, $H = [\vec{h}_1, \ldots, \vec{h}_r] = (\vec{v}_j^T)_{j=0}^{n-1}$ *be as in Definition 3.1.1, such that (1.1) holds. Then*

$$A = \sum_{m=1}^{r} \mathrm{diag}(\vec{g}_m) C(\vec{q}, \vec{t}) \mathrm{diag}(\vec{h}_m) = \left(\frac{\vec{u}_i^T \vec{v}_j}{q_i - t_j}\right)_{i,j=0}^{n-1}, \tag{1.2}$$

where $C(\vec{q}, \vec{t})$ *is a Cauchy matrix, and vice versa, (1.2) implies (1.1).*

Proof: (cf. [GO94]). The uniqueness of the equation follows from the fact that the numbers s_i, t_j are pairwise different (see e.g. [LT], p.411). Let matrix A be as given above. Then

$$F_{\{\mathrm{diag}(\vec{q}), \mathrm{diag}(\vec{t})\}}(A) = \sum_{m=1}^{r} \mathrm{diag}(\vec{g}_m) \cdot F_{\{\mathrm{diag}(\vec{q}), \mathrm{diag}(\vec{t})\}}(C(\vec{q}, \vec{t})) \cdot \mathrm{diag}(\vec{h}_m) =$$

$$\sum_{m=1}^{r} \mathrm{diag}(\vec{g}_m) \cdot \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix} \cdot \mathrm{diag}(\vec{h}_m) = \sum_{m=1}^{r} \vec{g}_m \cdot \vec{h}_m^T = G \cdot H^T.$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

□

It follows from (1.2) that (1.1) is satisfied by matrices A of the form $\left(\frac{\vec{u_i}^T\vec{v_j}}{q_i-t_j}\right)_{i,j=0}^{n-1}$, where $\vec{u_i}$ and $\vec{v_j}$ are r-dimensional vectors for $i, j = 0, 1, \ldots, n-1$. A Cauchy matrix $C(\vec{q}, \vec{t})$ and a Loewner matrix $\left(\frac{r_i-s_j}{q_i-t_j}\right)_{i,j=0}^{n-1}$ are two important special cases of Cauchy-like matrices; they have $[D(\vec{q}), D(\vec{t})]$-*ranks* 1 and 2, respectively.

Lemma 3.1.2 *(cf. [BP94], [Ger87]). Given an $n \times n$ Cauchy matrix A and an n-dimensional vector $\vec{v}$, the product $A\vec{v}$ can be computed in $C_{M_v}(n)$ ops (cf. (1.4)). If A is an $n \times n$ Cauchy-like matrix given with an $s.g._r(A)$, then the product $A\vec{v}$ can be computed in $(3n + C_{M_v}(n))r$ ops.*

Lemma 3.1.3 *$\vec{q_j} \in \mathbf{F}^{n\times 1}$, $j = 1, 2, 3$, where the vectors $\vec{q_1}$ and $\vec{q_3}$ share no components. Let $A_i \in \mathbf{F}^{n\times n}$, $F_{[D(\vec{q_i}),D(\vec{q}_{i+1})]}(A_i) = G_iH_i^T$, G_i, $H_i \in \mathbf{F}^{n\times r_i}$, $i = 1, 2$. Then the matrix $A = A_1A_2$ is a Cauchy-like matrix with $F_{[D(\vec{q_1}),D(\vec{q_3})]}(A) = GH^T$, $G = [G_1, A_1G_2]$, $H = [A_2^TH_1, H_2]$, G, $H \in \mathbf{F}^{n\times r}$, $r = r_1 + r_2$. Furthermore, $O(r_1r_2C_{M_v}(n))$ ops suffice to compute G and H.*

Lemma 3.1.4 *[H95]. Let $F_{[D(\vec{q}),D(\vec{t})]}(A) = GH^T$, $G = [\vec{g}_1, \cdots, \vec{g}_r] \in \mathbf{F}^{n\times r}$, $H = [\vec{h}_1, \cdots, \vec{h}_r] \in \mathbf{F}^{n\times r}$. Then $F_{[D(\vec{t}),D(\vec{q})]}(A^{-1}) = -UV^T$, where the matrices $U = [\vec{u}_1, \cdots, \vec{u}_r]$, $V = [\vec{v}_1, \cdots, \vec{v}_r]$ satisfy $AU = G$, $V^TA = H^T$.*

Corollary 3.1.1 *Under assumptions of Lemma 3.1.4, rank $F_{[D(\vec{t}),D(\vec{q})]}(A^{-1}) \le r$.*

Lemma 3.1.5 *Let an $n\times n$ Cauchy-like matrix A satisfy (1.1) and let $B_{I,J}$ be its $k\times d$ submatrix formed by its rows $i_1, \ldots, i_k$ and columns $j_1, \ldots, j_d$.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then $B_{I,J}$ has a $[D(\vec{q}_I), D(\vec{t}_J)]$-generator of a length at most r,

where $I = [i_1, \ldots, i_k]$, $J = [j_1, \ldots, j_d]$,

$D(\vec{q}_I) = \mathrm{diag}(q_{i_1}, \ldots, q_{i_k})$, $D(\vec{t}_J) = \mathrm{diag}(t_{j_1}, \ldots, t_{j_d})$.

Lemma 3.1.6 The matrices $A + B$ and $A - B$ have $[D(\vec{q}), D(\vec{t})]$-rank at most $r + r_1$ if A and B have $[D(\vec{q}), D(\vec{t})]$- ranks r and r_1, respectively.

Lemma 3.1.7 *(cf. [C841]).* An $n \times n$ Cauchy matrix $C(\vec{q}, \vec{t})$ is nonsingular if and only if all the $2n$ components of the vectors $\vec{q}$ and $\vec{t}$ are distinct. Every square submatrix of a nonsingular Cauchy matrix is nonsingular.

Fact 3.1.1 *(cf. Proposition A.6 of [P92b] or [BP94], Problem 2.2.11b).* Given an $s.g._{\tilde{r}}(A) = (G, H)$ and the scaling rank r of A, $r < \tilde{r} \le n$, one can compute an $s.g._r(A)$ by using $O(\tilde{r}^2 n)$ ops.

3.2 Recursive Factorization

3.2.1 The Case of a Strongly Nonsingular General Matrix

Definition 3.2.1 *We write $I = I_m \in \mathbf{F}^{m \times m}$ for the $m \times m$ identity matrix,* 0 *for a null matrix of appropriate size, W^T and W^H for the transpose and the Hermitian transpose of a matrix or a vector W, respectively.*

Definition 3.2.2 *$W^{(k)} \in \mathbf{F}^{k \times k}$ is the $k \times k$ leading principal (northwestern) submatrix of an $m \times n$ matrix W, $k = 1, \ldots, \min(m, n)$. A matrix W of* rank ρ *has generic rank*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

profile if its $k \times k$ leading principal submatrices $W^{(k)}$ are nonsingular for $k = 1, \dots, \rho$. A matrix is strongly nonsingular if its both nonsingular and has generic rank profile.

$$X = \begin{pmatrix} I & O \\ X_{21}X_{11}^{-1} & I \end{pmatrix} \begin{pmatrix} X_{11} & O \\ O & S \end{pmatrix} \begin{pmatrix} I & X_{11}^{-1}X_{12} \\ O & I \end{pmatrix}, \tag{2.1}$$

$$X^{-1} = \begin{pmatrix} I & -X_{11}^{-1}X_{12} \\ O & I \end{pmatrix} \begin{pmatrix} X_{11}^{-1} & O \\ O & S^{-1} \end{pmatrix} \begin{pmatrix} I & O \\ -X_{21}X_{11}^{-1} & I \end{pmatrix}, \tag{2.2}$$

where X is an $n \times n$ strongly nonsingular matrix,

$$X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}, \qquad S = X_{22} - X_{21}X_{11}^{-1}X_{12}, \tag{2.3}$$

By expanding (2.2), we obtain

$$X^{-1} = \begin{pmatrix} X_{11}^{-1} + X_{11}^{-1}X_{12}S^{-1}X_{21}X_{11}^{-1} & -X_{11}^{-1}X_{12}S^{-1} \\ -S^{-1}X_{21}X_{11}^{-1} & S^{-1} \end{pmatrix} \tag{2.4}$$

X_{11} is a $k \times k$ matrix, and $S = S(X_{11}, X)$ is called the *Schur complement of X_{11} in X*. (2.1) represents block Gauss-Jordan elimination applied to the 2×2 block matrix X of (2.3). If the matrix X is strongly nonsingular, then the matrix S of (2.3) can be obtained in $n - k$ steps of Gaussian elimination.

Proposition 3.2.1 *([BP94], Exercise 4 of ch. 2, page 212): If X is strongly nonsingular, so are X_{11} and S.*

Proposition 3.2.2 *(cf. [BP94], Proposition 2.2.3). Let X be an $n \times n$ strongly nonsingular matrix and let S be defined by (2.3). Let X_1 be a leading principal submatrix of S and let S_1 denote the Schur complement of X_1 in S. Then S^{-1} and S_1^{-1} form the respective southeastern blocks of X^{-1}.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proposition 3.2.3 *If (2.1) holds, then det* $X = (\det X_{11}) \det S$.

Proposition 3.2.4 *[BP94], [GL]. Suppose that the Schur complements* $S = S(X^{(k)}, X)$ *and* $S_1 = S(S^{(h)}, S)$ *are defined, that is, the matrices* $X^{(k)}$ *and* $S^{(h)}$ *are nonsingular. Then* $S_1 = S(X^{(k+h)}, X)$.

Due to Propostion 3.2.4, we may extend factorization (2.1) from X to X_{11} and S and then recursively continue such a *descending process* until we arrive at 1×1 matrices (compare [St69], [M74], [M80], [BA80] [AHU]). In actual computation, we apply *lifting process* that begins with the inversion of the 1×1 matrix $X^{(1)}$. Then we compute and invert its 1×1 Schur complement S_1 in the 2×2 matrix $X^{(2)}$ (this defines the factorization of $X^{(2)}$ and of its inverse), compute the inverse of $X^{(2)}$ and its 2×2 Schur complement S_2 in the 4×4 matrix $X^{(4)}$, and so on. In other words, we recursively proceed *bottom up*, that is, we invert 1×1 matrices and, other than that use only matrix multiplications and subtractions to compute all matrices specified in the recursive descending process, until we finally arrive at X^{-1}. The algorithm emulates Gaussian elimination steps except that it combines their scalar multiplications and subtractions into similar operations with matrix blocks. The entire computation will be called the CRF (or *complete recursive factorization*) of X. In the *balanced* CRFs, X_{11} of (2.1) is a $\lfloor \frac{n}{2} \rfloor \times \lfloor \frac{n}{2} \rfloor$ submatrix of X, and similar balancing is maintained in all subsequent recursive steps. The balanced CRF has depth at most $d = \lceil \log_2 n \rceil$.

Algorithm 3.2.1 *Recursive triangular factorization and inversion.*

Input: *a strongly nonsingular* $n \times n$ *matrix* X.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Output: *balanced CRF of X, including the matrix X^{-1}.*

Computations:

1. Apply Algorithm 3.2.1 to the matrix X_{11} (replacing X as its input) to compute the balanced CRF of X_{11} (including X_{11}^{-1}).

2. Compute the Schur complement $S = X_{22} - X_{21}X_{11}^{-1}X_{12}$.

3. Apply Algorithm 3.2.1 to the matrix S (replacing X as its input) to compute the balanced CRF of S (including S^{-1}).

4. Compute X^{-1} from (2.2).

As a by-product, Algorithm 3.2.1 may immediately compute the vector $\vec{y} = X^{-1}\vec{b}$ of the solution to a linear system $X\vec{y} = \vec{b}$ for a given vector $\vec{b}$. If we also seek $\det X$, then it suffices to add the request for computing $\det X_{11}$, $\det S$, and $\det X$ (see Lemma 3.2.3) at stages 1, 3, and 4, respectively.

It is well known [St69], [BP94], p.99, that the complexity of these computations (in terms of the number of ops involved) satisfies

$$Rf(n) = O(\sum_{i=1}^{h} 2^i M(n/2^i)), \tag{2.5}$$

where w.l.o.g. we assume $n = 2^h$ for an integer h, and $M(n)$ is the complexity of $n \times n$ matrix multiplication. Theoretically, for general matrices, $M(n) = O(n^\beta)$, for $2.8 < \beta \leq 3$, and we deduce from (2.5) that

$$RF(n) = O(M(n)) \;\; if \;\; M(n) \geq n^{1+d}, d < 0, \tag{2.6}$$

$$RF(n) = O(M(n)\log n) \;\; if \;\; M(n) = O(n \log^c n), \tag{2.7}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

for a constant c.

Now, let X have generic rank profile. In this case we apply **generalized Algorithm 3.2.1**, which includes a counter for the number of the inversions of 1×1 matrices involved (that is, for the number of divisions). If division by 0 occurs, the computations stop, and at this point, we have $\rho = \text{rank } X$ in the counter and a CRF of $X^{(\rho)}$ available, which immediately gives us $(X^{(\rho)})^{-1}$.

