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.

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

Inversion of Displacement Operators and Structured Matrices

by

Xinmao Wang

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

2003

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

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

4/8/03 — Victor Pan
Date — Chair of Examining Committee

4/8/03 — Murray Rocke
Date — Executive Officer

Professor Michael Anshel

Professor Alexei Miasnikov

Professor Victor Pan
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Acknowledgments

First, I would like to thank my adviser, Professor Victor Pan, for his consistent source of encouragement and support of all kinds. He was also generous with his suggestions and comments.

Secondly, it was a pleasure and honor to have Professor Michael Anshel and Professor Alexei Miasnikov serving in the Committee. I have appreciated their helpful comments and advice.

Moreover, I wish to thank Professors G. Baumslag, J. Dodziuk, E. Feldman, M. Fitting, S. Kaplan, R. Kossak, R. Kulkarni, R. Landsman, J. Pach, B. Randol, A. Rocha, J. Rosen, A. Vasquez and all other members of the Mathematics Program of Graduate School of CUNY.

Finally, the most of all, I owe tremendous debt of gratitude to all my family, my parents, my sister, my wife and my son, for their unconditional love, understanding and support.

Contents

1 Introduction 1

2 Definitions and Preliminaries 8

2.1 Displacement Operators 13

2.2 Operations with Structured Matrices 15

3 Inversion of Displacement Operators 18

3.1 Bilinear Expressions for Fundamental Structured Matrices . . 21

3.2 Bilinear Expressions for Confluent Type Matrices 31

3.3 Three Implications 39

4 Norms of the Inverse Displacement Operators 43

4.1 General Results 45

4.2 Some Examples 48

4.3 Decreasing the Norm $\|L^{-1}\|$ 53

5 Numerical Inversion of Structured Matrices **55**

5.1 SVD Truncation 57

5.2 Least Squares Truncation 58

5.3 Initial Approximation 59

6 Inversion of Integer Toeplitz-like Matrices **65**

6.1 The MBA Algorithm 66

6.2 p-adic Lifting 68

6.3 The Largest Invariant Factor 69

7 Rational Number Reconstruction **73**

7.1 Extended Euclidean Algorithm 76

7.2 EEA for Modified Input 79

7.3 Rational Number Reconstruction 85

Bibliography **86**

Chapter 1

Introduction

Structured matrices are omnipresent in computations for communication, sciences, and engineering (see extensive bibliography in [KS95], [KS99], and [P01]). Table 1.1 displays the four most popular classes of structured matrices. They are generalized to various other highly important matrix structures in the *displacement rank approach*, which originated from the seminal paper [KKM79]. We will next follow [P01] to outline this approach, which treats various matrix structures in a unified way, based on their association with the *displacement operators* and then focus on its most fundamental stage of the inversion of the displacement operators.

When a displacement operator L is applied to an $n \times n$ associated struc-

Toeplitz matrices $(t_{i-j})_{i,j=0}^{n-1}$	Hankel matrices $(h_{i+j})_{i,j=0}^{n-1}$
$\begin{pmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix}$	$\begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \iddots & h_n \\ \vdots & \iddots & \iddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{pmatrix}$
Vandermonde matrices $(t_i^j)_{i,j=0}^{n-1}$	**Cauchy matrices $(\frac{1}{s_i-t_j})_{i,j=0}^{n-1}$**
$\begin{pmatrix} 1 & t_0 & \cdots & t_0^{n-1} \\ 1 & t_1 & \cdots & t_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{n-1} \end{pmatrix}$	$\begin{pmatrix} \frac{1}{s_0-t_0} & \cdots & \frac{1}{s_0-t_{n-1}} \\ \frac{1}{s_1-t_0} & \cdots & \frac{1}{s_1-t_{n-1}} \\ \vdots & & \vdots \\ \frac{1}{s_{n-1}-t_0} & \cdots & \frac{1}{s_{n-1}-t_{n-1}} \end{pmatrix}$

Table 1.1: four classes of structured matrices

tured matrix M, the image $L(M)$, called the *displacement* of M, has small rank r, called the *displacement rank* of M. Therefore, the n^2 entries of the displacement $L(M)$ can be represented via fewer (say $2rn$) parameters. Such a compressed representation of $L(M)$ can be extended to the matrix M by inverting the displacement operator. This enables performing computations with the matrices M in terms of their *displacement generators* (G, H) by using much smaller computer memory and much less CPU time than with the general matrices as long as

a) the ranks of the displacements are kept small and

b) the desired output (e.g., the solution of a linear system of equations) is easily recovered at the end.

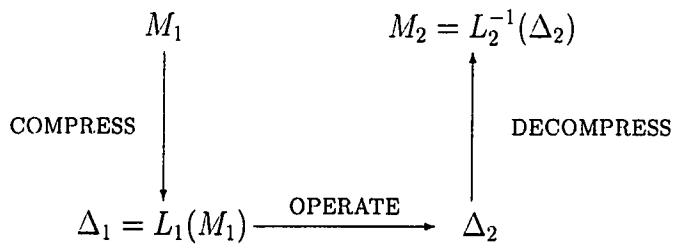

Table 1.2: flowchart for the displacement approach

The COMPRESS stage consists in choosing a short displacement generator for the input matrix M (e.g., [P92], [P93] by computing the SVD of its displacement $L(M) = U\Sigma^2V^T = GH^T$, $G = U\Sigma$, $H = V\Sigma$). Simple rules for operating with displacement generators at the OPERATE stage can be found, e.g., in Chapter 2 and [P01]. These expressions and algorithms are stated for *symbolic displacement*, where the operator L and matrix class M are not specified. Thus the rules and algorithms are unified over various classes of structured matrices. Application of these rules to various computations with structured matrices (such as the computation of short displacement generators for the inverses or for the bases of the null spaces) yielded effective algorithms, which also unified over various classes of structured matrices and are superfast, that is, run in $O(n \log^d n)$ time for $d \leq 3$, versus the orders of n^3 time in Gaussian elimination and n^2 time in various fast algorithms. Furthermore, in [P90] the *displacement transformation*

was proposed as a means for extending any successful algorithm available for one class of structured matrices to other classes, and sample transformations among the four classes of matrices of Hankel, Toeplitz, Vandermonde, and Cauchy types were displayed. This approach was pushed forward extensively, yielding effective practical algorithms [H95], [GKO95], [KO96], [G98], [G99]. On the other hand, the DECOMPRESS stage never had a systematic treatment it deserves, and the entire approach hinged on a few *ad hoc* formulas scattered in [KKM79], [AG91], [GO94] and [BP94]. Particularly valuable were the important applications using rectangular structured matrices and singular displacement operators. Insufficient development of the DECOMPRESS stage was, of course, a serious impediment for the users (who can be easily lost in the rapidly expanding area of the displacement rank methods) but also for the algorithm designers.

In Chapter 3, we specify bilinear expressions of structured matrices via their displacements covering the most popular classes of structured matrices. We treat the general case of rectangular matrices M and possibly singular operators L whose inversion on the orthogonal complement of their null spaces we extended by using the first or the last row and/or column of M (see Examples 3.6, 3.9, 3.11, 3.13, and 3.18). All this should serve as a

so far missing foundation for the DECOMPRESS stage of the displacement rank approach. Because of the high importance of the approach, our work should inevitably have substantial practical impact on the computations with structured matrices.

In Chapter 4, we extend our work to estimating the norm $\|L^{-1}\|$ of the inverse displacement operator, which is a critical numerical parameter for structured matrices. For example, whenever the solution of a linear system $M\mathbf{x} = \mathbf{b}$ is recovered from the displacement $L(M^{-1})$ computed numerically, the output errors are proportional to $\|L^{-1}\|$. In another example, a structured matrix is inverted by Newton's iteration, and the COMPRESS stage is recursively applied in each iterative step [P92], [P01]. The convergence rate of the process and even the convergence itself critically depend on the residual norm $r_i = \|I - X_i M\|$ where X_i is an approximation to M^{-1}, and is not computed explicitly but is represented by its compressed displacement. Then again, the residual norm r_i is proportional to $\|L^{-1}\|$, so the convergence is faster where $\|L^{-1}\|$ is smaller. Some upper estimates for the norm $\|L^{-1}\|$ for the most used displacement operators L have been obtained in [P92], [P93], [PRW00], [PRW01], [PKRCa]. In this chapter we use distinct techniques to obtain tighter upper and lower estimates.

In Chapter 5, we use Newton's iteration to compute numerically the inverse of a structured matrix because of the strong numerical stability, local quadratic convergence, and convenience for parallel implementation. Each iteration can be performed in $O(n \log n)$ flops. Since each iteration increases the displacement rank, we have to preserve the matrix structure. We propose two methods to control the growth of displacement rank. The convergence of the iteration strongly depends on the choice of good initial approximations, which we can compute by combining the preconditioned conjugate gradient method and the homotopy method.

In Chapter 6, we exploit the structure of Toeplitz-like matrices and use the current fastest MBA algorithm [M74], [M80], [BA80] to compute the inverse of a integer Toeplitz-like matrix M modulo a prime p. Then we use p-adic lifting to compute the inverse modulo a prime power q large enough. Finally we compute the largest Smith factor of M probabilistically by using rational number reconstruction.

In Chapter 7, we accelerate the known algorithms for computing a selected entry of the extended Euclidean algorithm for integers. The acceleration is from quadratic to nearly linear time, matching the known complexity bound for integer GCD, which our algorithm computes as a special case.

As a consequence, we accelerate the rational number reconstruction and the computation of the largest Smith factor of M.

Chapter 2

Definitions and Preliminaries

Let us start with some definitions and simple basic results (cf. [P01] on a more detailed and systematic exposition). We will assume computation in an arbitrary field $\mathbb{F}$, which in particular will cover computations in the fields of complex, real, or rational numbers ($\mathbb{C}$, $\mathbb{R}$, or $\mathbb{Q}$).

- $M \in \mathbb{F}^{m \times n}$ denotes an $m \times n$ matrix with the entries in the field $\mathbb{F}$.
- M^T, $\mathbf{v}^T$ are the *transposes* of a matrix M and a vector $\mathbf{v}$, respectively. M^H is the *Hermitian (conjugate) transpose* of M. M^{-T} is the transpose of M^{-1}, that is, $M^{-T} = (M^{-1})^T = (M^T)^{-1}$. M^{-H} is the Hermitian transpose of M^{-1}.

- $\mathbf{t}^{-1} = (t_i^{-1})_{1 \le i \le k}$ where $\mathbf{t} = (t_i)_{1 \le i \le k} \in \mathbb{F}^{k \times 1}$.

- $(M_1, \ldots, M_n)$ is the $1 \times n$ block matrix with the blocks $M_1, \ldots, M_n$.

- $D(\mathbf{v}) = \mathrm{diag}(\mathbf{v}) = \begin{pmatrix} v_1 & & \\ & \ddots & \\ & & v_n \end{pmatrix}$ is the $n \times n$ *diagonal* matrix where $\mathbf{v} = (v_i)_{1 \le i \le n}$. (Here and hereafter, blank space in matrix representation is assumed to be filled with the zeros.)

- $\mathbf{e}_i$ is the i-th coordinate vector, having its i-th coordinate 1 and all other coordinates 0, so that $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$.

- $\mathbf{1} = (1, \ldots, 1)^T$.

- $I = I_n = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} = (\mathbf{e}_1, \ldots, \mathbf{e}_n)$ denotes the $n \times n$ *identity* matrix.

- 0_n denotes the $n \times n$ null matrix.

- $J = J_n = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix} = (\mathbf{e}_n, \ldots, \mathbf{e}_1)$ denotes the $n \times n$ *reflection* matrix.

- $Z_f = \begin{pmatrix} & & f \\ & I_{n-1} & \\ & & \end{pmatrix} = (\mathbf{e}_2, \ldots, \mathbf{e}_n, f\mathbf{e}_1)$ is the $n \times n$ *unit f-circulant* matrix. $Z = Z_0$ is the $n \times n$ unit lower triangular Toeplitz matrix.

- For a vector $\mathbf{v} = (v_1, \ldots, v_m)^T$, write $Z_{f,m,n}(\mathbf{v}) = (z_{i,j})_{\substack{1 \le i \le m, \\ 1 \le j \le n}}$

$$z_{i,j} = \begin{cases} v_{i-j+1} & \text{if } i \ge j, \\ f^k v_{m-l} & \text{if } j - i - 1 = km + l,\ 0 \le l \le m-1, k \ge 0. \end{cases}$$

$Z_{f,m,n}(\mathbf{v})$ is the $m \times n$ f-circulant matrix with the first column $\mathbf{v}$. $Z_f(\mathbf{v}) = \sum_{i=1}^{m} v_i Z_f^{i-1} = Z_{f,m,m}(\mathbf{v})$. $Z(\mathbf{v}) = Z_0(\mathbf{v})$.

- $V_{m,n}(\mathbf{x}) = \left(x_i^{j-1}\right)_{\substack{1 \le i \le m \\ 1 \le j \le n}}$ is the $m \times n$ Vandermonde matrix defined by its second column vector $\mathbf{x} = (x_i)_{1 \le i \le m}$. $V(\mathbf{x}) = V_{m,m}(\mathbf{x})$.