Let us also show a simple extension (see e.g. [BP94], p. 110). Recall (2.1) with $X_{11} = X^{(\rho)}$ and write

$$F = \begin{pmatrix} I_\rho & -X_{11}^{-1}X_{12} \\ O & I_{n-\rho} \end{pmatrix}, N = F\begin{pmatrix} 0 \\ I_{n-\rho} \end{pmatrix}. \tag{2.8}$$

Then

$$XF = \begin{pmatrix} X_{11} & 0 \\ X_{21} & 0 \end{pmatrix}, \tag{2.9}$$

the columns of N (which are the last $n-\rho$ columns of the matrix F) form a basis for the null space of X, and the substitution $\vec{y} = F\vec{z}$ reduces the solution of a linear system $X\vec{y} = \vec{b}$ (or the determination of its inconsistency) to the case of the system $(XF)\vec{z} = \vec{b}$, for which the problem is simple because we have (2.9) and already know X_{11}^{-1}.

Definition 3.2.3 *A full output set of generalized Algorithm 3.2.1 consists of the set of the matrices of a CRF of a largest nonsingular submatrix of X, complemented by the rank ρ of X, a basis for the null space of X, a solution $\vec{y}$ to the linear system $X\vec{y} = \vec{b}$ for a given vector b (or the determination of its inconsistency), and if $\rho = n$, then also by X^{-1} and* $\det X$. *Without the CRF (but including a largest nonsingular submatrix), this is a partial output set.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

The next extension of (2.6), (2.7) to the computation of a full output set is immediately verified:

Theorem 3.2.1 *For an $n \times n$ matrix X having generic rank profile, generalized Algorithm 3.2.1 computes its full output set at the cost bounded according to (2.6), (2.7).*

One may extend the bound (2.6) of Theorem 3.2.1 to an arbitrary matrix X by using pivoting [IMH82], which only adds $O(n^2)$ comparison and some permutations of matrix rows and columns, so that the overall cost bound is still dominated by $O(M(n))$ under (2.6).

For a nonsingular real or complex matrix X, one may apply symmetrization instead of pivoting in order to compute (within the cost bounds (2.6), (2.7)) the balanced CRFs of the strongly nonsingular matrices $X^H X$ and/or XX^H (at the price of possible squaring the condition number of X), and then obtain X^{-1} as $(X^H X)^{-1} X^H$ or as $X^H (XX^H)^{-1}$ and $(\det X)^2 = \det(X^H X) = \det(XX^H)$. Symmetrization has no effect over the fields of positive characteristic as well as in the case of a singular input matrix X.

3.2.2 The Case of a Strongly Nonsingular Cauchy-like Matrix

Suppose that we are given a Cauchy-like input matrix X. Then, applying Algorithm 3.2.1, we will achieve a dramatic decrease of computer time and memory space involved by operating with the short generators of the matrices involved rather than with all their entries. Hereafter, we will assume for simplicity that $n = 2^d$ is an integer power of 2. We write

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$\vec{q}=(q_i)_{i=0}^{n-1}$, $\vec{q}^{\,(1)}=(q_i)_{i=0}^{\frac{n}{2}-1}$, $\vec{q}^{\,(2)}=(q_i)_{i=\frac{n}{2}}^{n-1}$, $\vec{t}=(t_i)_{i=0}^{n-1}$, $\vec{t}^{\,(1)}=(t_i)_{i=0}^{\frac{n}{2}-1}$, $\vec{t}^{\,(2)}=(t_i)_{i=\frac{n}{2}}^{n-1}$.

We will start with some auxiliary results.

Lemma 3.2.1 *[GO94], [OP98]. Let X be a Cauchy-like matrix of Lemma 3.1.1, partitioned into blocks according to (2.3). Let (G_0, H_0), (G_0, H_1), (G_1, H_0), (G_1, H_1) and (G_S, H_S) denote the five induced scaling generators of the blocks X_{11}, X_{12}, X_{21}, X_{22} of X and of the Schur complement S of (2.3), respectively. Then $G_S = G_1 - X_{21}X_{11}^{-1}G_0$, $H_S^T = H_1^T - H_0^T X_{11}^{-1}X_{12}$.*

Lemma 3.2.2 *Let X be an $n \times n$ strongly nonsingular Cauchy-like matrix with $F_{[D(\vec{q}),D(\vec{t})]}(X) = GH^T$, for $n \times r$ matrices G, H. Let X, X_{11}, X_{12}, X_{21}, X_{22}, and S satisfy (2.3). Then*

$$rank F_{[D(\vec{t}),\ D(\vec{q})]}(X^{-1}) \le r, \qquad rank F_{[D(\vec{t}^{\,(1)}),\ D(\vec{q}^{\,(1)})]}(X_{11}^{-1}) \le r, \tag{2.10}$$

$$rank F_{[D(\vec{q}^{\,(2)}),\ D(\vec{t}^{\,(2)})]}(S) \le r, \tag{2.11}$$

$$rank F_{[D(\vec{t}^{\,(2)}),\ D(\vec{q}^{\,(2)})]}(S^{-1}) \le r, \tag{2.12}$$

$$rank F_{[D(\vec{q}^{\,(1)}),\ D(\vec{t}^{\,(2)})]}(X_{12}) \le r, \qquad rank F_{[D(\vec{q}^{\,(2)}),\ D(\vec{t}^{\,(1)})]}(X_{21}) \le r. \tag{2.13}$$

Proof. Deduce (2.11) and (2.13) from Lemmas 3.1.5 and 3.2.1. Apply Corollary 3.1.1, obtain (2.10) and (2.12).

Fact 3.2.1 *(cf. Proposition A.6 of [P92b], [P93a]). Given an $s.g._{r^*}(X) = (G, H)$ and the scaling rank r of X, $r < r^* \le n$, one can compute an $s.g._r(X)$ by using $O((r^*)^2 n)$ ops.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Now we are ready to present the computational complexity estimates.

Theorem 3.2.2 *Let X denote an $n \times n$ strongly nonsingular Cauchy-like matrix with its F-generator of a length r for the operator $F = F_{[D(\vec{q}),D(\vec{t})]}$. Then the respective F-generators of all the matrices encountered in the balanced CRF of X (including an $s.g._r(X^{-1})$) and $\det X$ can be computed in $O(r^2T(n)\log n) = O(nr^2\log^3 n)$ ops (for $T(n)$ of Lemma 3.1.2) and can be stored by using $O(nr\log n)$ words of storage space.*

Proof.[BP94], [P2000], [PZ,a], [PACPS98], [PACPZ99] Let us apply the fast version of Algorithm 3.2.1 to the matrix X of Theorem 3.2.2, that is, instead of slower computations with matrices, let us perform faster computations with their short scaling generators. Let $\phi_r(n)$ops be involved in computing the balanced CRF of X (including the computation of an $s.g._r(X^{-1})$). Furthermore, let $\sigma_r(n)$ ops be used for computing an $s.g._r(S)$ from given $s.g._r(X_{11}^{-1})$, $s.g._r(X_{12})$, $s.g._r(-X_{21})$, and $s.g._r(X_{22})$ (cf. Lemma 3.2.1, and let $\mu_r(n)$ ops be required for computing an $s.g._r(X^{-1})$ from given $s.g._r(X_{11}^{-1})$, $s.g._r(X_{12})$, $s.g._r(X_{21})$, and $s.g._r(S^{-1})$ (cf. (2.2)). This is summarized below.

Input	$s.g._r(X)$	$s.g._r(X_{11}^{-1}), s.g._r(X_{12})$, $s.g._r(-X_{21}), s.g._r(X_{22})$	$s.g._r(X_{11}^{-1}), s.g._r(X_{12})$, $s.g._r(X_{21}), s.g._r(S)$
Output	CRF of X	$s.g._r(S)$	$s.g._r(X^{-1})$
ops	$\phi_r(n)$	$\sigma_r(n)$	$\mu_r(n)$

Let $\phi_r(k)$, $\sigma_r(k)$, and $\mu_r(k)$ denote the similar estimates a strongly nonsingular $k \times k$ input matrix W given with an $s.g._r(W)$. Then from Algorithm 3.2.1,

$$\phi_r(n) \le 2\phi_r(\frac{n}{2}) + \sigma_r(n) + \mu_r(n). \tag{2.14}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Now we apply (2.2), Lemmas 3.1.3, 3.1.5, 3.2.1, and 3.2.2 and deduce that

$$\sigma_r(n) = O(r^2 T(n)), \quad \mu_r(n) = O(r^2 T(n)). \tag{2.15}$$

Substitute (2.15) into (2.14), recursively extend (2.14), and deduce that

$$\phi_r(n) = O(r^2 T(n)),$$

which gives us the arithmetic time bound of Theorem 3.2.2. The storage space bound follows similarly when we inspect Algorithm 3.2.1 applied to the matrix A and apply Lemma 3.1.3, Lemma 3.1.5, Proposition 3.2.1, Lemma 3.2.1 and Lemma 3.2.2.

Remark 3.2.1 *In [PACPS98], [OP98], Lemma 3.2.1, was extended to the case of Hankel-like matrices H associated with operators $ZH - HZ^T$, Z being a shift matrix. This enabled practical improvement of the MBA Hankel/Toeplitz linear solver.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 4

Ensuring Strong Nonsingularity

4.1 Strong Nonsingularity by Preconditioning

To extend our algorithm to any nonsingular matrix A, we will seek a strongly nonsingular matrix X, such that the matrix AX is strongly nonsingular. Then we may apply our machinery to the matrices X and AX or XA, compute $(AX)^{-1} = X^{-1}A^{-1}$ or $(XA)^{-1} = A^{-1}X^{-1}$, $\det(AX) = \det(XA)$, and $\det X$, and then $A^{-1} = X(AX)^{-1} = (XA)^{-1}X$ and $\det A = \det(AX)/\det X$. If A is singular, the same algorithm will involve a division by 0 and thus will show us that $\det A = 0$. In [PZ,a] the algorithm is extended to computing rank A and solving consistent singular Cauchy-like linear systems.

Computing with reals we may set $X = A^T$, A^T denoting the transpose of A. Indeed, the matrix $XA = A^TA$ is positive definite and consequently strongly nonsingular provided that A is nonsingular. Moreover, in this case the condition numbers of X_{11}, S of (2.3), and all similar matrices of the CRF do not exceed the condition number of the input matrix

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

(cf. [BP94], Fact 2.1.4 and page 237). As a by-product, we immediately arrive at a least-squares (normal equations) solution $(A^TA)^{-1}A^T\vec{b}$ to a Cauchy-like linear system $A\vec{x} = \vec{b}$ for a Cauchy-like $m \times n$ rectangular matrix A having full rank n, $n \leq m$. For the operators $F_{[D(\vec{t}),D(\vec{t})]}$ and $F_{[D(\vec{q}),D(\vec{q})]}$ associated with the matrices $W = A^TA$ and $U = AA^T$, respectively, the assumption $q_i \neq t_j$ of Definition 3.1.1 is not extended, but we will operate with W represented as the product $C^{-1}(\vec{q},\vec{t})Y$, where $Y = C(\vec{q},\vec{t})W$ and $C^{-1}(\vec{q},\vec{t})$ are Cauchy-like matrices, and similarly for U. For a nonsingular real or complex matrix A, the matrices A^TA and AA^T are strongly nonsingular. By applying Algorithm 3.2.1 to these matrices, we may extend Theorem 3.2.2 to any nonsingular matrix A [PZ,a], [PACPS98]. For computations in finite fields such a symmetrization does not work, but we will next yield similar results over any field $\mathbf{F}$ by using random parameters always sampled from a fixed finite set $\mathbf{S}$ (in $\mathbf{F}$ or in its extension) independently of each other and under the uniform probability distribution on $\mathbf{S}$. We will keep using definitions of sections 2 and will rely on the following lemma.

Lemma 4.1.1 *[DL78] (cf. also [Z79], [S80]). Let $p(\vec{x}) = p(x_1, x_2, \ldots, x_m)$ be a non-zero m-variate polynomial of a total degree d. Let* $\mathbf{S}$ *be a finite set of cardinality* $|\mathbf{S}|$ *in the domain of the definition of $p(\vec{x})$, let the random values $x^*, \cdots, x^*_m$ be sampled from* $\mathbf{S}$, *that is, chosen from* $\mathbf{S}$ *independently of each other under the uniform probability distribution on* $\mathbf{S}$, *and let $\vec{x}^* = (x^*_1, x^*_2, \cdots, x^*_m)$. Then probability $(p(\vec{x}^*) = 0) < d/|\mathbf{S}|$.*

Theorem 4.1.1 *Let A be an $n \times n$ nonsingular matrix satisfying equation (1.2). Let*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

X be a matrix satisfying $X = YC(\vec{q},\vec{s})$, where

$$Y = \sum_{m=1}^{r} D(\vec{g}_m^*)C(\vec{t},\vec{q})D(\vec{h}_m^*) \tag{1.1}$$

$C(\vec{q},\vec{s}) = (\frac{1}{q_i - s_j})_{i,j=0}^{n-1}$ is a fixed nonsingular Cauchy matrix, $\vec{q} \in \mathbf{F}^{n\times 1}$, $\vec{t} \in \mathbf{F}^{n\times 1}$, $\vec{s} \in \mathbf{F}^{n\times 1}$, $\vec{q}$ and $\vec{t}$ are as in Lemma 3.1.4, $q_i \neq s_j$, $s_i \neq t_j$ for all i and j, $\vec{g}_m^* \in \mathbf{F}^{n\times 1}$, $\vec{h}_m^* \in \mathbf{F}^{n\times 1}$, $m = 1, \cdots, r$, and the $2nr$ components of the $2r$ latter vectors are random values from a fixed finite set $\mathbf{S}$. Then AX has F-rank at most $2r+1$ and, with a probability at least $1 - n(n+1)/|\mathbf{S}|$, is a strongly nonsingular matrix.

Proof First consider matrix Y of (1.1), where the random vectors $\vec{g}_m^*$ and $\vec{h}_m^*$ are replaced by generic vectors whose components are indeterminates. Recall that the $F_{[D(\vec{t}),D(\vec{q})]}$-rank of A^{-1} is at most r, due to Lemma 3.1.4. Therefore, there exists an assignment of values to the components of the vectors $\vec{g}_m^*$, $\vec{h}_m^*$, for which we have $AY = I$, and then the matrix $AX = C(\vec{q},\vec{s})$ is strongly nonsingular (cf. Lemma 3.1.7). On the other hand, the determinants of the $k \times k$ leading principal submatrices $(AX)_k$ of AX are polynomials of degrees at most $2k$ in the coordinates of $\vec{g}_m^*$, $\vec{h}_m^*$. Since $AX = C(\vec{q},\vec{s})$ for a particular assignment, these polynomials are not identically 0 if the components are indeterminates. Therefore, by Lemma 4.1.1, we obtain *probability*$(\det(AX)_k \neq 0, k = 1, \cdots, n) \geq \prod_{k=1}^{n} \frac{1-2k}{|\mathbf{S}|} \geq \frac{1-n(n+1)}{|\mathbf{S}|}$.

□

Corollary 4.1.1 . *Let an $n \times n$ nonsingular Cauchy-like matrix A be given with its F-generator of length r for the operator $F = F_{[D(\vec{q}),D(\vec{t})]}$. Then an $F_{[D(\vec{t}),D(\vec{q})]}$-generator of length at most r for A^{-1} can be computed by means of a randomized algorithm*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

using $2nr$ random parameters (sampled from the set $\mathbf{S}$*) and* $O(nr^2 \log^3 n)$ *ops and failing with a probability at most* $\frac{n(n+1)}{|\mathbf{S}|}$.

Proof: Let us define X as above. By Theorem 4.1.1, the Cauchy-like matrix AX is strongly nonsingular with a probability at least $1-n(n+1)/|\mathbf{S}|$, and then, by Theorem 3.2.2, we may compute the matrices $(AX)^{-1}$ and $A^{-1} = X(AX)^{-1}$ by using a total of $O(C_{M_v}(n)r^2 \log n)$ ops. Due to Lemma 4.1.1, we also obtain the desired bounds on the number of random parameters used and on the failure probabilities. Finally, we will decrease the length of the computed F-generator of A^{-1} by applying Fact 3.1.1.

Corollary 4.1.2 *[PACPZ99], [P2000], [PZ,a].* $\det A$ *and* $\rho = \mathrm{rank}\, A$ *can be computed by using* $2nr$ *random parameters and* $O(C_{M_v}(n)r^2 \log n)$ *ops for a matrix* A *of (1.1), (1.2). If* $C(\vec{t},\vec{q})$ *is a nonsingular Cauchy matrix, then (by Lemmas 3.1.7 and 4.1.1) the matrix* X *is strongly nonsingular, with a probability at least* $1 - n(n+1)/|\mathbf{S}|$*, and (by Theorem 3.2.2)* $\det\,(AX)$*,* $\det\, X$*, and* $\det\, A = \frac{\det\,(AX)}{\det\, X}$ *can be computed at the randomized cost* $O(C_{M_v}(n)r^2 \log n)$*. Furthermore, with a probability at least* $\rho(\rho+1)/|\mathbf{S}|$*,* $\rho \times \rho$ *is the maximum size of a nonsingular leading principal submatrix of* AX *for the matrix* X *of Theorem 4.1.1. Such a size is computed as a by-product of application of Algorithm 3.2.1 to* AX*.*

4.2 Fast Cauchy-like Computations - Singular Case

Studying the solution of a singular Cauchy-like linear system, we will use the next result and definition.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Lemma 4.2.1 *[K95]. Let A be an $n \times n$ matrix of* rank *ρ with entries from a fixed field $\mathbf{F}$ and suppose that the $\rho \times \rho$ leading principal submatrix A_ρ is nonsingular. Then for a vector $\vec{y}$ with coordinates from the field $\mathbf{F}$ the vector*

$$\vec{x} = \begin{pmatrix} A_\rho^{-1}\vec{b}\,' \\ \vec{0} \end{pmatrix} - \vec{y}$$

is a solution to $A\vec{x} = \vec{b}$, where the vector $\vec{b}\,'$ consists of the first ρ coordinates of $\vec{b} + A\vec{y}$, and $\vec{0}$ denotes the null vector of dimension $n - \rho$.

Definition 4.2.1 *Let A_i be the $i \times i$ leading principal submatrix of A, where $1 \le i \le n$. We say that A has* generic rank profile *if A_j is nonsingular for all j, $1 \le j \le \operatorname{rank}(A)$.*

The next theorem extends the known results from the Toeplitz-like (cf. [BP94], p. 206, or [KS91]) to the Cauchy-like case and can be an alternative to Theorem 4.1.1 where the input matrix is nonsingular (see Remark 4.2.1).

Theorem 4.2.1 *[P2000]. For an $n \times n$ Cauchy-like matrix A of rank ρ represented by an s.g.-$_r(A)$ and satisfying (3.1.1) and (3.1.2) consider the matrix product $\overline{A} = LAM$, where L and M are also Cauchy-like matrices with scaling generators of length 1. Assume the following relations:*

$$F_{[D(\vec{s}),\ D(\vec{q})]}(L) = YZ^T, F_{[D(\vec{t}),\ D(\vec{p})]}(M) = XW^T,$$

$$Y^T = [y_1, \cdots, y_n] \in \mathbf{F}^n, Z^T = [z_1, \cdots, z_n] \in \mathbf{F}^n,$$

$$X^T = [x_1, \cdots, x_n] \in \mathbf{F}^n, W^T = [w_1, \cdots, w_n] \in \mathbf{F}^n,$$

$$L = \left(\frac{y_{i+1}z_{j+1}}{s_i - q_j}\right)_{i,j=0}^{n-1}, M = \left(\frac{x_{i+1}w_{j+1}}{t_i - p_j}\right)_{i,j=0}^{n-1},$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

where the entries of the matrices Y, Z, X, W are random samples from a fixed finite subset $\mathbf{S}$ *of the field* $\mathbf{F}$ *and where* $\mathbf{S}$ *does not contain* 0. *Let* s_i, q_j, p_k *be all pairwise distinct for* $i, j, k = 0, \cdots, n-1$. *Then*

(1) L *and* M *are strongly nonsingular matrices and*

(2) $\overline{A}$ *has generic rank profile with a probability at most* $1 - \frac{2\rho(\rho+1)}{|\mathbf{S}|}$.

Proof: Part (1) follows from (1.2) and Lemma 3.1.7 since $\mathbf{S}$ does not contain 0. Let us prove part (2). For an $n \times n$ matrix D, denote by $D_{I,J}$ the determinant of the submatrix of D formed by removing from D all rows not contained in the set I and all columns not contained in the set J. First, let Y, Z, X, and W be generic matrices. For $I = [1, 2, \cdots, i]$, $J = [j_1, j_2, \cdots, j_i]$, $K = [k_1, k_2, \cdots, k_i]$, $i = 1, 2, \cdots, \rho$, we have from the Cauchy-Binet formula that

$$\overline{A}_{I,I} = \sum_J \sum_K L_{I,J} A_{J,K} M_{K,I}.$$

Let us prove that

$$\overline{A}_{I,I} \neq 0 \ for \ i = 1,\ 2,\ \cdots,\ \rho\,. \tag{2.1}$$

Observe that, for a fixed pair of $J = [j_1, j_2, \cdots, j_i]$ and $K = [k_1, k_2, \cdots, k_i]$, the determinant $L_{I,J}$ has the unique term

$$a y_1 y_2 \cdots y_i z_{j_1} \cdots z_{j_i},$$

where $a \neq 0$ is a constant. Likewise, $M_{K,I}$ has the unique term

$$b x_{k_1} \cdots x_{k_i} w_1 \cdots w_i,$$

where $b \neq 0$ is a constant. Therefore, $\overline{A}_{I,I} \neq 0$ provided that there exists a pair J, K such that $A_{J,K} \neq 0$. This is true for all $i \leq \rho$, since A has rank ρ, and we arrive at (2.1).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Now, we observe that $\overline{A}_{I,I}$ is a polynomial of degree at most $4i$ in the variables y_m, z_m, x_m, w_m, $m = 1, \ldots, n$. Under the random choice of their values, we apply Lemma 4.1.1 and obtain that $probability(\overline{A}_{I,I} \neq 0, i = 1, \cdots, \rho) \geq \prod_{i=1}^{\rho}(1 - 4i/|\mathbf{S}|) \geq 1 - 2\rho(\rho+1)/|\mathbf{S}|$. This proves part (2) of Theorem 4.2.1. □

Remark 4.2.1 *If the input Cauchy-like matrix is nonsingular, we may apply Theorem 4.2.1, as an alternative to Theorem 4.1.1. Application of Theorem 4.2.1, rather than Theorem 4.1.1, requires by factor $2/r$ fewer random parameters ($4n$ versus $2nr$) and involves scaling generators of roughly half length ($r+2$ versus $2r+1$), at the small price of doubling the probability of errors ($2n(n+1)/|\mathbf{S}|$ versus $n(n+1)/|\mathbf{S}|$).*

To prove Theorem 4.2.1, we devised a simple algorithm that, for an $n \times n$ Cauchy-like matrix A of rank ρ given with an $s.g._r(A)$, computes a random pair $s.g._1(L)$ and $s.g._1(M)$, where L and M are $n \times n$ matrices such that, with a probability at least $1 - 2\rho(\rho+1)/|\mathbf{S}|$, the matrix $\overline{A} = LAM$ has generic rank profile. Furthermore, based on Lemma 3.1.3, we computed $s.g._{r+2}(\overline{A})$ by using $O(r^2 C_{M_v}(n))$ ops (cf. (1.4)). Now, we assume that we have been already given $s.g._1(L)$, $s.g._1(M)$, and $s.g._{r+2}(\overline{A})$ for a pair of nonsingular matrices L and M and an $n \times n$ matrix $\overline{A} = LAM$ having generic rank profile and propose the following algorithm, using $O(r^2 C_{M_v}(n) \log n)$ ops (cf. (1.3)).

Algorithm 4.2.1 *(cf. [PZ,a], [PACPS98]) Computation of the largest nonsingular leading principal inverse.*

Input: vectors $\vec{q} = (q_i)_{i=0}^{n-1}$, $\vec{t} = (t_j)_{j=0}^{n-1}$, $q_i \neq t_j$, $i, j = 0, 1, \cdots, n-1$, and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$\vec{g}_1, \cdots, \vec{g}_{r+2}, \vec{h}_1, \cdots, \vec{h}_{r+2}$ such that the next matrix has generic rank profile:

$$\overline{A} = \sum_{m=1}^{r+2} D(\vec{g}_m) C(\vec{q}, \vec{t}) D(\vec{h}_m).$$

Output: An integer $\rho \leq n$ and vectors $\vec{u}_1, \cdots, \vec{u}_{\overline{r}}$, $\vec{v}_1, \cdots, \vec{v}_{\overline{r}}$, $\vec{u}_m, \vec{v}_m \in \mathbf{F}^{n \times 1}$, $m = 1, 2, \cdots, \overline{r}$, $\overline{r} \leq r + 2$, such that $\rho = \text{rank}(\overline{A})$ and

$$\overline{A}_\rho^{-1} = \sum_{m=1}^{\overline{r}} D(\vec{u}_m) C(\vec{t}, \vec{q}) D(\vec{v}_m).$$

1. Represent $\overline{A}$ as $\overline{A} = \begin{pmatrix} \overline{B} & \overline{C} \\ \overline{E} & \overline{J} \end{pmatrix}$, cf. (3.3), where $k = \lceil \frac{n}{2} \rceil$, and the $k \times k$ submatrix $\overline{B}$ of $\overline{A}$ is singular if and only if $k > \rho$ (since $\overline{A}$ has generic rank profile). Apply Algorithm 4.2.1 recursively to the input matrix $\overline{B}$ replacing $\overline{A}$. (Note that we are given an $s.g._r(\overline{B})$.) If $\rho \geq k$, the output of this stage is the desired output of the algorithm. Otherwise, the matrix $\overline{B}$ is nonsingular, and then we obtain $s.g._r(\overline{B}^{-1})$.

2. Apply Algorithm 3.2.1 to compute an $s.g._r(\overline{S})$ for the matrix $\overline{S} = \overline{J} - \overline{E}\,\overline{B}^{-1}\overline{C}$.

3. Apply the algorithm recursively to the Cauchy-like input matrix $\overline{S}$, replacing $\overline{A}$. Output $\rho = \text{rank}(\overline{A}) = k + \text{rank}(\overline{S})$.

4. By using the definitions and the results of section 2, compute an $s.g._{2r+4}(\overline{A}_\rho^{-1})$ (see further comments below).