- For natural numbers m, n, an $m \times m$ matrix P, and an m-dimensional column vector $\mathbf{v}$, the $m \times n$ *Krylov matrix*

$$K_{m,n}(P, \mathbf{v}) = (\mathbf{v}, P\mathbf{v}, \ldots, P^{n-1}\mathbf{v}).$$

- ω_n is a primitive n-th root of 1 (that is, $\omega_n^n = 1$, $\omega_n^s \neq 1$, $\forall s = 1, 2, \ldots, n-1$); e.g., $\omega_n = e^{2\pi\sqrt{-1}/n}$ in the complex number field $\mathbb{C}$.

- $\mathbf{w}_n = (\omega_n^{i-1})_{1 \le i \le n}$ is the vector of all n-th roots of 1.

- $\Omega_n = \left(\omega_n^{(i-1)(j-1)}\right)_{\substack{1 \le i \le n \\ 1 \le j \le n}}$ is the $n \times n$ matrix of the *discrete Fourier transform (DFT).*

- *The discrete Fourier transform* of a vector $\mathbf{v}$ of dimension n is the vector $DFT(\mathbf{v}) = \Omega_n \mathbf{v}$.

- $\lceil x \rceil$ and $\lfloor x \rfloor$ denote two integers closest to a real x such that $\lfloor x \rfloor \le x \le \lceil x \rceil$. $\{x\} = x - \lfloor x \rfloor$.

- $m \mid n$ means that integer m divides integer n, $m \nmid n$ means the opposite.
- $\gcd(m,n)$ and $\mathrm{lcm}(m,n)$ are the greatest common divisor and the least common multiple of two positive integers m and n, respectively.
- $m \bmod n$ is defined to be $m - n\lfloor m/n \rfloor$ for $m, n \in \mathbb{Z}$, and $n > 0$.
- For any real matrix $A = (a_{i,j})_{i,j}$, $|A| = \max_{i,j} |a_{i,j}|$.
- For any matrix A, let $\sigma_i(A)$ be its i-th largest singular value if $i \leq \mathrm{rank}(A)$, and let $\sigma_i(A) = 0$ if $i > \mathrm{rank}(A)$.
- For any $n \times n$ nonsingular matrix A, $\kappa(A) = \sigma_1(A)/\sigma_n(A)$ is the *condition number* of A.
- For any $n \times n$ matrix A, let $\mathrm{Spectrum}(A) = \{\lambda_1(A), \ldots, \lambda_n(A)\}$ be the set of all the eigenvalues of A (we repeat m times any eigenvalue having algebraic multiplicity m).

The following simple results can be easily verified.

Theorem 2.1. *$J^2 = I$, $J\mathbf{v} = (v_{n+1-i})_{1 \leq i \leq n}$, $JD(\mathbf{v})J = D(J\mathbf{v})$, for any vector $\mathbf{v} = (v_i)_{1 \leq i \leq n}$.*

Theorem 2.2. *For the $n \times n$ matrix Z_e and any scalar e, we have $Z_e^n = eI$, $Z_e^T = JZ_eJ$. For $e \neq 0$, we have $Z_e^{-1} = Z_{1/e}^T$.*

Theorem 2.3. *The Krylov matrix $K_{m,n}(P, \mathbf{v})$ turns into*

a) the $m \times n$ f-circulant matrix $Z_{f,m,n}(\mathbf{v})$ when $P = Z_f$,

b) $JZ_{f,m,n}(J\mathbf{v})$ when $P = Z_f^T$,

c) the product $D(\mathbf{v})V_{m,n}(P\mathbf{1})$ of the diagonal matrix $D(\mathbf{v})$ and the Vandermonde matrix $V_{m,n}(P\mathbf{1})$ when P is a diagonal matrix; $D(\mathbf{v}) = I_m$ when $\mathbf{v} = \mathbf{1}$.

Theorem 2.4. *[CPW74]. For the $n \times n$ matrix Z_e and any scalar $e \neq 0$, let*

$$V = V(\mathbf{t}), \quad V^{-1} = \frac{1}{n}V(\mathbf{t}^{-1})^T, \quad D = D(\mathbf{t}), \quad \mathbf{t} = (t_i)_{1 \le i \le n}$$

and let $t_1, \ldots, t_n$ be all the n-th roots of e. Then we have $Z_e = V^{-1}DV$. Furthermore, given any n-th root t of e, we may choose $t_i = t\omega_n^{i-1}$, and then

$$V = \Omega_n \operatorname{diag}\left(t^{i-1}\right)_{1 \le i \le n}, \quad V^{-1} = \frac{1}{n}\operatorname{diag}\left(t^{1-i}\right)_{1 \le i \le n}\overline{\Omega}_n, \quad D = t\mathbf{w}_n,$$

where each entry of the matrix $\overline{\Omega}_n$ is the complex conjugate of the respective entry of the matrix Ω_n.

Theorem 2.5. *$O(n \log n)$ flops are sufficient to multiply by a vector the matrices V and V^{-1} of Theorem 2.3 as well as the $n \times n$ Vandermonde matrix $V((\mathbf{t} + s\mathbf{1})^{-1})^T$ for any scalar s and for the vector $\mathbf{t}$ of the n-th roots of e (as in Theorem 2.4).*

Proof. Let $\mathbf{v} = (v_i)_i$, $\mathbf{u} = (u_k)_k$. Then $V(\mathbf{u})\mathbf{v}$ is the vector of the values of a polynomial $v(x) = \sum_{1 \le i \le n} v_i x^{i-1}$ at the nodes $x = u_k$, for $k = 1, \ldots, n$. For the node vectors $\mathbf{u} = \mathbf{t}$ and $\mathbf{u} = \mathbf{t}^{-1}$, the evaluation of a polynomial of degree $n-1$ is reduced to generalized (scaled) discrete Fourier transform, which uses $O(n \log n)$ flops [P01]. The transition to the node vector $\mathbf{t} + s\mathbf{1}$ is by the Taylor's shift of variable, which uses $O(n \log n)$ flops too (cf., e.g., [P01]). Further transition to the node vector $(\mathbf{t} + s\mathbf{1})^{-1}$ is by the reversion of the coefficients of the polynomial $v(x)$ (that is, the transition to the polynomial $x^n v(1/x)$). Finally, the multiplication by a vector requires as many flops for a nonsingular matrix as for its transpose [PSD70]. □

2.1 Displacement Operators

The modern study of structured matrices relies on their association with *displacement linear operators of Sylvester type*, $L = \nabla_{A,B}$,

$$\nabla_{A,B}(M) = AM - MB, \tag{2.1}$$

and *Stein type*, $L = \Delta_{A,B}$,

$$\Delta_{A,B}(M) = M - AMB, \tag{2.2}$$

where A, B are two fixed *operator matrices.* The image $L(M)$ is called the *L-displacement* of a matrix M or just its *displacement.* Suppose we have

$$L(M) = GH^T = \sum_{k=1}^{l} \mathbf{g}_k \mathbf{h}_k^T, \tag{2.3}$$

$G = (\mathbf{g}_1, \ldots, \mathbf{g}_l)$, $H = (\mathbf{h}_1, \ldots, \mathbf{h}_l)$, and l is "small" ($l = O(1)$ or $l \ll \min(m, n)$). Then M is called a *structured matrix with L-generator* (G, H) *of length* l. Hereafter, we confine our study to structured $m \times n$ matrices M with L-generators (G, H) of length l, for $L = \nabla_{A,B}$ or $L = \Delta_{A,B}$.

The operators of these two types can be transformed easily into each other if the matrices A and/or B are nonsingular.

Theorem 2.6. $\nabla_{A,B} = A\Delta_{A^{-1},B}$ *if the matrix* A *is nonsingular, and* $\nabla_{A,B} = -\Delta_{A,B^{-1}}B$ *if the matrix* B *is nonsingular.*

For any matrix $M = (\mathbf{m}_1, \mathbf{m}_2, \ldots)$, we may represent it by a vector

$$\overrightarrow{M} = \begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \end{pmatrix}.$$

Then the displacement linear operations can be written as

$$\overrightarrow{\nabla_{A,B}(M)} = (I \otimes A - B^T \otimes I)\overrightarrow{M}, \tag{2.4}$$

$$\overrightarrow{\Delta_{A,B}(M)} = (I - B^T \otimes A)\overrightarrow{M}, \tag{2.5}$$

where $\otimes$ is the *Kronecker product.*

Definition 2.7. A linear operator L is *nonsingular* if the matrix equation $L(M) = 0$ implies that $M = 0$.

Theorem 2.8. *$\nabla_{A,B}$ is nonsingular if and only if $\lambda_i(A) \neq \lambda_j(B)$ for all pairs of eigenvalues $(\lambda_i(A), \lambda_j(B))$; $\Delta_{A,B}$ is nonsingular if and only if $\lambda_i(A)\lambda_j(B) \neq 1$ for all pairs $(\lambda_i(A), \lambda_j(B))$.*

Proof. $\nabla_{A,B}$ is nonsingular if and only if the matrix $I \otimes A - B^T \otimes I$ is nonsingular; $\Delta_{A,B}$ is nonsingular if and only if the matrix $I - B^T \otimes A$ is nonsingular. Note that

$$\text{Spectrum}(I \otimes A - B^T \otimes I) = \{\lambda_i(A) - \lambda_j(B) \mid \text{for all } i, j\},$$

$$\text{Spectrum}(I - B^T \otimes A) = \{1 - \lambda_i(A)\lambda_j(B) \mid \text{for all } i, j\}. \qquad \square$$

Corollary 2.9. *If the operator $\nabla_{A,B}$ is nonsingular, then at least one of the operator matrices A and B is nonsingular.*

The above corollary guarantees any nonsingular Sylvester type operator can be transformed into a nonsingular Stein type operator, but not vice versa.

2.2 Operations with Structured Matrices

The next simple results relate the basic operations with matrices to operations with their displacements (cf. [P01]).

Theorem 2.10. *For two structured matrices M, N of the same type, any scalars α, β, we have*

$$\nabla_{A,B}(\alpha M + \beta N) = \alpha \nabla_{A,B}(M) + \beta \nabla_{A,B}(N),$$

$$\Delta_{A,B}(\alpha M + \beta N) = \alpha \Delta_{A,B}(M) + \beta \Delta_{A,B}(N).$$

Theorem 2.11 (Chain Rule). *For two structured matrices M, N of compatible type, we have*

$$\nabla_{A,C}(MN) = M\nabla_{B,C}(N) + \nabla_{A,B}(M)N,$$

$$\nabla_{A,C}(MN) = AM\Delta_{B,C}(N) - \Delta_{A,B}(M)NC,$$

$$\Delta_{A,C}(MN) = M\Delta_{B,C}(N) - \nabla_{A,B}(M)NC,$$

$$\Delta_{A,C}(MN) = AM\nabla_{B,C}(N) + \Delta_{A,B}(M)N.$$

Theorem 2.12. *For any nonsingular structured matrix M, we have*

$$\nabla_{B,A}(M^{-1}) = -M^{-1}\nabla_{A,B}(M)M^{-1},$$

$$\operatorname{rank}(\Delta_{B,A}(M^{-1})) = \operatorname{rank}(\Delta_{A,B}(M)).$$

Theorem 2.13. *For any structured matrix M, we have*

$$\nabla_{A,B}(M^T) = -\nabla_{B^T,A^T}(M)^T,$$

$$\Delta_{A,B}(M^T) = \Delta_{B^T,A^T}(M)^T.$$

Theorem 2.14. *For any structured matrix M, nonsingular matrices P and Q, write $\hat{A} = PAP^{-1}$, $\hat{B} = Q^{-1}BQ$, $\hat{M} = PMQ$, then we have*

$$\nabla_{\hat{A},\hat{B}}(\hat{M}) = P\nabla_{A,B}(M)Q,$$

$$\Delta_{\hat{A},\hat{B}}(\hat{M}) = P\Delta_{A,B}(M)Q.$$

Summary. The linear combination, product, inversion, transpose, Smith transformation of structured matrices (of compatible type) is again a structured matrix (maybe of different type).

Chapter 3

Inversion of Displacement Operators

First, let us give some general results. Obviously M can be get from (2.4), (2.5),

$$\overrightarrow{M} = (I \otimes A - B^T \otimes I)^{-1} \overrightarrow{\nabla_{A,B}(M)}, \tag{3.1}$$

$$\overrightarrow{M} = (I - B^T \otimes A)^{-1} \overrightarrow{\Delta_{A,B}(M)}. \tag{3.2}$$

But this is not our "COMPRESS $\longrightarrow$ OPERATE $\longrightarrow$ DECOMPRESS" displacement approach. The next simple theorem is fundamental for our task of obtaining explicit expressions for a matrix M via its displacement.