5. Apply Fact 3.1.1, to compute and output $s.g._{r+2}(\overline{A}_\rho^{-1})$.

Let us specify stage 4. Consider the $\rho \times \rho$ leading principal submatrix, $\overline{A}_\rho = \begin{pmatrix} \overline{B} & \overline{G} \\ \overline{D} & \overline{R} \end{pmatrix}$, $\overline{G}$, $\overline{D}^T \in \mathbf{F}^{k \times (\rho - k)}$, $\overline{R} \in \mathbf{F}^{(\rho-k)\times(\rho-k)}$. Write $\hat{S} = \overline{R} - \overline{D}\,\overline{B}^{-1}\overline{G}$. Note that at the preceding stages we have computed $s.g._{r+2}(\overline{G})$, $s.g._{r+2}(\overline{D})$, $s.g._{r+2}(\overline{B}^{-1})$, $s.g._{r+2}(\overline{D}\overline{B}^{-1})$, $s.g._{r+2}(\overline{B}^{-1}\overline{G})$,

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

and $s.g._{r+2}(\hat{S}^{-1})$ (cf. Theorem 3.2.2). Represent $\overline{A}_\rho^{-1}$ as follows:

$$\overline{A}_\rho^{-1} = \begin{pmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & \overline{S}^{-1} \end{pmatrix},$$

where $B_{1,2} = -\overline{B}^{-1}\overline{G}\,\overline{S}^{-1}$, $B_{2,1} = -\overline{S}^{-1}\overline{D}\,\overline{B}^{-1}$, $B_{1,1} = \overline{B}^{-1} - B_{1,2}\overline{D}\,\overline{B}^{-1}$ (cf. (2.4). Due to Lemma 3.1.5 and Corollary 3.1.1, the matrices $B_{1,1}$, $B_{1,2}$, $B_{2,1}$, and $\overline{S}^{-1}$ have scaling rank at most $r+2$, and we may apply Algorithm 3.2.1 and the results of chapter 3 to compute the respective short scaling generators of these matrices. Let us specify the operators defining these generators. Write

$$\vec{q}^{\,(1)} = (q_i)_{i=0}^{k-1}, \vec{q}^{\,(2)} = (q_i)_{i=k}^{\rho-1}, \vec{t}^{\,(1)} = (t_i)_{i=0}^{k-1}, \vec{t}^{\,(2)} = (t_i)_{i=0}^{\rho-1},$$

$$\vec{q}^{\,(0)} = \begin{pmatrix} \vec{q}^{\,(1)} \\ \vec{q}^{\,(2)} \end{pmatrix} \text{ and } \vec{t}^{\,(0)} = \begin{pmatrix} \vec{t}^{\,(1)} \\ \vec{t}^{\,(2)} \end{pmatrix}.$$

Now obtain that

$$F_{[D(\vec{t}^{(0)}),\, D(\vec{q}^{(0)})]}(\overline{A}_\rho^{-1}) = \begin{pmatrix} D(\vec{t}^{(1)}) & O \\ O & D(\vec{t}^{(2)}) \end{pmatrix} \overline{A}_\rho^{-1} - \overline{A}_\rho^{-1} \begin{pmatrix} D(\vec{q}^{(1)}) & O \\ O & D(\vec{q}^{(2)}) \end{pmatrix}$$

$$= \begin{pmatrix} F_{[D(\vec{t}^{\,(1)}),\, D(\vec{q}^{\,(1)})]}(B_{1,1}) & F_{[D(\vec{t}^{\,(1)}),\, D(\vec{q}^{\,(2)})]}(B_{1,2}) \\ F_{[D(\vec{t}^{\,(2)}),\, D(\vec{q}^{\,(1)})]}(B_{2,1}) & F_{[D(\vec{t}^{\,(2)}),\, D(\vec{q}^{\,(2)})]}(\hat{S}^{-1}) \end{pmatrix},$$

which gives us an $s.g._{2r+4}(\overline{A}_\rho^{-1})$.

To solve a singular Cauchy-like linear system $A\vec{x} = \vec{b}$, first compute a vector $\vec{y}$ that satisfies $LAM\vec{y} = L\vec{b}$ and then recover the vector $\vec{x} = M\vec{y}$ that satisfies $A\vec{x} = \vec{b}$. Since L and M are nonsingular, rank $(A) =$ rank (LAM). As a by-product, the algorithm computes rank $(\overline{A})$, which equals rank (A) with a probability at least $1 - 2\rho(\rho+1)/|\mathbf{S}|$, by Theorem 4.2.1.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

By using a standard technique (see e.g. [BP94], p.110), the algorithm can be immediately extended (at additional cost $O(rC_{M_v}(n))$) to the computation of a basis for the null space of A.

Finally, we observe that, by using $O(rC_{M_v}(n))$ ops, we may verify whether $A\vec{x} = \vec{b}$, that is, the overall cost bound for the algorithm covers the cost of the verification of its correctness; furthermore, similar property holds for the rank and null space computation by this approach.

4.3 Solving Singular Toeplitz-like Linear Systems

We will next follow and slightly improve the known best randomized algorithm of [K95] for the solution of a singular Toeplitz-like linear system. (We will use fewer ops and random parameters due to the incorporation of Lemma 4.3.4 below, which is a result from [P92b].)

Definition 4.3.1 *(cf. e.g. [BP94], Definition 11.1). For an $n \times n$ matrix T, define the two displacements,*

$$F_-(T) = T - Z^T T Z, \quad F_+(T) = T - Z T Z^T, \tag{3.1}$$

where $Z = (z_{i,j})$, is a down shift $n \times n$ matrix, $z_{i,j} = 0$ unless $i = j+1$, $z_{i,i+1} = 1$. If for a fixed field $\mathbf{F}$ and for $F = F_+$ or $F = F_-$, we have

$$F(T) = G^* H^{*T}; \; G^*, H^* \in \mathbf{F}^{n \times r}, \tag{3.2}$$

*then the pair of matrices (G^*, H^*) is called an F-*generator* or a* displacement generator *of T of length r and will be denoted by $d.g._r(T)$. The minimum r allowing the above*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

representation (3.2) is called the F-rank (*or* displacement rank) *of T. T is called a* Toeplitz-like *matrix if* $r = O(1)$.

Lemma 4.3.1 *[BA80]. For any $n \times n$ matrix A,*

$$\text{rank } (F_-(A)) - 2 \leq \text{rank } (F_+(A)) \leq \text{rank } (F_-(A)) + 2.$$

Furthermore, given a $d.g._r(T)$ under $F = F_+$ (resp. $F = F_-$), it suffices to use $O(rT_{Mv}(n))$ ops (for $T_{Mv}(n)$ of (2.1)) in order to compute a $d.g._{r+2}(T)$ under $F = F_-$ (resp. $F = F_+$).

Lemma 4.3.2 *[KKM]. Let F_-, F_+, T, G^*, H^*, and r be as in (3.1) and (3.2). Then $F(T) = G^*(H^*)^T = \sum_{i=1}^{r} \vec{g}_i^*(\vec{h}_i^*)^T$ if*

$$T = \sum_{i=1}^{r} L^T(\vec{g}_i^*)L(\vec{h}_i^*) \text{ for } F = F_-, \; T = \sum_{i=1}^{r} L(\vec{g}_i^*)L^T(\vec{h}_i^*) \text{ for } F = F_+,$$

where $G^* = [\vec{g}_1^*, \cdots, \vec{g}_r^*]$, $H^* = [\vec{h}_1^*, \cdots, \vec{h}_r^*]$, and $L(\vec{v})$ is a lower triangular Toeplitz matrix with the first column $\vec{v}$.

Lemma 4.3.3 *(cf. e.g. [BP94], Corollary 12.1). Let T_1 and T_2 be two Toeplitz-like matrices, given with their F-generators of lengths r_1 and r_2, respectively, for $F = F_+$ or $F = F_-$. Then an F-generator of length at most $r_1 + r_2 + 1$ for the matrix T_1T_2 can be computed at the cost of performing $O((r_1 + r_2)^2 T_{M_v}(n))$ ops (cf. (1.1)). Furthermore, a $d.g._r(UTL)$ for a given $d.g._r(T)$ and a given pair of lower triangular Toeplitz matrices L and U^T can be computed at the cost $2r^2T_{M_v}(n)$, provided that $F = F_-$.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Lemma 4.3.4 (cf. Proposition A.6 of [P92b] or [BP94], Problem 2.2.11b). *Given an $d.g._{\tilde{r}}(A) = (G, H)$ and the displacement rank r of A, $r < \tilde{r} \leq n$, one can compute a $d.g._r(A)$ by using $O(\tilde{r}^2 n)$ ops.*

Lemma 4.3.5 *[KKM]. Let T be a nonsingular Toeplitz-like matrix. Then we have* rank $(F_+(T^{-1})) =$ rank $(F_-(T))$.

Lemma 4.3.6 *(cf. [M80], [BA80], [BP94]). Let T be an $n \times n$ strongly nonsingular Toeplitz-like matrix such that*

$$T = \begin{pmatrix} B & C \\ E & J \end{pmatrix}, \qquad S = J - EB^{-1}C,$$

B is a $k \times k$ matrix, and S is the $(n-k) \times (n-k)$ Schur complement of B in T (cf. (2.3)). Let $r = \text{rank}(F_+(T))$. Then

$$\text{rank } (F_-(S^{-1})) = \text{rank } (F_+(S)) \leq r,$$

$$\text{rank } (F_-(B^{-1})) = \text{rank } (F_+(B)) \leq r,$$

$$\text{rank } (F_+(S^{-1})) = \text{rank } (F_-(S)) \leq r + 2,$$

$$\text{rank } (F_+(B^{-1})) = \text{rank } (F_-(B)) \leq r + 2.$$

Proof. The lemma follows from Proposition 3.2.2, Lemma 4.3.1, and Lemma 4.3.5 □

Theorem 4.3.1 *[K95]. For an $n \times n$ Toeplitz-like matrix T of rank ρ, represented by $d.g._r(T)$ satisfying (3.1) and (3.2), let $\tilde{T} = UTL$, where U^T and L are two unit lower triangular Toeplitz matrices whose $2n-2$ entries are randomly sampled from a subset $\mathbb{S}$ of a fixed field containing the entries of T. Then the matrix $\tilde{T}$ has generic rank profile with a probability at least $1 - \rho(\rho+1)/|\mathbb{S}|$.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Due to Lemma 4.3.3, we may compute $d.g._r(\tilde{T})$ at the cost of performing at most $2r^2T_{M_v}(n)$ ops.

Now, given a $d.g._r(\tilde{T})$ for a matrix $\tilde{T} \in \mathbf{F}^{n\times n}$ having generic rank profile, the following algorithm extends one of [K95] and supports (2.2).

Algorithm 4.3.1 *(cf. [PZ,a]) Computing the largest nonsingular leading principal inverse.*

Input: a field $\mathbf{F}$ and vectors $\vec{\tilde{g}}_1, \cdots, \vec{\tilde{g}}_r, \vec{\tilde{h}}_1, \cdots, \vec{\tilde{h}}_r$ from $\mathbf{F}^{n\times 1}$ such that the matrix $\tilde{T} = \sum_{i=1}^{r} L^T(\vec{\tilde{g}}_i)L(\vec{\tilde{h}}_i)$ has generic rank profile.

Output: An integer $\rho \leq n$ and vectors $\vec{\tilde{u}}_1, \cdots, \vec{\tilde{u}}_r$, $\vec{\tilde{v}}_1, \cdots, \vec{\tilde{v}}_r$, $\vec{\tilde{u}}_m, \vec{\tilde{v}}_m \in \mathbf{F}^{n\times 1}$,$m = 1, 2, \cdots, r$, such that $\rho = \text{rank}(\tilde{T})$ and $\tilde{T}_\rho^{-1} = \sum_{m=1}^{r} L(\vec{\tilde{u}}_m)L^T(\vec{\tilde{v}}_m)$.

1. Represent $\tilde{T}$ as $\tilde{T} = \begin{pmatrix} \tilde{B} & \tilde{C} \\ \tilde{E} & \tilde{J} \end{pmatrix}$, as in (2.3), for $k = \lceil \frac{n}{2} \rceil$,where the $k \times k$ submatrix $\tilde{B}$ of $\tilde{T}$ is singular if and only if $k > \rho$ (since $\tilde{T}$ has generic rank profile). Apply Algorithm 4.3.1 recursively to the input matrix $\tilde{B}$ replacing $\tilde{T}$. (Note that the first k components of the given vectors $\vec{\tilde{g}}_i$ and $\vec{\tilde{h}}_i$ define a $d.g._r(\tilde{B})$.) If $\rho \geq k$, the output of this stage is the desired output of the algorithm. Otherwise, the matrix $\tilde{B}$ is nonsingular, and then obtain a $d.g._{r+2}(\tilde{B}^{-1})$ for $F = F_-$ and a $d.g._r(\tilde{B}^{-1})$ for $F = F_+$.

2. Apply Lemma 4.3.3 for $F = F_+$ to compute a $d.g._r(\tilde{S})$ for $\tilde{S} = \tilde{J} - \tilde{E}\,\tilde{B}^{-1}\tilde{C}$.

3. Apply the algorithm recursively to the Toeplitz-like input matrix $\tilde{S}$, replacing $\tilde{T}$. Output $\rho = \text{rank}(\tilde{T}) = k + \text{rank}(\tilde{S})$.