Theorem 3.1. *[GO92], [W93], [PRW00], [PRW01]. For a structured matrix M associated with Stein type operator $\Delta_{A,B}$, and for all natural numbers k, we have*

$$M = A^k M B^k + \sum_{i=0}^{k-1} A^i \Delta_{A,B}(M) B^i.$$

Corollary 3.2.

$$M(I - aB^k) = \left(\sum_{i=0}^{k-1} A^i \Delta_{A,B}(M) B^i \right) \text{ if } A^k = aI,$$

$$(I - bA^k)M = \left(\sum_{i=0}^{k-1} A^i \Delta_{A,B}(M) B^i \right) \text{ if } B^k = bI.$$

Theorem 3.1 and Corollary 3.2 enable simple expressions of a matrix M via its displacement $\Delta_{A,B}(M)$, provided that $A^k = cI$ and/or $B^k = cI$ for a scalar c. We will next specify such expressions via the generators of M.

Theorem 3.3. *For an $m \times n$ structured matrix M of equation (2.3) and $L = \Delta_{A,B}$, we have*

$$M = A^k M B^k + \sum_{j=1}^{l} K_{m,k}(A, \mathbf{g}_j) K_{n,k}(B^T, \mathbf{h}_j)^T. \tag{3.3}$$

Similarly, for structured matrix associated with Sylvester type operator, by combining Corollary 2.9, Theorems 2.6 and 3.1, we obtain the next result.

Corollary 3.4. *For structured matrix M associated with Sylvester type op-*

erator $\nabla_{A,B}$, *and for all natural number* k, *we have*

$$M = A^{-k}MB^k + \sum_{i=0}^{k-1} A^{-i-1}\nabla_{A,B}(M)B^i \text{ if } A \text{ is nonsingular,}$$

$$M = A^kMB^{-k} - \sum_{i=0}^{k-1} A^i\nabla_{A,B}(M)B^{-i-1} \text{ if } B \text{ is nonsingular.}$$

Theorem 3.5. *For an* $m \times n$ *structured matrix* M *of equation (2.3) and* $L = \nabla_{A,B}$, *we have*

$$M = \begin{cases} A^{-k-1}MB^k + A^{-1}\sum_{j=1}^{l} K_{m,k}(A^{-1}, \mathbf{g}_j)K_{n,k}(B^T, \mathbf{h}_j)^T \text{ if } \det(A) \neq 0, \\ A^kMB^{-k-1} - \sum_{j=1}^{l} K_{m,k}(A, \mathbf{g}_j)K_{n,k}(B^{-T}, \mathbf{h}_j)^TB^{-1} \text{ if } \det(B) \neq 0. \end{cases} \tag{3.4}$$

Throughout this chapter, let A, B, and M be three matrices over a field $\mathbb{F}$ having sizes $m \times m$, $n \times n$, and $m \times n$, respectively; and let the displacement

$$L(M) = GH^T = \sum_{1 \le j \le l} \mathbf{g}_j\mathbf{h}_j^T,$$

$$G = (\mathbf{g}_1, \ldots, \mathbf{g}_l) = (g_{i,j})_{\substack{1 \le i \le m, \\ 1 \le j \le l}} \quad H = (\mathbf{h}_1, \ldots, \mathbf{h}_l) = (l_{i,j})_{\substack{1 \le i \le n. \\ 1 \le j \le l}}$$

3.1 Bilinear Expressions for Fundamental Structured Matrices

In this section, we will give expressions of a structured matrix M via its displacements $L(M)$ where $L = \Delta_{A,B}$ and $L = \nabla_{A,B}$ for some commonly used operator matrices A and B,

$$A,\ B \in \{Z_e, Z_e^T, D(\mathbf{v}) \mid \text{ for all scalars } e \text{ and vectors } \mathbf{v}\}.$$

Example 3.6. $L = \Delta_{A,B}$, $A = Z_e$, $B = Z_f$. Such operators L are of Stein type customarily associated with Hankel-like matrices [KKM79], [P90], [BP94]. Note that $A^m = eI_m$, $B^n = fI_n$. We will start with the two special cases where $e = 0$ and/or $f = 0$, then will supply the expressions in the cases where $eB^m \neq I$ or $fA^n \neq I$ (for $m = n$, this means that $ef \neq 1$), and finally will cover all choices of e and f.

1. $e = 0$.

$$M = \sum_{j=1}^{l} K_{m,m}(Z, \mathbf{g}_j) K_{n,m}(Z_f^T, \mathbf{h}_j)^T = \sum_{j=1}^{l} Z(\mathbf{g}_j) Z_{f,n,m}(J\mathbf{h}_j)^T J$$

$$= \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & & & \\ g_{2,j} & g_{1,j} & & \\ \vdots & \vdots & \ddots & \\ g_{m,j} & g_{m-1,j} & \cdots & g_{1,j} \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & fh_{1,j} \\ \vdots & \iddots & fh_{1,j} & fh_{2,j} \\ \vdots & \iddots & \iddots & \vdots \end{pmatrix}$$

2. $f = 0$.

$$M = \sum_{j=1}^{l} K_{m,n}(Z_e, \mathbf{g}_j) K_{n,n}(Z^T, \mathbf{h}_j)^T = \sum_{j=1}^{l} Z_{e,m,n}(\mathbf{g}_j) Z(J\mathbf{h}_j)^T J$$

$$= \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & eg_{m,j} & eg_{m-1,j} & \cdots \\ g_{2,j} & g_{1,j} & eg_{m-1,j} & \ddots \\ \vdots & \vdots & \ddots & \ddots \\ g_{m,j} & g_{m-1,j} & \cdots & \cdots \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & \\ \vdots & \iddots & & \\ h_{n,j} & & & \end{pmatrix}$$

3. If the operator Δ_{Z_e,Z_f} is nonsingular, and then both matrices $I_n - eZ_f^m$ and $I_m - fZ_e^n$ are nonsingular.

$$M = \sum_{j=1}^{l} K_{m,m}(Z_e, \mathbf{g}_j) K_{n,m}(Z_f^T, \mathbf{h}_j)^T (I_n - eZ_f^m)^{-1}$$

$$= \sum_{j=1}^{l} Z_e(\mathbf{g}_j) Z_{f,n,m}(J\mathbf{h}_j)^T J (I_n - eZ_f^m)^{-1}$$

$$= \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & eg_{m,j} & \cdots & eg_{2,j} \\ g_{2,j} & g_{1,j} & \ddots & \vdots \\ \vdots & \vdots & \ddots & eg_{m,j} \\ g_{m,j} & g_{m-1,j} & \cdots & g_{1,j} \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & fh_{1,j} \\ \vdots & \iddots & fh_{1,j} & fh_{2,j} \\ \vdots & \iddots & \iddots & \vdots \end{pmatrix} (I_n - eZ_f^m)^{-1},$$

and also

$$M = (I_m - fZ_e^n)^{-1} \sum_{j=1}^{l} K_{m,n}(Z_e, \mathbf{g}_j) K_{n,n}(Z_f^T, \mathbf{h}_j)^T$$

$$= (I_m - fZ_e^n)^{-1} \sum_{j=1}^{l} Z_{e,m,n}(\mathbf{g}_j) Z_f(J\mathbf{h}_j)^T J$$

$$= (I_m - fZ_e^n)^{-1} \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & eg_{m,j} & eg_{m-1,j} & \cdots \\ g_{2,j} & g_{1,j} & eg_{m,j} & \ddots \\ \vdots & \vdots & \ddots & \ddots \\ g_{m,j} & g_{m-1,j} & \cdots & \cdots \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & fh_{1,j} \\ \vdots & \iddots & \iddots & \vdots \\ h_{n,j} & fh_{1,j} & \cdots & fh_{n,j} \end{pmatrix}.$$

4. If the operator Δ_{Z_e,Z_f} is singular, then we cannot recover M solely from its displacement. We need extra information about the matrix M. We start with the two matrix equations

$$\Delta_{Z,Z_f}(M) = \Delta_{Z_e,Z_f}(M) + \left(0_{m-1}\ ^{e}\right) MZ_f = GH^T + e\mathbf{e}_1\mathbf{e}_m^T MZ_f,$$

$$\Delta_{Z_e,Z}(M) = \Delta_{Z_e,Z_f}(M) + Z_e M \left(0_{n-1}\ ^{f}\right) = GH^T + fZ_e M\mathbf{e}_1\mathbf{e}_n^T,$$

and deduce that

$$\begin{aligned}
M &= \sum_{j=1}^{l} K_{m,m}(Z, \mathbf{g}_j) K_{n,m}(Z_f^T, \mathbf{h}_j)^T \ + \ eK_{m,m}(Z, \mathbf{e}_1) K_{n,m}(Z_f^T, Z_f^T M^T \mathbf{e}_m)^T \\
&= \sum_{j=1}^{l} Z(\mathbf{g}_j) Z_{f,n,m}(J\mathbf{h}_j)^T J \ + \ eZ_{f,n,m}(JZ_f^T M^T \mathbf{e}_m)^T J \\
&= \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & & & \\ g_{2,j} & g_{1,j} & & \\ \vdots & \vdots & \ddots & \\ g_{m,j} & g_{m-1,j} & \cdots & g_{1,j} \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & fh_{1,j} \\ \vdots & \iddots & fh_{1,j} & fh_{2,j} \\ \vdots & \iddots & \iddots & \vdots \end{pmatrix} \\
&\quad + e \begin{pmatrix} M_{m,2} & \cdots & M_{m,n} & fM_{m,1} \\ M_{m,3} & \cdots & fM_{m,1} & fM_{m,2} \\ \vdots & \iddots & fM_{m,2} & fM_{m,3} \\ \vdots & \iddots & \iddots & \vdots \end{pmatrix},
\end{aligned}$$

and

$$
\begin{aligned}
M &= \sum_{j=1}^{l} K_{m,n}(Z_e, \mathbf{g}_j) K_{n,n}(Z^T, \mathbf{h}_j)^T \; + \; f K_{m,n}(Z_e, Z_e M \mathbf{e}_1) K_{n,n}(Z^T, \mathbf{e}_n)^T \\
&= \sum_{j=1}^{l} Z_{e,m,n}(\mathbf{g}_j) Z(J\mathbf{h}_j)^T J \; + \; f Z_{e,m,n}(J Z_e M \mathbf{e}_1) J \\
&= \sum_{j=1}^{l} \begin{pmatrix} g_{1,j} & eg_{m,j} & eg_{m-1,j} & \cdots \\ g_{2,j} & g_{1,j} & eg_{m-1,j} & \ddots \\ \vdots & \vdots & \ddots & \ddots \\ g_{m,j} & g_{m-1,j} & \cdots & \cdots \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & \\ \vdots & \iddots & & \\ h_{n,j} & & & \end{pmatrix} \\
&\quad + f \begin{pmatrix} eM_{m,1} & eM_{m-1,1} & eM_{m-2,1} & \cdots \\ M_{1,1} & eM_{m,1} & eM_{m-1,1} & \ddots \\ \vdots & \vdots & \ddots & \ddots \\ M_{m-1,1} & M_{m-2,1} & \cdots & \cdots \end{pmatrix} J,
\end{aligned}
$$

where $(M_{i,1})_{1\le i\le m} = M\mathbf{e}_0$ is the first column of the matrix M and $(M_{m,j})_{1\le j\le n} = \mathbf{e}_{n-1}^T M$ is its last row.

Remark 3.7. In Example 3.6 case 3, we involve the matrices

$$
\begin{aligned}
(I_m - fZ_e^n)^{-1} &= V_e^{-1} \operatorname{diag}\left(\frac{1}{1-fs_i^n}\right)_{1\le i\le m} V_e, \\
(I_n - eZ_f^m)^{-1} &= V_f^{-1} \operatorname{diag}\left(\frac{1}{1-et_i^m}\right)_{1\le i\le n} V_f,
\end{aligned}
$$

where $V_e = \left(s_i^{j-1}\right)_{\substack{1\le i\le m,\\ 1\le j\le m}}$, $V_f = \left(t_i^{j-1}\right)_{\substack{1\le i\le n\\ 1\le j\le n}}$ are Vandermonde matrices, $s_1, \ldots, s_m$ are all the m-th roots of e, and $t_1, \ldots, t_n$ are all the n-th roots of f.

Example 3.8. $L = \Delta_{Z_e,Z_f^T}$, $L = \Delta_{Z_e^T,Z_f}$, and $L = \Delta_{Z_e^T,Z_f^T}$. Such operators of Stein type are customarily associated with Toeplitz-like and Hankel-like matrices. By Theorem 2.2, we have

$$\Delta_{Z_e,Z_f}(MJ) = \Delta_{Z_e,Z_f^T}(M)J = G(JH)^T,$$

$$\Delta_{Z_e,Z_f}(JM) = J\Delta_{Z_e^T,Z_f}(M) = (JG)H^T,$$

$$\Delta_{Z_e,Z_f}(JMJ) = J\Delta_{Z_e^T,Z_f^T}(M)J = (JG)(JH)^T.$$

The latter equations reduce the problem to the case of the operators Δ_{Z_e,Z_f} of Example 3.6.

Example 3.9. $L = \nabla_{Z_e,Z_f}$. Such operators of Sylvester type are associated with Toeplitz-like matrices.

1. If $e \neq 0$, then by Theorems 2.2 and 2.6, we have

$$\Delta_{Z_{1/e}^T,Z_f}(M) = Z_{1/e}^T\nabla_{Z_e,Z_f}(M) = (Z_{1/e}^T G)H^T.$$

Likewise, if $f \neq 0$ we have

$$\Delta_{Z_e,Z_{1/f}^T}(M) = -\nabla_{Z_e,Z_f}(M)Z_{1/f}^T = G(Z_{1/f}H)^T.$$

The latter equations immediately reduce the problem to the case of the Stein type operators $\Delta_{Z_e^T,Z_f}$ and Δ_{Z_e,Z_f^T} of Example 3.8.

2. If $e = f = 0$, then we rely on the equation

$$\Delta_{Z^T,Z}(M) = M - Z^T M Z = Z^T \nabla_{Z,Z}(M) + \left(0_{m-1}\ {}_1\right) M = (Z^T G) H^T + \mathbf{e}_m \mathbf{e}_m^T M,$$

and deduce that

$$M = \sum_{j=1}^{l} K_{m,m}(Z^T, Z^T \mathbf{g}_j) K_{n,m}(Z^T, \mathbf{h}_j)^T + J K_{m,m}(Z^T, M^T \mathbf{e}_m)^T$$
$$= J \sum_{j=1}^{l} Z(J Z^T \mathbf{g}_j) Z_{0,n,m}(J\mathbf{h}_j)^T J + J Z_{0,n,m}(J M^T \mathbf{e}_m)^T J.$$

Example 3.10. Similarly to Example 3.9, we express Hankel-like and Toeplitz-like matrices M associated with the Sylvester type operators $L = \nabla_{Z_e,Z_f^T}$, $L = \nabla_{Z_e^T,Z_f}$, and $L = \nabla_{Z_e^T,Z_f^T}$.

Example 3.11. $L = \Delta_{A,B}$, $A = D(\mathbf{v})$, $B = Z_f$. Such operators L of Stein type are associated with the matrix structure of Vandermonde type.

1. If the operator $\Delta_{D(\mathbf{v}),Z_f}$ is nonsingular, then the matrix $I_m - f D(\mathbf{v})^n$ is nonsingular,

$$M = (I_m - f D(\mathbf{v})^n)^{-1} \sum_{j=1}^{l} K_{m,n}(D(\mathbf{v}), \mathbf{g}_j) K_{n,n}(Z_f^T, \mathbf{h}_j)^T$$
$$= \operatorname{diag}\left(\frac{1}{1 - f v_i^n}\right)_{1 \le i \le m} \sum_{j=1}^{l} D(\mathbf{g}_j) V_{m,n}(\mathbf{v}) Z_f (J\mathbf{h}_j)^T J$$
$$= \sum_{j=1}^{l} \operatorname{diag}\left(\frac{g_{i,j}}{1 - f v_i^n}\right)_{1 \le i \le m} \begin{pmatrix} 1 & v_1 & \cdots & v_1^{n-1} \\ 1 & v_2 & \cdots & v_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & v_m & \cdots & v_m^{n-1} \end{pmatrix} \begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & f h_{1,j} \\ \vdots & \iddots & \iddots & \vdots \\ h_{n,j} & f h_{1,j} & \cdots & f h_{n,j} \end{pmatrix}.$$

2. To relax the non-singularity assumption, observe that

$$\Delta_{D(\mathbf{v}),Z}(M) = \Delta_{D(\mathbf{v}),Z_f}(M)+fD(\mathbf{v})M\begin{pmatrix} 0_{n-1} & f \end{pmatrix} = GH^T+fD(\mathbf{v})M\mathbf{e}_1\mathbf{e}_n^T.$$

Therefore,

$$\begin{aligned} M &= \sum_{j=1}^{l} K_{m,n}(D(\mathbf{v}),\mathbf{g}_j)K_{n,n}(Z^T,\mathbf{h}_j)^T \;+\; fD(\mathbf{v})K_{m,n}(D(\mathbf{v}),M\mathbf{e}_1)K_{n,n}(Z^T,\mathbf{e}_n)^T \\ &= \sum_{j=1}^{l} D(\mathbf{g}_j)V_{m,n}(\mathbf{v})Z(J\mathbf{h}_j)^TJ \;+\; fD(\mathbf{v})D(M\mathbf{e}_1)V_{m,n}(\mathbf{v})J \\ &= \sum_{j=1}^{l} D(\mathbf{g}_j)\begin{pmatrix} 1 & v_1 & \cdots & v_1^{n-1} \\ 1 & v_2 & \cdots & v_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & v_m & \cdots & v_m^{n-1} \end{pmatrix}\begin{pmatrix} h_{1,j} & h_{2,j} & \cdots & h_{n,j} \\ h_{2,j} & \cdots & h_{n,j} & \\ \vdots & \iddots & & \\ h_{n,j} & & & \end{pmatrix} \\ &\quad + fD(M\mathbf{e}_1)\begin{pmatrix} 1 & v_1 & \cdots & v_1^{n-1} \\ 1 & v_2 & \cdots & v_2^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & v_m & \cdots & v_m^{n-1} \end{pmatrix}J. \end{aligned}$$

Example 3.12. $L = \Delta_{D(\mathbf{v}),Z_f^T}$, $L = \Delta_{Z_e,D(\mathbf{v})}$, and $L = \Delta_{Z_e^T,D(\mathbf{v})}$. Such operators of Stein type cover matrix structures of Vandermonde type, distinct from the ones covered by the operator $\Delta_{D(\mathbf{v}),Z_f}$. By Theorems 2.2 and 2.13, we have

$$\Delta_{D(\mathbf{v}),Z_f}(MJ) = \Delta_{D(\mathbf{v}),Z_f^T}(M)J = G(JH)^T,$$

$$\Delta_{D(\mathbf{v}),Z_e}(M^TJ) = (\Delta_{Z_e,D(\mathbf{v})}(M))^TJ = H(JG)^T,$$

$$\Delta_{D(\mathbf{v}),Z_e}(M^T) = (\Delta_{Z_e^T,D(\mathbf{v})}(M))^T = HG^T.$$

The latter equations reduce the problem to the case of the operators of Example 3.11.

Example 3.13. $L = \nabla_{D(\mathbf{v}),Z_f}$. Such operators of Sylvester type cover matrices with Vandermonde type structure.

1. If $D(\mathbf{v})$ is nonsingular, then by Theorem 2.6 we have

$$\Delta_{D(\mathbf{v})^{-1},Z_{1/f}}(M) = D(\mathbf{v})^{-1}\nabla_{D(\mathbf{v}),Z_f}(M) = (D(\mathbf{v})^{-1}G)H^T,$$

and the problem is reduced to the case of the Stein type operator $\Delta_{D(\mathbf{v}),Z_f}$ of Example 3.11.

Likewise, if $f \neq 0$, we deduce that

$$\Delta_{D(\mathbf{v}),Z_{1/f}^T}(M) = -\nabla_{D(\mathbf{v}),Z_f}(M)Z_{1/f}^T = G(Z_{1/f}H)^T,$$

and the problem is reduced to the case of the Stein type operator $\Delta_{D(\mathbf{v}),Z_f^T}$ of Example 3.12.

2. If $f = 0$, then we combine Theorem 3.3 with the equation

$$\Delta_{D(\mathbf{v}),Z^T}(M) = M - D(\mathbf{v})MZ^T = M - (\nabla_{D(\mathbf{v}),Z}(M) + MZ)Z^T = -G(ZH)^T + M\mathbf{e}_1\mathbf{e}_1^T,$$

and deduce that

$$M = -\sum_{j=1}^{l} K_{m,n}(D(\mathbf{v}), \mathbf{g}_j) K_{n,n}(Z, Z\mathbf{h}_j)^T + K_{m,n}(D(\mathbf{v}), M\mathbf{e}_1)$$
$$= -\sum_{j=1}^{l} D(\mathbf{g}_j) V_{m,n}(\mathbf{v}) Z (JZ\mathbf{h}_j)^T J + D(M\mathbf{e}_1) V_{m,n}(\mathbf{v}).$$

Example 3.14. $L = \nabla_{D(\mathbf{v}),Z_f^T}$, $L = \nabla_{Z_e,D(\mathbf{v})}$, and $L = \nabla_{Z_e^T,D(\mathbf{v})}$. Such operators of Sylvester type are associated with structured matrices of Vandermonde type.

By Theorems 2.2 and 2.13, we have

$$\nabla_{D(\mathbf{v}),Z_f}(MJ) = \nabla_{D(\mathbf{v}),Z_f^T}(M)J = G(JH)^T,$$

$$\nabla_{D(\mathbf{v}),Z_e}(M^T J) = -(\nabla_{Z_e,D(\mathbf{v})}(M))^T J = -H(JG)^T,$$

$$\nabla_{D(\mathbf{v}),Z_e}(M^T) = -(\nabla_{Z_e^T,D(\mathbf{v})}(M))^T = -HG^T.$$

The latter equations reduce the problem to the case of the operator $\nabla_{D(\mathbf{v}),Z_f}$ of Example 3.13.

Example 3.15. $L = \nabla_{A,B}$, $A = D(\mathbf{s})$, $B = D(\mathbf{t})$. Such operators L of Sylvester type are associated with Cauchy-like matrices M. Comparing the (i,j)-th entries of the matrices on both sides of equation (2.1), we obtain that

$$s_i M_{i,j} - M_{i,j} t_j = \sum_{k=1}^{l} g_{i,k} h_{j,k},$$

that is,

$$M_{i,j} = \begin{cases} \dfrac{1}{s_i - t_j} \sum_{k=1}^{l} g_{i,k} h_{j,k}, & \text{if } s_i \neq t_j, \\ \text{any value}, & \text{if } s_i = t_j. \end{cases}$$

Example 3.16. $L = \Delta_{A,B}$, $A = D(\mathbf{s})$, $B = D(\mathbf{t})$. Such operators are of the same type as in Example 3.15. Comparing the (i,j)-th entries of the matrices on both sides of equation (3.2), we obtain that

$$M_{i,j} - s_i M_{i,j} t_j = \sum_{k=1}^{l} g_{i,k} h_{j,k},$$

that is,

$$M_{i,j} = \begin{cases} \dfrac{1}{1 - s_i t_j} \sum_{k=1}^{l} g_{i,k} h_{j,k}, & \text{if } s_i t_j \neq 1, \\ \text{any value}, & \text{if } s_i t_j = 1. \end{cases}$$

Remark 3.17. In [P01], the solution of the tangential Nevanlinna-Pick problems was reduced to computations with matrices associated with the operator $\nabla_{Z_e^T, -Z_f}$ for a pair of scalars e and f, $e \neq f$. (Such matrices are called skew-Hankel-like in [P01].) The reduction relies on a variant of transformation techniques of [P90]. The inversion formulas of Example 3.8 are immediately extended to this case. (More generally, all presented expressions for the inversion of the operators $\nabla_{A,B}$ and $\Delta_{A,B}$ can be immediately extended to operators $\nabla_{aA,bB}$ and $\Delta_{aA,bB}$ for any pair of nonzero scalars a and b.)

Such an extension enables application of the divide-and-conquer algorithms of [OP98], [P99], [P00], and [P01] to achieve the solution of the important rational interpolation problems by using $O(N \log^2 N)$ field operations (N being the input size) versus $O(N \log^3 N)$ of [OP98] and order of N^2 in all preceding works. The new acceleration relies on our algebraic techniques but results in numerically stable transformation algorithms [P01].

3.2 Bilinear Expressions for Confluent Type Matrices

In this section, let us consider operators $L = \nabla_{A,B}$ where $A = \lambda I_m + Z_e$, $B = \mu I_n + Z_f$. These operators cover the confluent matrices involved in the confluent tangential Nevanlinna-Pick problem of rational interpolation [BGR90], [GO94a], [OS99], [OS00], [P01]. We observe immediately that $\nabla_{A,B} = \nabla_{(\lambda-\mu)I_m+Z_e,Z_f} = \nabla_{Z_e,(\mu-\lambda)I_n+Z_f}$. We will consider separately several cases depending on the values e, f, and $\lambda - \mu$.

Case 1. $e = f = 0$.

Assume that $\lambda \neq \mu$. Otherwise see Example 3.9 part 2. From the equation

$$\nabla_{A,B}(M) = ((\lambda - \mu)I_m + Z)M - MZ = GH^T,$$

we deduce that

$$\begin{aligned} M &= \sum_{j=0}^{n-1}((\lambda-\mu)I_m + Z)^{-j-1}GH^TZ^j \\ &= \sum_{j=0}^{n-1}\sum_{i=0}^{m-1}\binom{-j-1}{i}(\lambda-\mu)^{-j-1-i}Z^iGH^TZ^j \\ &= \sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\frac{(-1)^i(i+j)!}{i!(\lambda-\mu)^{i+j+1}j!}Z^iGH^TZ^j \\ &= \sum_{k=1}^{l}K_{m,m}(Z,\mathbf{g}_k)\Theta_0(\lambda-\mu)K_{n,n}(Z^T,\mathbf{h}_k)^T \\ &= \sum_{k=1}^{l}Z(\mathbf{g}_k)\Theta_0(\lambda-\mu)Z(J\mathbf{h}_k)^TJ, \end{aligned}$$

where $\Theta_0(s) = \left(\frac{(-1)^{i-1}(i+j-2)!}{(i-1)!s^{i+j-1}(j-1)!}\right)_{\substack{1\le i\le m\\ 1\le j\le n}}$ is an $m \times n$ matrix. $\Theta_0(s) = \operatorname{diag}\left(\frac{(-1)^{i-1}}{(i-1)!}\right)_{1\le i\le m} H \operatorname{diag}\left(\frac{1}{(j-1)!}\right)_{1\le j\le n}$, $H = \left(\frac{(i+j-2)!}{s^{i+j-1}}\right)_{\substack{1\le i\le m\\ 1\le j\le n}}$ is an $m \times n$ Hankel matrix.