4. By using Definition 4.3.1 and Lemmas 4.3.1-4.3.6, compute $s.g._r(\tilde{T}_\rho^{-1})$ for $F = F_+$.

Let us specify stage 4. Consider the $\rho \times \rho$ leading principal submatrix, $\tilde{T}_\rho = \begin{pmatrix} \tilde{B} & \tilde{G} \\ \tilde{D} & \tilde{R} \end{pmatrix}$, $\tilde{G}$, $\tilde{D}^T \in C^{k\times(\rho-k)}$, $\tilde{R} \in C^{(\rho-k)\times(\rho-k)}$. Write $\check{S} = \tilde{R} - \tilde{D}\,\tilde{B}^{-1}\tilde{G}$. Note that at the preceding

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

stages we have computed $d.g._r(\tilde{G})$ and $d.g._r(\tilde{D})$ for $F = F_-$, $d.g._r(\tilde{B}^{-1})$, $d.g._{2r+1}(-\tilde{B}^{-1}\tilde{G})$, $d.g._{2r+1}(-\tilde{D}\tilde{B}^{-1})$, and $d.g._r(\check{S}^{-1})$ for $F = F_+$. We obtain the following block representation:

$$\tilde{T}_\rho^{-1} = \begin{pmatrix} M_{1,1} & M_{1,2} \\ M_{2,1} & \check{S}^{-1} \end{pmatrix},$$

where $M_{1,2} = -\tilde{B}^{-1}\tilde{G}\ \check{S}^{-1}$, $M_{2,1} = -\check{S}^{-1}\tilde{D}\ \tilde{B}^{-1}$, $M_{1,1} = \tilde{B}^{-1} - M_{1,2}\tilde{D}\ \tilde{B}^{-1}$. By applying Lemmas 4.3.1-4.3.6, we compute $d.g._r(\tilde{T}_\rho^{-1})$ for $F = F_+$.

As in the Cauchy-like case of chapter 4, section 2, algorithm 4.3.1 outputs rank (A) as a by-product and has immediate extension to the computation of a basis for the null space of A.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Chapter 5

Unified Algorithm for Computations with Structured Matrices

5.1 Some Major Classes of Structured Matrices

Hankel matrices and *Hankel-like matrices* of displacement rank r are obtained from Toeplitz matrices and Toeplitz-like matrices of displacement rank r, respectively, by their pre-multiplication (as well as by their post-multiplication) by the *reflection matrix* J, having ones on its antidiagonal and zero entries elsewhere. (Note that J^2 is the identity matrix.) Toeplitz and Toeplitz-like matrix computations and, in particular, all results of chapter 4, section 3 are immediately extended to Hankel and Hankel-like matrix computations, e.g. $H^{-1} = T^{-1}J$, rank (H) = rank (T), and the null spaces of H and T are the same where

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$T = JH$. It is also straightforward to extend Algorithm 4.3.1 to Hankel and Hankel-like computations directly; moreover (as noted in [PACPS98]), Remark 3.2.1 also applies directly to the Hankel-like Schur complements and Hankel-like extensions of Algorithms 3.2.1, 4.2.1 and 4.3.1 but not to Toeplitz-like Schur complements and Algorithm 4.3.1.

$A = (x_i^j)_{i,j=0}^{n-1}$ is an $n \times n$ *Vandermonde matrix* (denoted by $V(\vec{x})$). Vandermonde-like structure can be defined in terms of the operator $A \rightarrow D^{-1}(\vec{t})A - AZ^T$, for a fixed vector $\vec{t}$ (or in terms of some similar linear operators [BP94], [GO94a]): For a field $\mathbf{F}$ and for a vector $\vec{t} = (t_i)$, $t_i \neq 0$, $i = 1, \ldots, n-1$, we call a matrix $A \in \mathbf{F}^{n \times n}$ a *Vandermonde-like* matrix if

$$D^{-1}(\vec{t})A - AZ^T = GH^T; \; G, H \in \mathbf{F}^{n \times r}, \tag{1.1}$$

$r = O(1)$. (Clearly, $r = 1$ for $V(\vec{t})$.) Then the pair of matrices (G, H^T) is a $(D^{-1}(\vec{t}), Z^T)$-*generator* (or *scaling/displacement generator*) of length r for A, and we have

$$A = D(\vec{t}) \sum_{m=1}^{r} D(\vec{g}_m) V(\vec{t}) L^T(\vec{h}_m)$$

for $\vec{g}_m$, $\vec{h}_m$ defined as in Lemma 3.1.1 and for $L(\vec{v})$ defined as in Lemma 4.3.2. The minimum r in all such representations of A is called the $(D^{-1}(\vec{t}), Z^T)$-rank of A.

Our study of Cauchy-like matrices in sections 2-5 and, in particular, Theorems 3.2.2, 4.1.1, 4.2.1, Corollaries 4.1.1, 4.1.2, Remark 3.2.1, Algorithm 4.2.1 and the complexity bound (2.3) can be easily extended to the Vandermonde-like case. On the other hand, the latter bound can be improved to (2.5) for all cited computations with both Cauchy-like and Vandermonde-like input matrices. This is achieved by means of general transformations

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

proposed in [P90], which at the cost $V_{M_v}(n)$ reduce such Cauchy-like and Vandermonde-like computations to ones with Toeplitz-like (or, alternatively, Hankel-like) matrices. In particular, for a matrix A of (1.1), the matrix $\overline{A} = V^T(\vec{t}^{-1})A$ (where $\vec{t}^{-1} = (t_i^{-1})$) has F_--rank r (cf. [BP94], Proposition 2.12 on p. 193) and shares with the above matrix A its rank and null space (because $V^T(\vec{t}^{-1})$ is a nonsingular matrix). The rank and the null space of $\overline{A}$ can be computed based on Algorithm 4.3.1 at the randomized cost bounded by (2.5). The transition from A to $\overline{A}$ costs $O(V_{M_v(n)}))$. On the other hand, $s.g._r(AF^{-1})$ is immediately recovered from (1.1) , where F is the $n \times n$ matrix of discrete Fourier transform [H95].

5.2 Operations with Matrices Represented by Their Short (K, L)-generators

Our next goal is to accelerate the computations with structured matrices (versus ones with general matrices) by relying on compact representation of these matrices via their (K, L)-generators and on the following well-known estimates (cf. [P2000]).

Theorem 5.2.1 *Let us write $Z_{f,v}(n)$, $T_v(n)$, $H_v(n)$, $V_v(n)$, $V_v^T(n)$ and $Z_v(n)$ to denote the numbers of ops required to multiply (over a field $\mathbf{F}$) an $n \times n$ f-circulant, Toeplitz, Hankel, Vandermonde, transposed Vandermonde, and Cauchy matrices, respectively, by a vector. Then we have $Z_{f,v}(n) = O((n \log n) \log\log n)$, $T_v(n) = H_v(n) = O((n \log n) \log\log n)$, $V_v(n) = V_v^T(n) = O((n \log^2 n) \log\log n)$, $Z_v(n) = O((n \log^2 n) \log\log n)$. The factor $\log\log n$ can be removed if $\mathbf{F}$ supports discrete Fourier transforms at the*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

order of n points.

Proof. $V_v^T(n) = V_v(n)$ by Tellegen's theorem (see [PSD70]). For other estimates, see e.g. [BP94], [GO94]. □

Corollary 5.2.1 *For three given scalars e, f , and $(1-ef)^{-1}$ two matrices G, $H \in \mathbf{F}^{n\times \ell}$, and three vectors $\vec{s}$, $\vec{t}$, and $\vec{t} - f\vec{t}^{n+1}$, let the matrices T_f, T_0, $V_f(\vec{t}, G, H)$ and $Z(\vec{s}, \vec{t}, G, H)$ be represented in the form (2.33)-(2.39). Then such matrices can be multiplied by a vector by using at most $\ell Z_{e,v}(n) + (\ell+1)Z_{f,v}(n) + (\ell+2)n + l$, $(2\ell+1)Z_{0,v} + n\ell$, $(Z_{f,v}(n) + V_v(n) + 2n)\ell$, and $Z_v(n)\ell + (3\ell-1)n$ ops, respectively.*

The next simple results (cf. [P2000]) show that the linear combinations, products and inverses of matrices as well as the matrix blocks inherit the matrix structure of the input. In the next section, we will specify and exemplify the applications of these results to the design of superfast algorithms for computations with structured matrices.

Proposition 5.2.1 *(a generator of a linear combination). Let $XK + LX = G_X H_X^T$, $YK + LY = G_Y H_Y^T$. Then $(X + aY)K + L(X + aY) = (G_X, aG_Y)\begin{pmatrix} H_X \\ H_Y^T \end{pmatrix}$ for a scalar a.*

The next result defines a generator of the matrix product and will enable transformations among various classes of structured matrices.

Proposition 5.2.2 *(cf. [P90]). Let $-XW + MX = G_X H_X^T$, $YU + WY = G_Y - H_Y^T$.*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Then

$$(XY)U + M(XY) = (G_X, XG_Y)\begin{pmatrix} H_X^T Y \\ H_Y^T \end{pmatrix}$$

.

Proof $(XY)U + M(XY) = X(YU + WY) + (-XW + MX)Y$. □

For convenience, we summarize in Table 5 the correlation among the basic matrix pairs (K, L) for the matrices X, Y, and XY of Proposition 5.2.2.

Proposition 5.2.3 *(generators of blocks). Let $n = 2m$ be even. Let us write*

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}, L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix}, X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}, Z_f = \begin{pmatrix} Z_0^{(m)} & fU \\ U & Z_0^{(m)} \end{pmatrix},$$

$$G = \begin{pmatrix} G_1 \\ G_2 \end{pmatrix}, H = \begin{pmatrix} H_1 \\ H_2 \end{pmatrix}, D(\vec{v}) = \begin{pmatrix} D_1(\vec{v}) & 0 \\ 0 & D_2(\vec{v}) \end{pmatrix},$$

where $K_{i,j}$, $L_{i,j}$, $X_{i,j}$, and $D_i(\vec{v})$, for $i, j = 1, 2$, as well as $Z_0^{(m)}$ and U are $m \times m$ matrices, from $\mathbf{F}^{m\times m}$; G_i and H_i, for $i = 1, 2$, are $m \times \ell$ matrices, from $\mathbf{F}^{m\times \ell}$; $U = \vec{e}_0^{(m)}(\vec{e}_{m-1}^{(m)})^T$ is a rank 1 matrix; $\vec{e}_0^{(m)}$, $\vec{e}_{m-1}^{(m)}$, are unit coordinate vectors, from $\mathbf{F}^{m\times \ell}$. Let (2.37) hold. Then $X_{ij}\bar{K}_{jj} + \bar{L}_{ii}X_{ij} = G_i H_j^T + R_{ij}$, $i, j = 1, 2$, where

a)

$$R_{11} = (fX_{11} - X_{12})U + fU(X_{21} - X_{11}),$$

$$R_{12} = f(X_{12} - X_{11})U + fU(X_{22} - X_{12}),$$

$$R_{21} = (fX_{21} - X_{22})U + U(X_{11} - fX_{21}),$$

$$R_{22} = f(X_{22} - X_{21})U + U(X_{12} - fX_{22}),$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

if $X = T_f$, $K = Z_f$, $L = -Z_f$, $\bar{K}_{jj} = K_{jj} + fU = -\bar{L}_{ii} = -L_{ij} + fU = Z_0^{(m)} + fU$,

b)

$$R_{11} = f(X_{12} - X_{11})U^T, R_{12} = (X_{11} - fX_{12})U^T,$$

$$R_{21} = f(X_{22} - X_{21})U^T, R_{22} = (X_{21} - fX_{22})U^T,$$

if $X = V_f(\vec{t}, G, H)$, $K = -Z_f^T$, $L = D^{-1}(\vec{t})$, $\bar{K}_{jj} = K_{jj} - fU^T = -(Z_0^{(m)} + fU)^T$, $\bar{L}_{ii} = L_{ii} = D_i(\vec{t})$,

c)

$$R_{11} = R_{12} = R_{21} = R_{22} = 0$$

if $X = C(\vec{s}, \vec{t}, G, H)$, $K = -D(\vec{t})$, $L = D(\vec{s})$, $\bar{K}_{jj} = K_{jj} = -D_j(\vec{t})$, $\bar{L}_{ii} = L_{ii} = D_i(\vec{s})$.

Proposition 5.2.4 *(a generator of the inverse). Let* $XK + LX = G_X H_X^T$ *and let* $\det X \neq 0$ *in* $\mathbf{F}$. *Then*

$$X^{-1}L + KX^{-1} = (X^{-1}G_X)(H_X^T X^{-1})$$

.

Proof. Pre- and post-multiply the matrix equation $XK + LX = G_X H_X^T$ by X^{-1}. □

(Note that the basic matrix pair (K, L) for X is mapped into the dual one for X^{-1}.)

5.3 Divide-and-Conquer Algorithm for Structured Matrices

Suppose that the input matrix X of generalized Algorithm 3.2.1 is given with its short (K, L)-generator for (K, L) from Tables 2-4. The associated descending process defines the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

basic matrix pairs for all matrices involved in the CRF. This stage relies on Proposition 5.2.3 for submatrices, Proposition 5.2.4 for the matrix inversions, and Propositions 5.2.1 and 5.2.2 for matrix additions/subtractions and multiplications. To define the basic matrix pair for a Schur complement (cf. (2.3)), we could have applied Propositions 5.2.1, 5.2.2, and 5.2.4, but it is even simpler to compute the basic matrix pairs successively for X^{-1} (based on Proposition 5.2.4), its southeastern block S^{-1} (based on Propositions 3.2.2 and 5.2.3), and finally, S (by applying Proposition 5.2.4). Similarly, we proceed for other Schur complements involved. Together with the basic matrix pairs (K, L), the descending process defines the (K, L) ranks for all matrices of the CRF. (Here and hereafter, we slightly abuse the notation by writing the same letters K, L for the basic matrix pairs of all matrices of the CRF.)