Case 2. $e \neq 0$, $f = 0$.

Subcase 2.1. $(\mu - \lambda)^m \neq e$. Write $V = V(\mathbf{t})$, from the equation

$$\nabla_{A,B}(M) = ((\lambda - \mu)I_m + Z_e)M - MZ = GH^T$$

we deduce that

$$
\begin{aligned}
M &= \sum_{j=0}^{n-1}((\lambda-\mu)I_m + Z_e)^{-j-1}GH^TZ^j \\
&= \sum_{j=0}^{n-1} V^{-1}((\lambda-\mu)I_m + D)^{-j-1}VGH^TZ^j \\
&= \sum_{k=1}^{l} V^{-1}((\lambda-\mu)I_m + D)^{-1}K_{m,n}(((\lambda-\mu)I_m + D)^{-1}, V\mathbf{g}_k)K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} V^{-1}\,\mathrm{diag}(V\mathbf{g}_k)\left((\tfrac{1}{\lambda-\mu+t_i})^j\right)_{\substack{1\le i\le m\\ 1\le j\le n}} K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l}\left(V^{-1}\sum_{r=1}^{m} g_{r,k}D^{r-1}\right)\left((\tfrac{1}{\lambda-\mu+t_i})^j\right)_{\substack{1\le i\le m\\ 1\le j\le n}} K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l}\left(\sum_{r=1}^{m} g_{r,k}Z_e^{r-1}V^{-1}\right)\left((\tfrac{1}{\lambda-\mu+t_i})^j\right)_{\substack{1\le i\le m\\ 1\le j\le n}} K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} K_{m,m}(Z_e, \mathbf{g}_k)V^{-1}\left((\tfrac{1}{\lambda-\mu+t_i})^j\right)_{\substack{1\le i\le m\\ 1\le j\le n}} K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} K_{m,m}(Z_e, \mathbf{g}_k)\Theta_1(\lambda-\mu)K_{n,n}(Z^T, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} Z_e(\mathbf{g}_k)\Theta_1(\lambda-\mu)Z(J\mathbf{h}_k)^TJ,
\end{aligned}
$$

where $\Theta_1(s) = V^{-1}\left((\frac{1}{s+t_i})^j\right)_{\substack{1\le i\le m\\ 1\le j\le n}}$ is an $m \times n$ matrix, $\mathbf{t} = (t_i)_{1\le i\le m}$ and $t_1, \ldots, t_m$ are all the m-th roots of e. $\Theta_1(s) = \frac{1}{m}V(\mathbf{t}^{-1})^TV_{m,n}((\mathbf{t}+s\mathbf{1})^{-1})D(\mathbf{t}+s\mathbf{1})^{-1}$.

Subcase 2.2. $(\mu-\lambda)^m = e$, operator L is singular. Rely on the equation

$$((\lambda-\mu)I_m + Z)M - MZ = \nabla_{A,B}(M) + (Z - Z_e)M = GH^T - e\mathbf{e}_1\mathbf{e}_m^T M$$

we deduce that

$$\begin{aligned} M &= \sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k)\Theta_0(\lambda-\mu)K_{n,n}(Z^T, \mathbf{h}_k)^T \; - \; e\Theta_0(\lambda-\mu)K_{n,n}(Z^T, M^T\mathbf{e}_m)^T \\ &= \sum_{k=1}^{l} Z(\mathbf{g}_k)\Theta_0(\lambda-\mu)Z(J\mathbf{h}_k)^T J \; - \; e\Theta_0(\lambda-\mu)Z(JM^T\mathbf{e}_m)^T J. \end{aligned}$$

Case 3. $e = 0$, $f \neq 0$.

Subcase 3.1. $(\lambda-\mu)^n \neq f$. Write $V = V(\mathbf{t})$, from the equation

$$\nabla_{A,B}(M) = ZM - M((\mu-\lambda)I_n + Z_f) = GH^T,$$

we obtain that

$$\begin{aligned} M &= -\sum_{i=0}^{m-1} Z^i GH^T((\mu-\lambda)I_n + Z_f)^{-i-1} \\ &= -\sum_{i=0}^{m-1} Z^i GH^T V^{-1}((\mu-\lambda)I_n + D)^{-i-1}V \\ &= -\sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k)K_{n,m}(((\mu-\lambda)I_n + D)^{-1}, V^{-T}\mathbf{h}_k)^T((\mu-\lambda)I_n + D)^{-1}V \\ &= -\sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k)\left(\left(\tfrac{1}{\mu-\lambda+t_j}\right)^i\right)_{\substack{1\le i\le m \\ 1\le j\le n}} \operatorname{diag}(\mathbf{h}_k^T V^{-1})V \end{aligned}$$

$$
\begin{aligned}
&= -\sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \left((\tfrac{1}{\mu-\lambda+t_j})^i \right)_{\substack{1\le i\le m \\ 1\le j\le n}} \left(\sum_{r=1}^{n} \tfrac{1}{n} h_{k,r} D^{1-j} V \right) \\
&= -\sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \left((\tfrac{1}{\mu-\lambda+t_j})^i \right)_{\substack{1\le i\le m \\ 1\le j\le n}} \left(\sum_{r=1}^{n} \tfrac{1}{n} h_{k,r} V Z_f^{1-j} \right) \\
&= -\sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \left((\tfrac{1}{\mu-\lambda+t_j})^i \right)_{\substack{1\le i\le m \\ 1\le j\le n}} (\tfrac{1}{n} V) K_n(Z_f^{-T}, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \Theta_2(\mu-\lambda) K_{n,n}(Z_{1/f}, \mathbf{h}_k)^T \\
&= \sum_{k=1}^{l} Z(\mathbf{g}_k) \Theta_2(\mu-\lambda) Z_{1/f}(\mathbf{h}_k)^T,
\end{aligned}
$$

where $\Theta_2(s) = -\frac{1}{n} \left((\frac{1}{s+t_j})^i \right)_{\substack{1\le i\le m \\ 1\le j\le n}} V$ is an $m \times n$ matrix, $\mathbf{t} = (t_i)_{1\le i\le n}$, and $t_1, \ldots, t_n$ are all the n-th roots of f. $\Theta_2(s) = -\frac{1}{n} D(\mathbf{t} + s\mathbf{1})^{-1} V_{m,n}((\mathbf{t} + s\mathbf{1})^{-1})^T V(\mathbf{t})$.

Subcase 3.2. $(\lambda - \mu)^n = f$, operator L is singular. Rely on the equation

$$
((\lambda-\mu) I_m + Z) M - MZ = \nabla_{A,B}(M) + M(Z_f - Z) = GH^T + eM\mathbf{e}_1\mathbf{e}_n^T,
$$

we deduce that

$$
\begin{aligned}
M &= \sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \Theta_0(\lambda-\mu) K_{n,n}(Z^T, \mathbf{h}_k)^T \; + \; f K_{m,m}(Z, M\mathbf{e}_1) \Theta_0(\lambda-\mu) J \\
&= \sum_{k=1}^{l} Z(\mathbf{g}_k) \Theta_0(\lambda-\mu) Z(J\mathbf{h}_k)^T J \; + \; f Z(M\mathbf{e}_1) \Theta_0(\lambda-\mu) J.
\end{aligned}
$$

Case 4. $ef \neq 0$.

Subcase 4.1. Operator L is nonsingular. Then both matrices $I - f((\lambda - \mu)I_m + Z_e)^n$ and $I - e((\mu - \lambda)I_n + Z_f)^m$ are nonsingular. We have

$$M = Me((\mu - \lambda)I_n + Z_f)^m + \sum_{i=0}^{m-1} Z_e^{-i-1} GH^T((\mu - \lambda)I_n + Z_f)^i$$

$$M = f((\lambda - \mu)I_m + Z_e)^n M - \sum_{j=0}^{n-1}((\lambda - \mu)I_m + Z_e)^j GH^T Z_f^{-j-1}.$$

Therefore,

$$\begin{aligned}
M &= \left(\sum_{i=0}^{m-1} Z_e^{-i-1} GH^T((\mu - \lambda)I_n + Z_f)^i\right)(I_n - e((\mu - \lambda)I_n + Z_f)^m)^{-1} \\
&= \left(\sum_{i=0}^{m-1} Z_e^{-i-1} GH^T \sum_{j=0}^{n-1} \binom{i}{j}(\mu - \lambda)^{i-j} Z_f^j\right)(I_n - e((\mu - \lambda)I_n + Z_f)^m)^{-1} \\
&= \left(\sum_{i=1}^{m}\sum_{j=1}^{n} \binom{i-1}{j-1}(\mu - \lambda)^{i-j} Z_e^{-i} GH^T Z_f^{j-1}\right)(I_n - e((\mu - \lambda)I_n + Z_f)^m)^{-1} \\
&= \left(\sum_{k=1}^{l} K_{m,m}(Z_e^{-1}, Z_e^{-1}\mathbf{g}_k)\Theta_3(\mu - \lambda)K_{n,m}(Z_f^T, \mathbf{h}_k)^T\right)(I_n - e((\mu - \lambda)I_n + Z_f)^m)^{-1}, \\
&= \left(\sum_{k=1}^{l} JZ_{1/e}(JZ_e^{-1}\mathbf{g}_k)\Theta_3(\mu - \lambda)Z_{f,n,m}(J\mathbf{h}_k)^T J\right)(I_n - e((\mu - \lambda)I_n + Z_f)^m)^{-1}.
\end{aligned}$$

where $\Theta_3(s) = \left(\frac{(i-1)!s^{i-j}}{(j-1)!(i-j)!}\right)_{\substack{1\le i\le m \\ 1\le j\le i}}$ is an $m \times m$ lower triangular matrix. $\Theta_3(s) = \operatorname{diag}((i-1)!)_{1\le i\le m}\left(\frac{s^{i-j}}{(i-j)!}\right)_{\substack{1\le i\le m \\ 1\le j\le i}}\operatorname{diag}\left(\frac{1}{(j-1)!}\right)_{1\le j\le m}$. Similarly, we

obtain that

$$
\begin{aligned}
M &= -(I_m - f((\lambda-\mu)I_m + Z_e)^n)^{-1}\left(\sum_{j=0}^{n-1}((\lambda-\mu)I_m + Z_e)^j GH^T Z_f^{-j-1}\right)\\
&= -(I_m - f((\lambda-\mu)I_m + Z_e)^n)^{-1}\left(\sum_{j=0}^{n-1}\sum_{i=0}^{j}\binom{j}{i}(\lambda-\mu)^{j-i}Z_e^i GH^T Z_f^{-j-1}\right)\\
&= -(I_m - f((\lambda-\mu)I_m + Z_e)^n)^{-1}\left(\sum_{j=1}^{n}\sum_{i=1}^{j}\binom{j-1}{i-1}(\lambda-\mu)^{j-i}Z_e^{i-1} GH^T Z_f^{-j}\right)\\
&= (I_m - f((\lambda-\mu)I_m + Z_e)^n)^{-1}\left(\sum_{k=1}^{l} K_{m,n}(Z_e, \mathbf{g}_k)\Theta_4(\lambda-\mu)K_{n,n}(Z_{1/f}, Z_{1/f}\mathbf{h}_k)\right)\\
&= (I_m - f((\lambda-\mu)I_m + Z_e)^n)^{-1}\left(\sum_{k=1}^{l} Z_e(\mathbf{g}_k)\Theta_4(\lambda-\mu)Z_{1/f}(Z_{1/f}\mathbf{h}_k)\right),
\end{aligned}
$$

where $\Theta_4(s) = -\left(\frac{(j-1)!s^{j-i}}{(i-1)!(j-i)!}\right)_{\substack{1\le i\le j\\ 1\le j\le n}}$ is an $n\times n$ upper triangular matrix. $\Theta_4(s) = -\operatorname{diag}\left(\frac{1}{(i-1)!}\right)_{1\le i\le n}\left(\frac{s^{j-i}}{(j-i)!}\right)_{\substack{1\le i\le j\\ 1\le j\le n}}\operatorname{diag}\left((j-1)!\right)_{1\le j\le n}$.

Subcase 4.2. Operator L is singular. For any 4-tuple (λ, μ, e, f), we apply the equations

$$(\lambda I_m + Z_e)M - M(\mu I_n + Z) = \nabla_{A,B}(M) + M(Z_f - Z) = GH^T + fM\mathbf{e}_1\mathbf{e}_n^T,$$

where $Z^n = 0$, and

$$(\lambda I_m + Z)M - M(\mu I_n + Z_f) = \nabla_{A,B}(M) + (Z - Z_e)M = GH^T - e\mathbf{e}_1\mathbf{e}_m^T M,$$

where $Z^m = 0$, and deduce that

$$M = \sum_{k=1}^{l} K_{m,m}(Z_e, \mathbf{g}_k)\Theta_1(\lambda - \mu)K_{n,n}(Z^T, \mathbf{h}_k)^T + fK_{m,m}(Z_e, M\mathbf{e}_1)\Theta_1(\lambda - \mu)J$$
$$= \sum_{k=1}^{l} Z_e(\mathbf{g}_k)\Theta_1(\lambda - \mu)Z(J\mathbf{h}_k)^T J + fZ_e(M\mathbf{e}_1)\Theta_1(\lambda - \mu)J$$

and

$$M = \sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k)\Theta_2(\mu - \lambda)K_{n,n}(Z_{1/f}, \mathbf{h}_k)^T - e\Theta_2(\mu - \lambda)K_{n,n}(Z_{1/f}^T, M^T\mathbf{e}_m)$$
$$= \sum_{k=1}^{l} Z(\mathbf{g}_k)\Theta_2(\mu - \lambda)Z_{1/f}(\mathbf{h}_k)^T - e\Theta_2(\mu - \lambda)Z_{1/f}(JM^T\mathbf{e}_m)^T J.$$

Remark 3.18. Assume that $ef \neq 0$ and write $V_e = \left(s_i^{j-1}\right)_{\substack{1\le i\le m\\ 1\le j\le m}}$, $V_f = \left(t_i^{j-1}\right)_{\substack{1\le i\le n\\ 1\le j\le n}}$, $s_1, \ldots, s_m$ are all the m-th roots of e, and $t_1, \ldots, t_n$ are all the n-th roots of f, so that the Vandermonde matrices V_e and V_f are nonsingular.

Now, based on Theorem 2.4, we obtain the following equations:

$$(I_m - f((\lambda - \mu)I_m + Z_e)^{-n})^{-1} = V_e^{-1}\,\mathrm{diag}\left(\frac{(\lambda-\mu+s_i)^n}{(\lambda-\mu+s_i)^n - f}\right)_{1\le i\le m} V_e,$$

$$(I_n - e((\mu - \lambda)I_n + Z_f)^{-m})^{-1} = V_f^{-1}\,\mathrm{diag}\left(\frac{(\mu-\lambda+t_i)^m}{(\mu-\lambda+t_i)^m - e}\right)_{1\le i\le n} V_f.$$

Remark 3.19. By Theorem 2.2, we have

$$\nabla_{\lambda I_m + Z_e, \mu I_n + Z_f}(MJ) = \nabla_{\lambda I_m + Z_e, \mu I_n + Z_f^T}(M)J = G(JH)^T,$$

$$\nabla_{\lambda I_m + Z_e, \mu I_n + Z_f}(JM) = J\nabla_{\lambda I_m + Z_e^T, \mu I_n + Z_f}(M) = (JG)H^T,$$

$$\nabla_{\lambda I_m + Z_e, \mu I_n + Z_f}(JMJ) = J\nabla_{\lambda I_m + Z_e^T, \mu I_n + Z_f^T}(M)J = (JG)(JH)^T.$$

Based on the latter equations, we extend our results to obtain the expressions for structured matrices M via their displacements defined by the Sylvester type operators $L = \nabla_{\lambda I_m + Z_e, \mu I_n + Z_f^T}$, $L = \nabla_{\lambda I_m + Z_e^T, \mu I_n + Z_f}$, and $L = \nabla_{\lambda I_m + Z_e^T, \mu I_n + Z_f^T}$.

3.3 Three Implications

1. The known algorithms enable superfast multiplication of the basic structured matrices by vectors in nearly linear time. Our bilinear expressions of structured matrices via their generators enable immediate extension of these algorithms to multiply structured matrices of various much more general classes by vectors in nearly linear time. In particular, for $m = n$, the expressions of Examples 3.6, 3.8-3.10 enables us to perform multiplication-by-vector of the matrices M of these examples to $O(ln \log n)$ flops. Similarly, we yield the cost bound of $O(ln \log^2 n)$ flops for multiplication by vector of the matrices of Examples 3.11-3.16.

2. For a structured matrix M associated with a Sylvester type operator $L = \nabla_{A,B}$, suppose we know the Jordan blocks of the operator matrices

$n \times n$ Matrix M	Number of parameters	Number of flops for computation of $M\mathbf{v}$
General	n^2	$2n^n - n$
Toeplitz-like	$2ln$	$O(ln \log n)$
Hankel-like	$2ln$	$O(ln \log n)$
Vandermonde-like	$2ln$	$O(ln \log^2 n)$
Cauchy-like	$2ln$	$O(ln \log^2 n)$
Confluent type	$2ln$	$O(ln \log^2 n)$

Table 3.1: Parameter and flop count for matrix representation and its multiplication by a vector

A and B,

$$\hat{A} = PAP^{-1} = \text{diag}(\lambda_i(A)I_{m_i} + Z)_{1 \le i \le p},$$

$$\hat{B} = Q^{-1}BQ = \text{diag}(\lambda_j(B)I_{n_j} + Z)_{1 \le j \le q}.$$

By Theorem 2.14, we can recover $\hat{M} = PMQ$ first by $\nabla_{\hat{A},\hat{B}}(\hat{M}) = P\nabla_{A,B}(M)Q$, and then get $M = P^{-1}\hat{M}Q^{-1}$. Already for $P = I_m$, $Q = I_n$, this covers the more general class of confluent matrices associated with the tangential confluent Nevanlinna-Pick problem [BGR90]. We recover the matrix M from its displacement $L(M) = GH^T$ by applying the following steps:

(a) Write $A_i = \lambda_i(A)I_{m_i} + Z$, $i = 1, \ldots, p$; $B_j = \lambda_j(B)I_{n_j} + Z$, $j = 1, \ldots, q$;

(b) Represent the matrix $\hat{M}$ as a $p \times q$ block matrix with blocks $M_{i,j}$ of size $m_i \times n_j$; represent the matrix PG as a $p \times 1$ block matrix with blocks G_i of size $m_i \times l$; represent the matrix $H^T Q$ as a $1 \times q$ block matrix with blocks H_j^T of size $l \times n_j$.

(c) Replace the matrix equation $\nabla_{A,B}(M) = GH^T$ by the set of block equations

$$\nabla_{A_i,B_j}(M_{i,j}) = G_i H_j^T$$

for all pairs (i,j), $i = 1,\ldots,p$; $j = 1,\ldots,q$.

(d) Recover the blocks $M_{i,j}$ from their displacement generators (G_i, H_j) as in Section 3.2.

(e) Finally, compute the matrix $M = P^{-1}(M_{i,j})_{\substack{1\le i\le p\\ 1\le j\le q}} Q^{-1}$.

3. For a structured matrix M associated with a Stein type operator $L = \Delta_{A,B}$. Similar to the above, we have

$$\Delta_{A_i,B_j}(M_{i,j}) = G_i H_j^T$$

for all pairs (i,j), $i = 1,\ldots,p$; $j = 1,\ldots,q$. After recover the blocks $M_{i,j}$, we finally get M.

Next, let us give bilinear expressions for an $m \times n$ matrices M associated

with operator $L = \Delta_{\lambda I_m + Z, \mu I_n + Z}$. Suppose $L(M) = GH^T = \sum_{k=1}^{l} \mathbf{g}_k \mathbf{h}_k^T$.

(a) If $\lambda = 0$, then

$$\begin{aligned} M &= \sum_{i=0}^{m-1} Z^i G H^T (\mu I_n + Z)^i \\ &= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \binom{i}{j} \mu^{i-j} Z^i G H^T Z^j \\ &= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{i! \mu^{i-j}}{j!(i-j)!} Z^i G H^T Z^j \\ &= \sum_{k=1}^{l} K_{m,m}(Z, \mathbf{g}_k) \Theta_3(\mu) K_{n,n}(Z^T, \mathbf{h}_k)^T \\ &= \sum_{k=1}^{l} Z(\mathbf{g}_k) \Theta_3(\mu) Z(J\mathbf{h}_k)^T J. \end{aligned}$$

(b) If $\lambda \neq 0$, let $R = \left(\frac{\lambda^i (m-j)! (-\lambda)^j}{(m-i)!(i-j)!} \right)_{\substack{1 \leq i \leq m \\ 1 \leq j \leq m}} = \operatorname{diag}\left(\frac{\lambda^i}{(m-i)!} \right)_{1 \leq i \leq m}$ $\left(\frac{1}{(i-j)!} \right)_{\substack{1 \leq i \leq m \\ 1 \leq j \leq m}} \operatorname{diag}\Big((m-j)! (-\lambda)^j \Big)_{1 \leq j \leq m}$, then

$$(\lambda I_m + Z)^{-1} = R(\lambda^{-1} I_m + Z) R^{-1}$$

and

$$\nabla_{\lambda^{-1} I_m + Z, \mu I_n + Z}(R^{-1} M) = (\lambda^{-1} I_m + Z) R^{-1} G H^T.$$

Chapter 4

Norms of the Inverse Displacement Operators

Norm is a critical numerical parameter for structured matrices, especially for our displacement rank approach. For example, when we recover the matrix M from its displacement $L(M)$, the output error is proportional to the norm $\|L^{-1}\|$. If our displacement operator L is so bad that $\|L^{-1}\|$ is tremendously large, then the results are unacceptable. Therefore to choose a good operator is to our interests. There are many different norms for vector, matrices, operators. We use the following definitions and notations.

Definition 4.1. *Norms of vectors, operators and matrices.*

- For a vector $\mathbf{x} = (x_i)$, we define its norm (Euclidean norm) by

$$\|\mathbf{x}\| = \left(\sum_i |x_i|^2 \right)^{1/2}.$$

- For a linear operator L on vector space V, we define its norm by

$$\|L\| = \sup_{\mathbf{x} \in V} \frac{\|L(\mathbf{x})\|}{\|\mathbf{x}\|}.$$

- Viewing a matrix $A = (a_{ij})$ as the vector $\vec{A}$, we define its Frobenius norm to be

$$\|A\|_F = \|\vec{A}\| = \left(\sum_{i,j} |a_{i,j}|^2 \right)^{1/2}.$$

Alternatively, we may view the matrix as a linear operator $L_A : \mathbf{x} \mapsto A\mathbf{x}$ (or $R_A : \mathbf{x} \mapsto \mathbf{x}^T A$) and define its norm to be

$$\|A\| = \|L_A\| = \|R_A\|.$$

- Given a linear operator L on the matrix space, we may restrict L on the matrices having rank of at most r and define

$$\|L\|_r = \sup_{\text{rank}(A) \leq r} \frac{\|L(A)\|}{\|A\|}.$$

Here are some simple results.

Theorem 4.2. *For $r \geq 1$ and a linear operator L on the matrix space, we have*

$$\|L\|_{r-1} \leq \|L\|_r \leq r\|L\|_1.$$

Theorem 4.3. *For any matrix* A, $\|A\| = \sigma_1(A)$, $\|A\|_F = \left(\sum_i \sigma_i(A)^2\right)^{1/2}$. *Therefore,*

$$\|A\|_F/\sqrt{\operatorname{rank}(A)} \leq \|A\| \leq \|A\|_F.$$

Furthermore, when A *is a square matrix,*

$$\|A\| \geq \max_i |\lambda_i(A)|$$

Example 4.4. For $n \times n$ unit f-circulant matrix Z_f,

$$\|Z_f^k\| = \begin{cases} |f|^{k/n}, & \text{if } n \mid k, \\ |f|^{\lfloor k/n \rfloor} \max(1, |f|), & \text{if } n \nmid k. \end{cases}$$

4.1 General Results

In this section, we estimate the operator norm $\|L^{-1}\|$ for the Sylvester type operators $L = \nabla_{A,B}$ and Stein type operators $L = \Delta_{A,B}$ on the space of $m \times n$ matrices, with the general operator matrices A and B.

Theorem 4.5.

$$\max_{i,j} |\lambda_i(A) - \lambda_j(B)|^{-1} \leq \|\nabla_{A,B}^{-1}\|_r \leq \sqrt{r}\|(I \otimes A - B^T \otimes I)^{-1}\|,$$

$$\max_{i,j} |1 - \lambda_i(A)\lambda_j(B)|^{-1} \leq \|\Delta_{A,B}^{-1}\|_r \leq \sqrt{r}\|(I - B^T \otimes A)^{-1}\|.$$

Proof. (1) Given any i, j, let $\mathbf{g}$ and $\mathbf{h}$ be the eigenvectors of A and B respectively, such that

$$A\mathbf{g} = \lambda_i(A)\mathbf{g}, \qquad B^T\mathbf{h} = \lambda_j(B)\mathbf{h}.$$

Let $M = \mathbf{g}\mathbf{h}^T$, we have

$$\nabla_{A,B}(M) = (\lambda_i(A) - \lambda_j(B))M,$$

$$\Delta_{A,B}(M) = (1 - \lambda_i(A)\lambda_j(B))M.$$

Then we have the lower bounds.