Proposition 5.3.1 *If the input matrix X of generalized Algorithm 3.2.1 is given with its (K, L)-generator from Tables 2-4, then in the associated descending process the computed basic matrix pairs (K, L) for all matrices of the resulting CRF stay in Tables 2-4.*

Proof is by recursive application of Propositions 3.2.2, 3.2.4, 5.2.1-5.2.4. □

In the lifting process of generalized Algorithm 3.2.1, we compute the $O((m_1 + m)\ell)$ entries of a short (K, L)-generator (of length ℓ) for every $m \times m_1$ matrix Y of the CRF (for K, L defined in the descending process and for $\ell = r_{K,L}(Y)$) and, in the Toeplitz/Hankel-like case, also the m or m_1 entries of the first (or last) column or row of Y, respectively, so that the output fully defines the matrix by (2.33)-(2.38). The computation relies on

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

equations (2.1)-(2.4), Propositions 5.2.1-5.2.3, Corollary 5.2.1, and Fact 1.2.3 (applied to every matrix Y computed with its (K, L)-generator of length ℓ where $\ell > r_{K,L}(Y)$). Such a lifting process will be called compressed computation of Y. Similarly, we operate with short generators when we represent the matrices F and N of (2.8), e.g. implicitly by a short generator of the matrix $-X_{11}^{-1}X_{12}$. Summarizing and taking into account the bound $O(r)$ of Proposition 5.3.1, we obtain the following estimates (cf. (2.7) where c is defined by (2.33)-(2.36), Theorem 1.2.5 and Corollary 5.2.1).

Theorem 5.3.1 *(cf. [P2000]). Let an $n \times n$ input matrix X of generalized Algorithm 3.2.1 have generic rank profile and have (K, L)-rank r for (K, L) of Theorem 1.2.5. Let X be given with its (K, L)-generator of length ℓ, $r < \ell < n$. Let $Z_{f,v}(n)$, $T_v(n)$, $V_v(n)$ and $Z_v(n)$ be defined as in Corollary 5.2.1. Then the compressed computation of a full output set of this algorithm requires $O(n\ell)$ words of memory and*

$$F_{K,L}(n) = O(\ell^2 n + M_{v,K,L}(n) r^2 \log n) \tag{3.1}$$

ops where

$$M_{v,K,L}(n) = Z_{f,v}(n) \tag{3.2}$$

for $K = -L = Z_f$ and $K = Z_f^T$, $L = -Z_f$, for any scalar f,

$$M_{v,K,L}(n) = V_v(n) + Z_{f,v}(n) \tag{3.3}$$

for $K = -Z_f^T$, $L = D^{-l}(\vec{t})$ for any scalar f,

$$M_{v,K,L}(n) = Z_v(n), \tag{3.4}$$

for $K = -D(\vec{t})$, $L = D(\vec{s})$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark 5.3.1 *Generalized Algorithm 3.2.1 amounts to recursive application of Propositions 5.2.1-5.2.3, Corollary 5.2.1 and Fact 1.2.3. As soon as the latter results are extended to the matrix structure definied by a fixed basic pair (K, L), the algorithm, Theorem 5.3.1 and Corollary 5.4.1 of the next section can be extended. In particular, based on equations (2.37), (2.39), one may immediately extend the estimates of Theorem 5.3.1 and Corollary 5.4.1 to the case of all basic matrix pairs (K, L) from Tables 2-4. We also have an immediate extension to the case of the basic matrix pairs $(Z_0 + Z_0^T,\ -Z_0 - Z_0^T)$ and $(Z_0 - Z_0^T,\ Z_0^T - Z_0)$. Both of these pairs define the class of Toeplitz-like + Hankel-like matrices (see [BP93], [BP94], pp.185-188).*

5.4 Transformations Among Structured Matrices and Acceleration of Vandermonde-like and Cauchy-like Computations

For the Vandermonde-like and Cauchy-like input matrices X, we may further improve the asymptotic cost estimates of (3.1), (3.3), (3.4) to yield the Toeplitz/Hankel-like level of (3.1), (3.2) by applying Proposition 5.2.2 and Table 5 (cf. [P90]).

Indeed, let $Y = V_f(\vec{t}, G, H)$ (cf. (2.35)) and write $U = -Z_f^T$, $W = D^{-1}(\vec{t})$, $X = -V^T(\vec{t})$, $M = Z_f$, $YU + WY = G_Y H_Y^T$; $G_Y, H_Y \in \mathbf{F}^{n \times \ell}$. Then we have $MX - XW = -V^T(\vec{t})D^{-1}(\vec{t}) - Z_f V^T(\vec{t}) = \vec{e}_0(\vec{t}^{-1} - f\vec{t}^{n-1})^T Y$, and application of Proposition 5.2.2 yields $-(XY)Z_f^T + Z_f(XY) = (XY)U + M(XY) = (XG_Y)H_Y^T + \vec{e}_0((\vec{t}^{-1} - f\vec{t}^{n-1})^T Y)$, that is, we obtain a $(-Z_f^T, Z_f)$-generator of length $\ell + 1$ for the matrix $-V_f^T(\vec{t})Y$. By (2.35) and

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Corollary 5.2.1, this computation requires multiplication of $X = -V^T(\vec{t})$ by G_Y, and of the vector $(\vec{t}^{-1} - f\vec{t}^{n-1})^T$ by Y, that is, $(V_v^T(n) + 2n - 1)\ell = O((\ell n \log^2 n) \log\log n)$ ops (not counting $O(n \log n)$ ops for the computation of the vector $\vec{t}^{-1} - f\vec{t}^{n-1}$).

Similarly, we compute a $(Z_f^T, -Z_f)$-generator of length $\ell + 2$ for the matrix $V^T(\vec{s}^{-1})YV(\vec{t}^{-1})$ provided that a matrix $Y = C(\vec{s}, \vec{t}, G, H)$ of (2.36) is given with its $(-D(\vec{t}), D(\vec{s}))$-generator of length ℓ. The computation uses $(V_v^T(n) + V_v(n) + O(n))\ell = O(n\ell \log^2 n) \log\log n)$ ops. By Theorem 5.2.1, we may compute a full output set of generalized Algorithm 3.2.1 applied to the respective Hankel-like input matrices, $X = -V_f^T(\vec{t})V_f(\vec{t}, G, H)$ or $X = V^T(\vec{s}^{-1})C(\vec{s}, \vec{t}, G, H)V(\vec{t}^{-1})$, at the cost bounded according to (3.1), (3.2). Then we may obtain easily the partial output sets (cf. Definition 3.2.2) of the same algorithm for the matrices $V_f(\vec{t}, G, H)$ and $C(\vec{s}, \vec{t}, G, H)$ at the cost dominated by (3.1), (3.2). More precisely, we obtain

Corollary 5.4.1 *Under the assumptions of Theorem 5.3.1, generalized Algorithm 3.2.1 applied to compressed computation of a partial output set of the matrix X requires $O(n\ell)$ words of memory and $V_v(n) + F_{K,L}(n)$ ops, for $F_{K,L}(n)$ bounded by (3.1), (3.2), that is, $F_{K,L}(n) = O(n\ell^2 + (nr^2 \log^2 n) \log\log n)$ over any field of constants.*

The above transformations among various classes of structured matrices use *Vandermonde multipliers*, as this was proposed in [P90]. In the special case where $\vec{s}$ and/or $\vec{t}$ are the (scaled) vectors of roots of 1, the Vandermonde multipliers turn into the matrices of (scaled) Fourier transforms (*Fourier multipliers*). In this special case the transformations are simplified [H95], [GKO95]. In particular, next, we will simplify the above

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

maps to Toeplitz/Hankel-like matrices for some important subclasses of the input classes of Vandermonde-like and Cauchy-like matrices in the case where the field **F** supports the n-point FFT. We will use the following definition and auxiliary results.

Definition 5.4.1 *Write* $\vec{e} = (g^i)_{i=0}^{n-1}$, $f = (h^i)_{i=0}^{n-1}$ *provided that* $g^n = e$, $h^n = f$; $\vec{w} = (\omega^i)_{i=0}^{n-1}$ *where* ω *is a primitive* n*-th root of* 1, $\omega^n = 1$, $\omega^s \neq 1$ *for* $s = 1, \ldots, n-1$; $\Omega = (\omega^{ij})_{i,j=0}^{n-1}$. *(*$\Omega$ *denotes the matrix of the* n*-point discrete Fourier transform, DFT. In the complex field* C*, we may choose* $\omega = \exp(2\pi\sqrt{-1}/n)$*, and we have* $\Omega H \Omega = I$*.)*

Proposition 5.4.1 *(cf. [CPW74]). Let* $f \neq 0$, $h^n = f$. *Then*

$$Z_f = U_f^{-1} D(g\vec{w}) U_f \tag{4.1}$$

where

$$U_f = \Omega D(\vec{f}). \tag{4.2}$$

Proposition 5.4.2 *Let* $J\vec{t} = h\vec{w}$, $f = h^n \neq 0$, $X = JV_f(\vec{t}, G, H)J$,

$$XZ_f - D^{-1}(J\vec{t})X = G_X H_X^T \tag{4.3}$$

(cf. Table 3). Then we have (cf. Theorem 1.2.5a) that $T_f Z_f - Z_f T_f = GH^T$,

$$G = U_f^{-1} G_X, H = H_X, T_f = U_f^{-1} X, \tag{4.4}$$

for U_f *of (4.2).*

Proof. Pre-multiply (4.3) by U_f^{-1} and substitute equations (4.1), (4.2) and (4.4). □

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proposition 5.4.3 *(cf. [P90], [H95], [GKO95]). Let $\vec{s} = g\vec{w}$, $\vec{t} = h\vec{w}$, $e = g^n \neq 0$, $f = h^n \neq 0$, $Y = C(\vec{s}, \vec{t}, G, H)$. Define U_e, and U_f by (4.2). Let*

$$YD(\vec{t}) - D(\vec{s})Y = G_Y, H_Y^T \tag{4.5}$$

(cf. Table 4). Then we have (cf. Theorem 1.2.5a) that $\tilde{T}Z_e - Z_f\tilde{T} = GH^T$,

$$G = U_f^{-1}G_Y, H^T = H_Y^T U_e, \tilde{T} = U_f^{-1}YU_e. \tag{4.6}$$

Proof. Pre-multiply (4.5) by U_f^{-1} , post-multiply by U_e, and substitute equations (4.1), (4.2), (4.6), and $U_e = \Omega D(\vec{e})$, which is a variation of (4.2). □

Remark 5.4.1 *The equation $\tilde{T}Z_e - Z_f\tilde{T} = GH^T$ defines a $(Z_e, -Z_f)$-generator (G, H) for $\tilde{T}$. Equivalently, we have $\tilde{T}Z_f - Z_f\tilde{T} = \tilde{G}\tilde{H}^T$ where $\tilde{G} = (G, \tilde{T}\vec{e}^{(0)})$, $\tilde{H} = (H, (f - e)\vec{e}^{n-1})$. By Theorem 1.2.5a) for $\tilde{T}$, $\tilde{G}$, $\tilde{H}$ replacing T_f, G, H, respectively, we conclude that $(\tilde{G}, \tilde{H})$ is a $(Z_f, -Z_f)$-generator for $\tilde{T}$, whose length increases at most by 1 versus the $(Z_e - Z_f)$ generator (G, H). Proposition 5.4.3 enables us to avoid even such a minor increase of the length. Indeed, by Theorem 1.2.5d, we express $Y = C(\vec{s}, \vec{t}, G, H)$ by applying (2.36) and then to obtain a (Z_e, Z_f)-generator of the same length ℓ for the matrix $\tilde{T}$, by applying (4.6).*

By extending Propositions 5.4.2 and 5.4.3, we may transform the (K, L)-generators of all the Vandermonde- like and Cauchy-like matrices associated with basic matrix pairs (K, L) of Tables 3 and 4 (at the cost of 1 or 2 diagonal scalings and performing 1 or 2 DFTS) into generators of the same length for Toeplitz/Hankel-like matrices provided that K and L are of the form $\pm Z_f$, $\pm Z_f^T$ and/or $\pm D(g\vec{w})$ for two scalars, g and $f = g^n \neq 0$. The first (or

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

last) column or row of every resulting matrix can be also computed easily. (All the above computations are further simplified slightly in the case where $g = f = 1$, $D(\vec{f}) = I$, and $U_f = \Omega$.)

Remark 5.4.2 *One may post-multiply (4.3) by U_f^{-1} (cf. (4.2)) and then substitute (4.1) to map the Vandermonde-like matrix X of (4.3) into the Cauchy-like matrix XU_f^{-1} satisfying the matrix equation $(XU_f^{-1})D(g\vec{w}) - D^{-1}(J\vec{t})(XU_f^{-1}) = G_X(H_X^T U_f^{-1})$. Similarly, we may map every matrix pair of Table 3 into one of Table 4 and also any Toeplitz/Hankel-like matrix into a closely related matrix from either of Tables 3 and 4. [GKO95] uses such maps to improve Toeplitz-like matrix computations with pivoting.*

5.5 Transformations of a General Matrix X into a Matrix with Generic Rank Profile and an Extended Randomized Algorithm

To relax the generic rank profile assumption for the input, we may apply our computations with pivoting, but pivoting generally destroys matrix structure. As an exception, we may preserve Cauchy-like structure in Gaussian but not block Gaussian elimination, with pivoting, that is, where we eliminate the variables one by one, rather than by blocks, but in this case the best resulting algorithms involve order of n^2 ops [GKO95].

On the other hand, for nonsingular real or complex matrices X, symmetrization ensures strong nonsingularity and preserves all the structures of Tables 2-4. (Exception is the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Cauchy-like complex case where symmetrization generally destroys the structure. In the real Cauchy-like case, one should use implicit representation of X^TX or XX^T by the pair of generators of X^T and X in order to satisfy the assumption of Definition 1.2.2 that $s_i \neq t_i$ for all i.)

Next, we will apply randomization to transform (over any field $\mathbf{F}$) a general (unstructued or structured and possibly singular) matrix X into one having generic rank profile. We will use the following result.

We also have

Theorem 5.5.1 *(cf. [M82], [C841]). An $n \times n$ Cauchy matrix $C(\vec{s}, \vec{t})$ is nonsingular if and only if all the $2n$ components of the vectors $\vec{s}$ and $\vec{t}$ are distinct. Every square submatrix of nonsingular Cauchy matrix is nonsingular.*

Next, for a given matrix X, we will define some pairs of random structured matrices (preconditioners) A and B to yield generic rank profile for the matrix

$$\tilde{X} = AXB. \tag{5.1}$$

Our next result generalizes one of [KS91] for triangular Toeplitz preconditioners A_2, B_2 (cf. also unstructured preconditioners used in the pioneering paper [BGH82]).

Theorem 5.5.2 *(cf. [P2000]). Let us be given a matrix $X \in \mathbf{F}^{n\times n}$, and six vectors $\vec{p}$, $\vec{q}$, $\vec{u}$, $\vec{v}$, $\vec{y} = (y_j)$, $\vec{z} = (z_j) \in \mathbf{F}^{n\times 1}$, where each of the two pairs of vectors $\vec{p}$, $\vec{q}$ and $\vec{u}$, $\vec{v}$ is filled with $2n$ distinct scalars, $y_0 = z_0 = 1$, and y_j, z_j are indeterminates for $j > 0$. Let $\vec{1} = (1)_{i=0}^{n-1} = \sum_{i=0}^{n-1} \vec{e}_i$. Then the matrix $\tilde{X}$ of (5.1) has generic rank*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

profile if $(A,B) = (A_\alpha, B_\beta)$, $(\alpha,\beta) \in \{(1,1),(2,2),(1,2),(2,1)\}$ *where* $A_1 = C(\vec{p},\vec{q},\vec{1},\vec{y})$, $B_1, = C(\vec{u},\vec{v},\vec{z},\vec{1})$, $A_2 = Z_0^T(\vec{y})$, $B_2 = Z_0(\vec{z})$.

Proof. See proof of Theorem 4.2.1. □

Corollary 5.5.1 *Under the assumptions of Theorem 5.5.2, let the values of the* $2n-2$ *variables,* y_j *and* z_j, $j = 1,\ldots,n-1$, *be randomly sampled from a fixed finite set* $\mathbf{S}$ *of cardinality* $|bfS|$. *Then the matrix* $\tilde{X} = AXB$ *of rank* ρ *has generic rank profile with a probability at least* $1-(\rho+1)\rho/|\mathbf{S}|$.

Proof. $\det \tilde{X}$ is a polynomial in the variables $y_1, z_1, \ldots, y_{n-1}, z_{n-1}$, of a total degree at most $2k$ for $k < \rho$ it does not vanish identically in these variables by Theorem 5.5.2. By Lemma 4.1.1, it may vanish with a probability at most $2k/|S|$ under the random sampling of the variables. Therefore, the probability that neither of $\det \tilde{X}^{(k)}$ vanishes under the random sampling is at least $\prod_{k-1}^{\rho}(1-2k/|S|) > 1-(\rho+1)/|S|$. □

Corollary 5.5.1 can be combined with generalized Algorithm 3.2.1 as follows.

Algorithm 5.5.1 *Randomized computation of a partial output set for a general matrix.*

Input: *a field* $\mathbf{F}$, *a pair* (α,β), α, $\beta \in \{1,2\}$ *an* $n \times n$ *matrix* $X \in \mathbf{F}^{n\times n}$, *and a vector* $\vec{b} \in \mathbf{F}^{n\times 1}$.

Output: *a partial output set of generalized Algorithm 3.2.1 applied to* X.

Computations:

1. *Fix a finite set* S *of non-zero elements of* $\mathbf{F}$ *(cf. Remark 5.5.3 at the end of the section) or its extension and randomly sample from* S *the* $2n-2$ *elements*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

$y_i, z_i, i = 1, \ldots, n-1$, *defining two matrices,* $A = A_\alpha$ *and* $B = B_\beta$, *of Theorem 5.5.2.*

2. *Compute the matrix* $\tilde{X} = AXB$.

3. *Apply-generalized Algorithm 3.2.1 to the matrix* $\tilde{X}$, *which in particular outputs* ρ *rank* $\tilde{X}$.

4. *Compute the matrices* F *and* N *of (2.8) for* X *replaced by* $\tilde{X}$. *Verify whether the matrix* $\tilde{X}N$ *(formed by the* $n - \rho$ *last columms of the matrix* $\tilde{X}F$*) is a null matrix. If "not", output FAILURE, which indicates that the randomization failed to insure the generic rank profile property for* X.

5. *Otherwise compute the matrix* BN *whose columns form a basis for the null space of* X.

6. *If the linear system* $\tilde{X}\vec{w} = A\vec{b}$ *has no solution* $\vec{w}$, *then also the system* $X\vec{y} = \vec{b}$ *has no solution* $\vec{y}$. *In this case output INCONSISTENT. Otherwise compute a solution* $\vec{w}$ *to the former system and then a solution* $\vec{y} = B\vec{w}$ *to the latter one.*

7. *If* $\rho = n$, *compute* $X^{-1} = B\tilde{X}^{-1}A$ *and* $\det X = (\det \tilde{X})/((\det A) \det B)$.

Correctness of Algorithm 5.5.1 is easily verified based on (2.8), (2.9) and (5.1).

The computational cost (in the case of a general matrix X) is clearly dominated by the cost of the application of generalized Algorithm 3.2.1 (we ignore the cost of generation of random parameters).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Remark 5.5.1 *One may re-apply Algorithm 5.5.1 in the case if FAILURE is output. In ν applications, the probability of outputing FAILURE ν times is at most $((\rho+1)\rho/|S|)^{\nu}$*

Remark 5.5.2 *To increase* $|\mathbf{S}|$*, one may need to extend* $\mathbf{F}$ *if* $\mathbf{F}$ *contains only a few elements. Operating in the extension* $\mathbf{F}[x]/q(x)$ *for an irreducible polynomial* $q(x)$ *of degree* d *increases the cardinality of* $\mathbf{F}$ *by factor* 2^d *at the price of increasing the time cost of our computations by, factor* $O(d \log d)$.

Remark 5.5.3 *The restriction that the set* $\mathbf{S}$ *should not contain* 0 *is needed to insure nonsingularity of the matrices* A *and* B *except for the case where* $A = A_2$, $B = B_2$. *In the latter case we may let* $\mathbf{S}$ *contain* 0.

Remark 5.5.4 *Instead of* A_2 *and* B_2*, we may choose* $A_3 = Z_f^T(\vec{y})$, $B_3 = Z_f(\vec{z})$, *for an indeterminate* f. *Then, clearly, Theorem 5.5.2 is extended. If the value* f *is randomly sampled from the set* S*, the proof of Corollary 5.5.1 is also extended, except that the degree of* $\det \tilde{X}$ *(as a polynomial in* $f, y_1, z_1, \ldots, y_{n-1}, z_{n-1}$*) is doubled, and so is the failure probability. Such a minor deficiency can be weighted versus the minor acceleration (by roughly factor* 2*) of the complexity of the computations with triangular Toeplitz matrices* $A_2 = Z_0^T(\vec{y})$, $B_2 = Z_0^T(\vec{Z})$ *in the transition to the* f*-circulant matrices* $A_3 = Z_f^T(\vec{y})$, $B_3 = Z_f(\vec{z})$ *for* $f \neq 0$.

Remark 5.5.5 *The results of this section rely on using* Cauchy-like multipliers A_1, B_1, *and* Toeplitz multipliers A_2, B_2, A_3, B_3. *Vandermonde multipliers of section 6 could*

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

be also used, but then we would have dealt with polynomials of degree as high as $(\rho+1)\rho n/2$*, with the resulting increase by factor* n *of the failure probability bound.*

5.6 Randomization for a Structured Input Matrix X

Let us next show appropriate choices of the matrices A and B of (5.1) that preserve the structure of some sample matrices from Tables 2-4. (The extension to all other matrices of these tables is straightforward.)

a) For $X = T_f$, $X = JT_f$, and $X = T_f J$ (cf. (2.33), (2.34)) and also for $X = T_f + JT_e$ we may choose $A = A_2$, $B = B_2$ or $A = A_3$, $B = B_3$ (cf. Remark 5.5.4), thus preserving the Toeplitz/Hankel-like structure of X in the transition to $\tilde{X}$ (cf. Proposition 5.2.2 and Tables 2 and 5).

b) For $X = C(\vec{s}, \vec{t}, G, H)$ (cf. (2.36) and Tables 4 and 5), we choose $A = A_l$, $B = B_1$, $\vec{q} = \vec{s}$, $\vec{u} = \vec{t}$ and arrive at $\tilde{X} = C(\vec{p}, \vec{v}, \tilde{G}, \tilde{H})$ where we may choose any pair $(\vec{p}, \vec{v})$.

c) For $X = V_f^T(\vec{t}, G, H)$ (cf. (2.35) and Tables 3 and 5), we choose $A = A_2$, $B = B_1$, with $\vec{u} = -\vec{t}^{-1}$ and arrive at the matrix $\tilde{X} = V_f^T(-\vec{v}^{-1}, \tilde{G}, \tilde{H})$, for $\tilde{G}$, $\tilde{H}$ defined by Proposition 5.2.2, where we may choose any vector $\vec{v}^{-1}$. Likewise, for $X = V_f(\vec{t}, G, H)$, we choose $A = A_1$, $B = B_2$ with $\vec{q} = \vec{t}^{-1}$ and arrive at the matrix $X = V_f(-\vec{v}, \overline{G}, \overline{H})$, for $\overline{G}, \overline{H}$ defined by Proposition 5.2.2 where we may choose any vector $\vec{v}$.

Clearly, in all cases a)-c) above, the matrix $\tilde{X}$ belongs to the same class of structured matrices of Tables 2-4 as X does. We also deduce from Proposition 5.2.2 that the length of the generator for X increases by at most 2 in the above sample transitions to $\tilde{X}$. It follows immediately that the estimates of Theorem 5.3.1 and Corollary 5.4.1 apply to the

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

randomized computational cost of performing Algorithm 5.5.1 too.

Moreover, in the case of Vandermonde-like and Cauchy-like computations over the fields $\mathbf{F}$ that support FFT at $n = 2^h$ points for an integer h, we may achieve the cost level (3.1), (3.2) in the cases b) and c) above as a by-product of randomization, by choosing $\vec{p} = g\vec{w}$ and $\vec{v} = h\vec{w}$ for any nonzero scalars g and h such that the vectors $g\vec{w}$ and $h\vec{w}$ have $2n$ distinct components and by applying Propositions 5.4.2 and 5.4.3 to the matrix $\tilde{X}$ and to the matrices of smaller size involved in the computations by generalized Algorithm 3.2.1 applied to $\tilde{X}$. (By Propositions 5.2.1-5.2.4, the structure allowing application of Propositions 5.4.2 and 5.4.3 is preserved for the latter matrices of smaller size.)

5.7 Simplified Expressions for the Generators of Schur Complements

The computations of short generators of Schur complements in the generalized Algorithm 3.2.1 can be a little simplified based on the next result (cf. [P2000]), which generalizes one of [GO94a], [OP98] (where it was assumed that $K_{21} = L_{12} = 0$).