(2) Write $\nabla = \nabla_{A,B}(M)$, $\Delta = \Delta_{A,B}(M)$. Combining Theorem 4.3 and equations (2.4), (2.5), we have

$$\frac{\|M\|}{\|\nabla\|} \le \frac{\|M\|_F}{\|\nabla\|_F/\sqrt{\text{rank}(\nabla)}} \le \sqrt{\text{rank}(\nabla)}\,\left\|(I \otimes A - B^T \otimes I)^{-1}\right\|,$$

$$\frac{\|M\|}{\|\Delta\|} \le \frac{\|M\|_F}{\|\Delta\|_F/\sqrt{\text{rank}(\Delta)}} \le \sqrt{\text{rank}(\Delta)}\,\left\|(I - B^T \otimes A)^{-1}\right\|.$$

Now we have the upper bounds. □

Theorem 4.5 involves the computation of $\|(I \otimes A - B^T \otimes I)^{-1}\|$ and $\|(I - B^T \otimes A)^{-1}\|$, which are usually unavailable. However, if we know the entries of $(I \otimes A - B^T \otimes I)^{-1}$ and $(I - B^T \otimes A)^{-1}$, we may apply the following theorem to estimate $\|(I \otimes A - B^T \otimes I)^{-1}\|$ and $\|(I - B^T \otimes A)^{-1}\|$.

Theorem 4.6. *[GL96]. For any $m \times n$ matrix $X = (x_{ij})_{\substack{1\le i\le m,\\ 1\le j\le n}}$, we have*

$$\|X\|^2 \le \left(\max_{1\le j\le n}\sum_{i=1}^{m}|x_{ij}|\right)\left(\max_{1\le i\le m}\sum_{j=1}^{n}|x_{ij}|\right).$$

Our next upper bounds of $\|L^{-1}\|$ rely on the bilinear expressions for the structured matrices M via their L-generators and the transformation between displacement operators.

Theorem 4.7. *For any natural number r, we have*

$$\|\Delta^{-1}_{A^{-1},B}\|_r/\|A\| \le \|\nabla^{-1}_{A,B}\|_r \le \|A^{-1}\|\cdot\|\Delta^{-1}_{A^{-1},B}\|_r \text{ if } A \text{ is nonsingular,}$$

$$\|\Delta^{-1}_{A,B^{-1}}\|_r/\|B\| \le \|\nabla^{-1}_{A,B}\|_r \le \|B^{-1}\|\cdot\|\Delta^{-1}_{A,B^{-1}}\|_r \text{ if } B \text{ is nonsingular.}$$

Proof. Corollary of Theorem 2.6. □

Theorem 4.8. *For nonsingular matrices P and Q, write $\hat{A} = PAP^{-1}$, $\hat{B} = Q^{-1}BQ$, then we have*

$$\|\nabla^{-1}_{\hat{A},\hat{B}}\|_r/(\kappa(P)\kappa(Q)) \le \|\nabla^{-1}_{A,B}\|_r \le \kappa(P)\kappa(Q)\|\nabla^{-1}_{\hat{A},\hat{B}}\|_r,$$

$$\|\Delta^{-1}_{\hat{A},\hat{B}}\|_r/(\kappa(P)\kappa(Q)) \le \|\Delta^{-1}_{A,B}\|_r \le \kappa(P)\kappa(Q)\|\Delta^{-1}_{\hat{A},\hat{B}}\|_r.$$

Proof. Corollary of Theorem 2.14. □

Theorem 4.9. *For operators $\Delta_{A,B}$ of Corollary 3.2, we have*

$$\|\Delta_{A,B}^{-1}\| \le \sum_{i=1}^{k-1} \|A^i\| \cdot \|B^i(I - aB^k)^{-1}\| \text{ if } A^k = aI,$$

$$\|\Delta_{A,B}^{-1}\| \le \sum_{i=1}^{k-1} \|(I - bA^k)^{-1}A^i\| \cdot \|B^i\| \text{ if } B^k = bI.$$

4.2 Some Examples

In this section, we will apply the results of Section 4.1 to those operators we discussed in Section 3.2. Explicitly, we only estimate $\|L^{-1}\|$ where $L = \Delta_{Z_e,Z_f}$, $L = \nabla_{Z_e,Z_f}$, $L = \Delta_{Z_e,D(\mathbf{v})}$, $L = \nabla_{Z_e,D(\mathbf{v})}$, and $L = \Delta_{D(\mathbf{u}),D(\mathbf{v})}$, $L = \nabla_{D(\mathbf{u}),D(\mathbf{v})}$. All our proofs and estimates, however, are invariant to interchanging the operator matrices A and B and to the transposition any of A and B, so the same estimates are immediately extended to the operators $\Delta_{Z_e^T,Z_f}$, $\nabla_{Z_e^T,Z_f}$, Δ_{Z_e,Z_f^T}, ∇_{Z_e,Z_f^T}, $\Delta_{Z_e^T,Z_f^T}$, $\nabla_{Z_e^T,Z_f^T}$, $\Delta_{Z_e^T,D(\mathbf{v})}$, $\nabla_{Z_e^T,D(\mathbf{v})}$, $\Delta_{D(\mathbf{v}),Z_e}$, $\nabla_{D(\mathbf{v}),Z_e}$, $\Delta_{D(\mathbf{v}),Z_e^T}$, $\nabla_{D(\mathbf{v}),Z_e^T}$, respectively.

Example 4.10. Write $\hat{e} = \max(1, |e|)$, $\hat{f} = \max(1, |f|)$, $\ell = \mathrm{lcm}(m, n)$. By Example 4.4 and Theorem 4.9, we have

$$\|\Delta_{Z_e,Z_f}^{-1}\| \le \frac{\hat{e}\hat{f}}{|1 - e^{\frac{\ell}{m}} f^{\frac{\ell}{n}}|} \sum_{k=0}^{\ell-1} |e|^{\lfloor \frac{k}{m} \rfloor} |f|^{\lfloor \frac{k}{n} \rfloor} \tag{4.1}$$

$$\|\nabla_{Z_e,Z_f}^{-1}\| \le \frac{\hat{e}\hat{f}}{|e^{\frac{\ell}{m}} - f^{\frac{\ell}{n}}|} \sum_{k=0}^{\ell-1} |e|^{\lfloor \frac{\ell-1-k}{m} \rfloor} |f|^{\lfloor \frac{k}{n} \rfloor} \tag{4.2}$$

Remark 4.11. Suppose $m = n$, then $\mathrm{lcm}(m,n) = n$. By Theorems 4.5, we have

$$\left|1 - |ef|^{\frac{1}{n}}\omega_{2n}\right|^{-1} \leq \|\Delta^{-1}_{Z_e,Z_f}\| \leq \frac{n\hat{e}\hat{f}}{|1-ef|},$$
$$\left||e|^{\frac{1}{n}} - |f|^{\frac{1}{n}}\omega_{2n}\right|^{-1} \leq \|\nabla^{-1}_{Z_e,Z_f}\| \leq \frac{n\hat{e}\hat{f}}{|e-f|}.$$

Comparing the lower and upper bounds as $n \to \infty$, we obtain

$$\lim_{n\to\infty} \frac{|1-ef|}{n\hat{e}\hat{f}\left|1 - |ef|^{\frac{1}{n}}\omega_{2n}\right|} = \begin{cases} \frac{|1-ef|}{\hat{e}\hat{f}\sqrt{\pi^2+\ln^2|ef|}} & \text{if } ef \neq 0, \\ 0 & \text{if } ef = 0, \end{cases}$$

$$\lim_{n\to\infty} \frac{|e-f|}{n\hat{e}\hat{f}\left||e|^{\frac{1}{n}} - |f|^{\frac{1}{n}}\omega_{2n}\right|} = \begin{cases} \frac{|e-f|}{\hat{e}\hat{f}\sqrt{\pi^2+\ln^2|e/f|}} & \text{if } ef \neq 0, \\ 0 & \text{if } ef = 0, \end{cases}$$

where ln denotes the natural logarithm. That is, our estimates (4.1), (4.2) are asymptotically tight as $n \to \infty$ provided $ef \neq 0$ and $ef \neq 1$ (for Δ_{Z_e,Z_f}) or $e \neq f$ (for ∇_{Z_e,Z_f}).

In the case $ef = 0$, say $f = 0$. Let $\Delta = \Delta_{Z_e,Z}(M) = \mathbf{1}\,\mathbf{1}^T$, then

$$M = \begin{pmatrix} 1 & e & \cdots & e \\ 1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & e \\ 1 & \cdots & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & \cdots & 1 & 1 \\ \vdots & \iddots & 1 & \\ 1 & \iddots & & \\ 1 & & & \end{pmatrix}.$$

Write $e = x + y\sqrt{-1}$ where x, y are real numbers. Since $\|\Delta\| = n$ and

$$\begin{aligned}\|M\|^2 \geq \frac{1}{n}\|\mathbf{1}^T M\|^2 &= \frac{1}{n}\sum_{i=1}^{n}\left|in + \frac{i(i-1)}{2}(e-1)\right|^2 \\ &\geq \left(\frac{n^4}{20} - \frac{n^3}{8} + \frac{n^2}{12}\right)(x-1)^2 + \left(\frac{n^4}{4} - \frac{n^3}{3}\right)(x-1) + \frac{n^4}{3} \\ &\geq \frac{n^4}{48} + O(n^3),\end{aligned}$$

we have $\|\Delta^{-1}_{Z_e,Z}\|_1 > cn$ for some constant $c > 0$. Similarly, we have $\|\nabla^{-1}_{Z_e,Z}\|_1 \geq cn$ for another constant $c > 0$. This leads to much tighter bounds than applying Theorem 4.5 for $A = Z_e$, $B = Z_0$.

In both cases, we have

$$\|\nabla^{-1}_{Z_e,Z_f}\|_1 = \Omega(n) \text{ for all } ef \neq 1, \tag{4.3}$$

$$\|\Delta^{-1}_{Z_e,Z_f}\|_1 = \Omega(n) \text{ for all } e \neq f. \tag{4.4}$$

Example 4.12.

$$\|\Delta^{-1}_{Z_e,D(\mathbf{v})}\| \leq \hat{e}\sum_{k=0}^{m-1}\max_j\left|\frac{v_j^k}{1-ev_j^m}\right|, \tag{4.5}$$

$$\|\nabla^{-1}_{Z_e,D(\mathbf{v})}\| \leq \hat{e}\sum_{k=0}^{m-1}\max_j\left|\frac{v_j^k}{e-v_j^m}\right|. \tag{4.6}$$

Remark 4.13. Suppose $|v_j| \notin (1-\epsilon, 1+\epsilon)$ for a constant $\epsilon > 0$ and for all

j. If $e \neq 0$ then

$$\lim_{m\to\infty} \sum_{k=0}^{m-1} \max_j |\tfrac{v_j^k}{1-ev_j^m}| < \frac{1}{\epsilon} \max(1, \tfrac{1}{|e|}),$$
$$\lim_{m\to\infty} \sum_{k=0}^{m-1} \max_j |\tfrac{v_j^k}{e-v_j^m}| < \frac{1}{\epsilon} \max(1, \tfrac{1}{|e|}).$$

If $e = 0$, let $\Delta = \Delta_{Z,D(\mathbf{v})}(M) = \mathbf{e}_1 \mathbf{e}_j^T$, then

$$M = \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & v_j & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & v_j^{n-1} & 0 & \cdots & 0 \end{pmatrix}.$$

Write $v = \max_j |v_j|$, since $\|\Delta\| = 1$, we have

$$\sqrt{\frac{v^{2m}-1}{v^2-1}} \leq \|\Delta_{Z,D(\mathbf{v})}^{-1}\|_1 \leq \frac{v^m-1}{v-1}.$$

Compare the lower and upper bounds as $m \to \infty$, we have

$$\lim_{m\to\infty} \frac{\sqrt{\frac{v^{2m}-1}{v^2-1}}}{\frac{v^m-1}{v-1}} = |\tfrac{v+1}{v-1}|.$$

Similarly, we have

$$u\sqrt{\frac{u^{2m}-1}{u^2-1}} \leq \|\nabla_{Z,D(\mathbf{v})}^{-1}\|_1 \leq u\frac{u^m-1}{u-1},$$

where $u = \max_j |v_j|^{-1}$. That is, our estimates (4.5), (4.6) are asymptotically tight as $m \to \infty$ provided that $\{v_j\}$ are not clustered around 1.

Example 4.14.

$$\|\Delta^{-1}_{D(\mathbf{u}),D(\mathbf{v})}\|_r \le \frac{\sqrt{r}}{\min_{i,j}|1-u_iv_j|}, \tag{4.7}$$

$$\|\nabla^{-1}_{D(\mathbf{u}),D(\mathbf{v})}\|_r \le \frac{\sqrt{r}}{\min_{i,j}|u_i-v_j|}. \tag{4.8}$$

Remark 4.15. By Theorem 4.5, $\|\Delta^{-1}_{D(\mathbf{u}),D(\mathbf{v})}\|_1 \ge \frac{1}{\min_{i,j}|1-u_iv_j|}$, $\|\nabla^{-1}_{D(\mathbf{u}),D(\mathbf{v})}\|_1 \ge \frac{1}{\min_{i,j}|u_i-v_j|}$. This is within $\sqrt{r}$ from the upper bounds of Example 4.14. That is, our estimates are tight for "small" r, which is the most interesting case where we study structured matrices with "small" displacement rank.