Proposition 5.7.1 *Under the assumptions of Proposition 5.2.3, let X_{11}, be a nonsingular matrix. Then the Schur complement S of X_{11}, in X satisfies the following matrix equations:*

$$SK_{22} + L_{22}S = (G_2 - X_{21}X_{11}^{-1}G_1)(H_2^T - H_1^T X_{11}^{-1} X_{12}) + R,$$

$$R = SK_{21}X_{11}^{-1}X_{12} + X_{21}X_{11}^{-1}L_{12}S. \tag{7.1}$$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Proof. Recall Proposition 3.2.2 and the definitions of Proposition 5.2.3 and also write

$$\overline{X} = X^{-1} = \begin{pmatrix} \overline{X}_{11} & \overline{X}_{12} \\ \overline{X}_{21} & S^{-1} \end{pmatrix}$$

Deduce from Proposition 5.2.4 that $K\overline{X} + \overline{X}L = \overline{X}GH^T\overline{X}$. Deduce from the latter equation that $K_{22}S^{-1} + S^{-1}L_{22} + K_{21}\overline{X}_{12} + \overline{X}_{21}L_{12} = (\overline{X}_{21}G_1 + S^{-1}G_2)(H_1^T\overline{X}_{12} + H_2^TS^{-1})$ and consequently $SK_{22} + L_{22}S = (S\overline{X}_{21}G_1 + G_2)(H_1^T\overline{X}_{12}S + H_2^T) - SK_{21}\overline{X}_{12}S - S\overline{X}_{21}L_{12}S$. Substitute the expressions from (2.4) for $\overline{X}_{12}S$ and $S\overline{X}_{21}$, and obtain Proposition 5.7.1. □

By (7.1), we have rank $R <$ rank K_{21} + rank L_{12}. The expressions of Proposition 5.7.1 for a (K, L)-generator of S are the simplest where $K_{21} = L_{12} = 0$, which implies that $R = 0$ (cf. (7.1)). This is the case where $K = -L^T = Z_0^T$; $K = -Z_0^T$, $L = D^{-1}(\bar{t})$, and $K = -D(\bar{t})$, $L = D(\bar{s})$, which correspond to the Hankel-like, Vandermonde-like and Cauchy-like structures, respectively. In the Hankel-like case, this implies dealing with Z_f^T for $f = 0$, that is, we must operate with triangular Toeplitz matrices, which means a minor slowdown (roughly by factor 2, [BP94], pp. 133-135) versus the same operations with f-circulant matrices involved in the case where $f \neq 0$. The cost of such a slowdown can be weighted versus the decrease of the cost of handling non-null matrices R of (7.1) when we shift from $f \neq 0$ to $f = 0$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Bibliography

[AHU] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts, 1974.

[BA80] 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.

[BGH82] A. Borodin, J. von zur Gathen, J. Hopcroft, Fast Parallel Matrix and GCD Computation, *Information and Control*, **52**, **3**, 241-256, 1982.

[BP93] D. Bini, V. Y. Pan, Improved Parallel Computations with Toeplitz-like and Hankel-like Matrices, *Linear Algebra Appl.*, **188**, **189**, 3-29, 1993.

[BP94] D. Bini, V. Y. Pan, *Polynomial and Matrix Computations, Volume 1: Fundamental Algorithms*, Birkhäuser, Boston, 1994.

[C841] A. L. Cauchy, Mémorie sur les Fonctions Alternées et sur les Somme Alternées, *Exercises d' Analyse et de Phys. Math.*, **II**, 151-159, 1841.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[CKL-A87] J. Chun, T. Kailath, H. Lev-Ari, Fast Parallel Algorithm for QR-factorization of Structured Matrices, *SIAM J. Sci. Stat. Comput.*, **8**, **6**, 899-913, 1987.

[CPW74] R. E. Cline, R. J. Plemmons, G. Worm, Generalized Inverses of Certain Toeplitz Matrices, *Linear Algebra Appl.*, **8**, 25-33, 1974.

[DL78] R.A. Demillo, R.J. Lipton, A Probabilistic Remark on Algebraic Program Testing, *Information Process. Letters*, **7**, **4**, 193-195, 1978.

[FHR93] T. Fink, G. Heinig,K. Rost, An Inversion Formula and Fast Algorithms for Cauchy-Vandermonde Matrices, *Linear Algebra Appl.*, **183**, 179–191, 1993.

[Ger87] A. Gerasoulis, A Fast Algorithm for the Multiplication of Generalized Hilbert Matrices with Vectors, *Math. Comp.*, **50**, **181**, 179–188, 1987.

[GGS] A. Gerasoulis, M.D. Grigoriadis Liping Sun, A fast algorithm for Trummer's problem, *SIAM J. Sci. Statist. Comput.*, **v. 8**, 1987, pp. s135-s138

[GKO95] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian Elimination with Partial Pivoting for Matrices with Displacement Structure, *Math. of Computation*, **64**, 1557-1576, 1995.

[GK] I. Gohberg and I. Koltracht, *Efficient algorithm for Toeplitz plus Hankel matrices*, Integral Equations and Operator Theory, **12** (1989), 136-142.

[GL] G.H. Golub, C.F. Van Loan, *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, Maryland, 1996 (third edition).

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[GO2] I. Gohberg, V. Olshevsky, Fast Inversion of Chebyshev-Vandermonde matrices, *Numerische Mathematik*, **67**, **1**, 71-92, 1994.

[GO92] I. Gohberg, V. Olshevsky, Circulants, Displacements and Decompositions of Matrices, *Integral Equations and Operator Theory*, **15**, **5**, 730-743, 1992.

[GO94] I. Gohberg, V. Olshevsky, Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems, *Integral Equations and Operator Theory*, **20**, 1, 44-83, 1994.

[GO94a] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra Appl.*, **202**, 163-192, 1994.

[GR87] L. Greengard and V. Rokhlin, A Fast Algorithm for Particle Simulations, *J. of Comput. Physics*, **73**, **2**, December 1987

[H95] G. Heinig, Inversion of Generalized Cauchy Matrices and the Other Classes of Structured Matrices, *Linear Algebra for Signal Processing, IMA Volume in Math. and Its Applications*, **69**, 95-114, Springer, 1995.

[HJR] G. Heinig, P. Jankowski and K. Rost, *Fast inversion of Toeplitz-plus-Hankel matrices*, Numer. Math. **52** (1988), 665-682.

[HR] Heinig G., Rost K., *Algebraic methods for Toeplitz-like matrices and operators*, Operator Theory, vol. 13, Birkauser, Basel, 1984.

[IMH82] O. H. Ibarra, S. Moran, R. Hui, A Generalization of the Fast LUP Matrix Decomposition Algorithm and Applications, *J. of Algorithms*, **3**, 45-56, 1982.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[K95] E. Kaltofen, Analysis of Coppersmith's Block Wiedemann Algorithm for the Parallel Solution of Sparse Linear Systems, *Mathematics of Computation*, **64, 210**, 777-806, 1995.

[KKM] T. Kailath, S.Y. Kung, M. Morf, Displacement Ranks of Matrices and Linear Equations, *J.Math. Anal. Appl.*, **68, 2**, 395-407, 1979.

[KO] T. Kailath and V. Olshevsky, *Displacement structure approach to Chebyshev-Vandermonde and related matrices*, submitted, 1994.

[KO96] T. Kailath, V. Olshevsky, Displacement Structure Approach to Discrete Transform Based Preconditioners of G. Strang Type and of T. Chan Type, *Calcolo*, **33**, 191-208, 1996.

[KO97] T. Kailath, V. Olshevsky, Displacement Structure Approach to Polynomial Vandermonde and Related Matrices, *Linear Algebra and Its Applications*, **261**, 49-90, 1997.

[KS91] E. Kaltofen, B. D. Saunders, On Wiedemann's Method for Solving Sparse Linear Systems, *Proc. AAECC-5, Lecture Notes in Computer Science*, **536**, 29-38, Springer, Berlin, 1991.

[KS99] T. Kailath, A. Sayed (editors), *Fast Reliable Algorithms for Matrices with Structure*, SIAM Publications, Philadelphia, 1999.

[LT] P. Lancaster, M. Tismenetzky, Theory of matrices with Applications, 2nd ed., *Academic Press*, Orlando, 1985.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[M74] M. Morf, Fast Algorithms for Multivariable Systems, PhD Thesis, Stanford University, Stanford, CA, 1974.

[M80] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proc. IEEE Internat. Conf. on ASSP*, 954-959,1980.

[M82] L. Mirsky, *An Introduction to Linear Algebra*, Dover, New York, 1992.

[OP98] V. Olshevsky,V. Y. Pan, A Unified Superfast Algorithm for Boundary Rational Tangential Interpolation Problem, *Proc. 39th Ann. IEEE Symp. Foundations of Comp. Sci.*, 192-201, IEEE Comp. Soc. Press, 1998.

[OS88] A.M. Odlyzko, A. Schönhage, Fast Algorithms for Multiple Evaluations of the Riemann Zeta Function, *Trans. Am. Math.Soc.*, 309, 2, 797–809, 1988.

[OS99] V. Olshevsky, M. A. Shokrollahi, A Displacement Approach to Efficient Decoding of Algebraic-Geometric Codes, *Proc. 31st Ann. Symp. on Theory of Computing*, ACM Press, New York, 1999.

[P2000] Victor Y. Pan, Nearly Optimal Computations with Structured Matrices, *Proc. of 11th Ann. ACM-SIAM Symposium on Discrete Algorithms*, ACM Press, New York, and SIAM Publications, Philadelphia, January 2000.

[P90] V.Y. Pan, On Computations with Dense Structured Matrices, *Proc. ACM-SIGSAM Intern. Symp. on Symbolic and Alg. Comp.*, 34-42, ACM Press, New York, 1989, and *Math. of Computation.*, **55, 191**, 179-190, 1990.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[P92] V.Y. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, *Computers and Math. (with Applics.)*, **24**, **3**, 61-75, 1992.

[P92b] V.Y. Pan, Parametrization of Newton's Iteration for Computations with Structured Matrices and Applications, *Computers and Mathematics (with Applications)*, **24**, **3**, 61-75, 1992.

[P93a] V.Y. Pan, Decreasing the Displacement Rank of a Matrix, *SIAM J. Matrix Anal. Appl.*, **14**, **1**, 118-121, 1993.

[P95] V. Y. Pan, An Algebraic Approach to Approximate Evaluation of a Polynomial on a Set of Real Points, *Advances in Computational Mathematics*, **3**,41–58, 1995.

[PACLS,a] V. Y. Pan, M. Abu Tabanjeh, Z. Q. Chen, E. Landowne, A. Sadikou, New Transformations of Cauchy Matrices and Trummer's Problem, *Computers & Math. (with Applications)*, **35**, **12**, 1-5, 1998.

[PACPS98] V. Y. Pan, M. Abu Tabanjeh, Z. Chen, S. Providence, A. Sadikou, Transformations of Cauchy Matrices for Trummer's Problem and a Cauchy-like Linear Solver, *Proc. of 5th Annual International Symposium on Solving Irregularly Structured Problems in Parallel(Irregular98)*, (A. Ferreira, J. Rolim, H. Simon, S.-H. Teng Editors), *Lecture Notes in Computer Science*, **1457**, 274-284, Springer, 1998.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[PACPZ99] V. Y. Pan, M. Abu Tabanjeh, Z. Chen, S. Providence, A. Zheng, Superfast Computations with Singular Structured Matrices over Abstract Fields, *Proc. 2nd Workshop on Computer Algebra in Scientific Computing (CASC'99)*, (V.G. Ganzha, E.W. Mayr, and E.V. Vorontsov, Editors), 323-338, Springer, Berlin, 1999

[PLST93] V.Y. Pan, E. Landowne, A. Sadikou, O. Tiga, A New Approach to Fast Polynomial Interpolation and Multipoint Evaluation, *Computers & Math. (with Application)* 25, 9, 25–30,1993.

[PSD70] P. Penfield Jr., R. Spencer, S. Duinker, *Tellegen's Theorem and Electrical Networks*, MIT Press, Cambridge, Massachusetts, 1970.

[PZ,a] V. Y. Pan, A. Zheng, Superfast Algorithms for Cauchy-like Matrix Computations and Extensions, accepted by *Linear Algebra and Its Applications.*

[PZHY97] V. Y. Pan, A. Zheng, X. Huang, Y. Yu, Fast Multipoint Polynomial Evaluation and Interpolation via Computation with Structured Matrices, *Annals of Numerical Math.*,4, 483–510, 1997.

[Rok85] V. Rokhlin, Rapid Solution of Integral Equations of Classical Potential Theory, *J. Comput. Physics*, **60**, 187-207, 1985.

[R88] V. Rokhlin, A Fast Algorithm for the Discrete Laplace Transformation, *J. of Complexity*, **4**, 12–32, 1988.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

[S80] J.T. Schwartz, Fast Probabilistic Algorithms for Verification of Polynomial Identities, *J. of ACM*, **27**, **4**, 701-717, 1980.

[SLAK] A. Sayed, H. Lev-Ari and T. Kailath, *Fast triangular factorization of the sum of quasi-Toeplitz and quasi-Hankel matrices*, Linear Algebra and Appl., **191** (1993), 77-106.

[St69] V. Strassen, Gaussian Elimination is Not Optimal, *Numer. Math.*, **13**, 354-356, 1969.

[SW98] M. A. Shokrollahi, H. Wasserman. Decoding Algebraic-Geometric Codes Beyond the Error-Correction Bound, *Proc. 30th Annual Symp. on Theory of Computing*, 241-248, ACM Press, New York, 1998.

[Su97] M. Sudan, Decoding of Reed-Solomon Codes Beyond the Error-Correction Bound, *J. of Complexity*, **13**, 180-193, 1997.

[T86] M. Trummer, An Efficient Implementation of a Conformal Mapping Method Using the Szegõ Kernel, *SIAM J. on Numerical Analysis*, **23**, 853-872, 1986.

[Z79] R.Z. Zippel, Probabilistic Algorithm for Sparse Polynominals, *Proc. EUROSAM 79, Lecture Notes in Computer Science*, **72**, 216-226, Springer, Berlin, 1979.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.