By combining Theorems 2.4 and 4.8, we transform the operators $\Delta^{-1}_{Z_e,Z_f}$, $\nabla^{-1}_{Z_e,Z_f}$, and $\Delta^{-1}_{Z_e,D(\mathbf{v})}$, $\nabla^{-1}_{Z_e,D(\mathbf{v})}$ into operators $\Delta^{-1}_{D(\mathbf{u}),D(\mathbf{v})}$, $\nabla^{-1}_{D(\mathbf{u}),D(\mathbf{v})}$ and then extend the bounds of Example 4.14 to the former operators.

Corollary 4.16. *Suppose $ef \neq 0$, $e^{\frac{1}{m}}$ and $f^{\frac{1}{n}}$ are any m-th and n-th roots of e and f, respectively. Write $\tilde{e} = \max(|e|, \frac{1}{|e|})$, $\tilde{f} = \max(|f|, \frac{1}{|f|})$, then*

$$\|\Delta^{-1}_{Z_e,Z_f}\|_r \le \sqrt{r}\tilde{e}^{\frac{m-1}{m}}\tilde{f}^{\frac{n-1}{n}} \max_{i,j}\left|1-e^{\frac{1}{m}}\omega_m^i f^{\frac{1}{n}}\omega_n^j\right|^{-1}, \tag{4.9}$$

$$\|\nabla^{-1}_{Z_e,Z_f}\|_r \le \sqrt{r}\tilde{e}^{\frac{m-1}{m}}\tilde{f}^{\frac{n-1}{n}} \max_{i,j}\left|e^{\frac{1}{m}}\omega_m^i - f^{\frac{1}{n}}\omega_n^j\right|^{-1}, \tag{4.10}$$

$$\|\Delta_{Z_e,D(\mathbf{v})}\|_r \le \sqrt{r}\tilde{e}^{\frac{m-1}{m}} \max_{i,j}\left|1-e^{\frac{1}{m}}\omega_m^i v_j\right|^{-1}, \tag{4.11}$$

$$\|\nabla_{Z_e,D(\mathbf{v})}\|_r \le \sqrt{r}\tilde{e}^{\frac{m-1}{m}} \max_{i,j}\left|e^{\frac{1}{m}}\omega_m^i - v_j\right|^{-1}. \tag{4.12}$$

4.3 Decreasing the Norm $||L^{-1}||$

Typically, in the DECOMPRESS stage of the displacement rank approach, the numerical problems generally diminish where $||L^{-1}||$ is smaller. In particular, this factor is critical for rapid convergence of Newton's iteration with recursive compression applied to invert a structured matrix [P92], [PRW00], [P01], [PRW01], [PKRCa]. It is, therefore, desired to decrease $||L^{-1}||$. This is usually possible based on the displacement transformation approach proposed in [P90]. According to this approach, given a successful algorithm or method of study for a specific class of structured matrices, one may extend these method or algorithm to other classes of structured matrices via appropriate transformation of the associated displacement operators. At the end of the preceding section, we applied this approach to extend our estimates of Example 4.14 from Cauchy-like to Toeplitz/Hankel-like and Vandermonde-like matrices. Let us demonstrate how this works by another example. Suppose we seek the solution of a nonsingular linear system $M\mathbf{x} = \mathbf{b}$ where the Cauchy-like input matrix M is associated with an operator $L_0 = \nabla_{D(\mathbf{s}),D(\mathbf{t})}$ and the norm $||L_0^{-1}||$ is too large. Let us solve the problem by using the displacement transformation method. Choose a vector $\mathbf{r} = (a\omega_n^i)_{i=0}^{n-1}$ for a scalar a such that $||L^{-1}||$ is small for $L = \nabla_{D(\mathbf{r}),D(\mathbf{t})}$. According to Example

4.14, this is the case if the component sets of the vectors $\mathbf{r}$ and $\mathbf{t}$ are well isolated from each other. Solve the linear system $C(\mathbf{r},\mathbf{s})M\mathbf{x} = C(\mathbf{r},\mathbf{s})\mathbf{b}$ whose coefficient matrix is associated with the operator $L = \nabla_{D(\mathbf{r}),D(\mathbf{t})}$ and is typically not worse conditioned than M. Due to the transition from L_0 to L, the critical stage of the solution can be dramatically simplified (for instance, if the solution is by using Newton's iteration). The above recipe can be immediately extended to the case of Toeplitz-like, Hankel-like, Vandermonde-like, and other structured matrices M based on their well-known simple transformations into Cauchy-like matrices (see Theorems 2.4 and 4.8 and [P90], [P91]).

Chapter 5

Numerical Inversion of Structured Matrices

Computing the inverse of a structured matrix is among the most important practical computational problems. It is essentially equivalent to solving linear systems. The classical algorithms compute the inversion of a general $n \times n$ matrix in $O(n^3)$ flops by using Gaussian elimination. The two "superfast" algorithms proposed by Brent et al. [BGY80] and by Morf [M74], [M80] and Bitmead/Anderson [BA80] solve the Toeplitz linear system of n equations in $O(n \log^2 n)$ flops. We refer to the two algorithms as the BGY and MBA algorithms, respectively. The BGY algorithm applies only to Toeplitz/Hankel

linear systems, whereas the MBA algorithm covers also the more general class of Toeplitz/Hankel-like linear systems [KS99], [P01].

In this chapter, we discuss an algorithm of different type. We use Newton's iteration to compute the inverse of a structured matrix. This iterative method has been long and well known [S33], [B-I66], [B-IC66], [SS74], [PS91]. It is very attractive because of the strong numerical stability, local quadratic convergence, and convenience for parallel implementation. However, the applications in the case of general matrices (especially for general matrices of large size) was limited because each iteration step involves two expensive operations of matrix multiplication. This problem disappears when the matrices are structured, but another problem arises. One has to preserve the matrix structure (i.e., reduce the displacement rank) during the iteration. Here is our algorithm.

Algorithm 5.1 (Newton's Iteration for Structured Matrices).

Input: An $n \times n$ structured matrix M with $\nabla_{A,B}$-generator (G, H) of length r, an initial approximation X_0 to M^{-1} with $\nabla_{B,A}$-generator (G_0, H_0) of length r.

Output: A $\nabla_{B,A}$-generator of M^{-1}.

Computation: For $i = 0, 1, 2, \ldots, k$, compute

$$Y_i = 2X_i - X_i M X_i,$$

$$\nabla_{B,A}(X_{i+1}) = Truncate(\nabla_{B,A}(Y_i))$$

until $\{X_i\}$ converges and output X_k.

In the next, we give two methods for the subroutine *Truncate.*

5.1 SVD Truncation

Our first method is to minimize

$$\|\nabla_{B,A}(X_{i+1}) - \nabla_{B,A}(Y_i)\|$$

for rank $\nabla_{B,A}(X_{i+1}) = r$. From [P01, Theorem 6.1.3], we know that

$$\nabla_{B,A}(X_{i+1}) = U \,\mathrm{diag}[\sigma_1, \ldots, \sigma_r, 0, \ldots, 0]V$$

where

$$\nabla_{B,A}(Y_i) = U \,\mathrm{diag}[\sigma_1, \ldots, \sigma_l]V$$

is the SVD. Since

$$\|\nabla_{B,A}(X_{i+1}) - \nabla_{B,A}(Y_i)\| \leq \|\nabla_{B,A}(M^{-1}) - \nabla_{B,A}(Y_i)\|,$$

we have

$$\|\nabla_{B,A}(X_{i+1}) - \nabla_{B,A}(M^{-1})\| \leq 2\|\nabla_{B,A}(Y_i) - \nabla_{B,A}(M^{-1})\|.$$

Therefore,

$$\|X_{i+1} - M^{-1}\| \leq 2\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|Y_i - M^{-1}\|$$

$$\|I - MX_{i+1}\| \leq 2\kappa(M)\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|I - MY_i\|$$

$$\leq 2\kappa(M)\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|I - MX_i\|^2.$$

That is, the iteration has quadratic convergence if

$$\|I - MX_0\| < \left(4\kappa(M)\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\|\right)^{-1}. \tag{5.1}$$

5.2 Least Squares Truncation

Recall that

$$M\nabla_{B,A}(M^{-1})M + \nabla_{A,B}(M) = 0.$$

Our second method is to minimize

$$\|M\nabla_{B,A}(X_{i+1})M + \nabla_{A,B}(M)\|$$

for $\nabla_{B,A}(X_{i+1}) = \tilde{G}_i\Omega\tilde{H}_i^T$, where $(\tilde{G}_i, \tilde{H}_i)$ is a $\nabla_{B,A}$-generator of Y_i. This least squares problem has a solution

$$\Omega = -(M\tilde{G}_i)^+ GH^T(\tilde{H}_i^T M)^+,$$

where $(\ )^+$ is the generalized inverse. Therefore,

$$\|M\nabla_{B,A}(X_{i+1})M + \nabla_{A,B}(M)\| \leq \|M\nabla_{B,A}(Y_i)M + \nabla_{A,B}(M)\|$$

$$\|\nabla_{B,A}(X_{i+1}) - \nabla_{B,A}(M^{-1})\| \leq \kappa(M)^2\|\nabla_{B,A}(Y_i) - \nabla_{B,A}(M^{-1})\|$$

$$\|X_{i+1} - M^{-1}\| \leq \kappa(M)^2\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|Y_i - M^{-1}\|$$

$$\|I - MX_{i+1}\| \leq \kappa(M)^3\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|I - MY_i\|$$

$$\leq \kappa(M)^3\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|I - MX_i\|^2.$$

That is, the iteration has quadratic convergence if

$$\|I - MX_0\| < \left(2\kappa(M)^3\|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\|\right)^{-1}. \tag{5.2}$$

Remark 5.2. $\nabla_{B,A}(X_{i+1})$ is independent of the choice of full rank displacement generator $(\tilde{G}_i, \tilde{H}_i)$. $\nabla_{B,A}(X_{i+1})$ depends only on the linear vector space spanned by the column vectors of $\tilde{G}_i$ and $\tilde{H}_i$.

5.3 Initial Approximation

Our two methods of above need very close initial approximations X_0 (see (5.1) and (5.2)). We use the well-known Hestenes-Stiefel *conjugate gradient* method [HS52], [R71] to compute the generators of M^{-1}, i.e., to solve the linear systems $M\mathbf{x} = \mathbf{g}_i$ and $M^T\mathbf{x} = \mathbf{h}_i$, $i = 1, 2, \ldots, r$. The conjugate

gradient method can be outlined as follows.

Algorithm 5.3 (Conjugate Gradient).

Input: Matrix M, vectors $\mathbf{b}$, $\mathbf{x}_0$, scalar $\epsilon > 0$.

Output: vector $\mathbf{x}_k$ such that $\|\mathbf{b} - M\mathbf{x}_k\| < \epsilon$.

Computation: Let $\mathbf{p}_0 = \mathbf{0}$. For $i = 0, 1, \ldots, k$, compute

$$\mathbf{r}_i = \mathbf{b} - M\mathbf{x}_i$$
$$\mathbf{p}_{i+1} = M^H\mathbf{r}_i + \frac{\|M^H\mathbf{r}_i\|}{\|M^H\mathbf{r}_{i-1}\|}\mathbf{p}_i$$
$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\|M^H\mathbf{r}_i\|}{\|M\mathbf{p}_{i+1}\|}\mathbf{p}_{i+1}$$

until $\|\mathbf{r}_k\| < \epsilon$.

Theorem 5.4. *[GL96, Theorem 10.2.6]*

$$\|\mathbf{r}_k\| \leq 2\left(\frac{\kappa(M) - 1}{\kappa(M) + 1}\right)^k \|\mathbf{r}_0\|.$$

Remark 5.5. For $n \times n$ structured matrix M, each iteration step can be performed in $O(n \log^d n)$ flops. If $\kappa(M)$ is small (say, $\kappa(M) \leq 4$), then the iteration finishes in $k = O(|\log \epsilon|)$ steps.

Remark 5.6. For positive definite matrix M, the iterations of Algorithm 5.3 can be replaced by

$$\begin{aligned}
\mathbf{r}_i &= \mathbf{b} - M\mathbf{x}_i \\
\mathbf{p}_{i+1} &= \mathbf{r}_i + \frac{\|\mathbf{r}_i\|}{\|\mathbf{r}_{i-1}\|}\mathbf{p}_i \\
\mathbf{x}_{i+1} &= \mathbf{x}_i + \frac{\|\mathbf{r}_i\|}{\mathbf{p}_{i+1}^T M \mathbf{p}_{i+1}}\mathbf{p}_{i+1}.
\end{aligned}$$

Then we have

$$\|\mathbf{r}_k\| \leq 2\sqrt{\kappa(M)}\left(\frac{\sqrt{\kappa(M)}-1}{\sqrt{\kappa(M)}+1}\right)^k \|\mathbf{r}_0\|.$$

For ill-conditioned matrix M, we cannot use the conjugate gradient algorithm directly because of the long iteration steps. However, we may overcome this difficulty by using homotopy method.

- First, we assume M is a Toeplitz-like matrix. For structured matrix M of other classes, we may reduce the problem to solving Toeplitz-like linear system of equations.

- Next, we assume M is positive definite because $M^{-1} = (M^H M)^{-1} M^H$ and $M^H M$ is a positive definite Toeplitz-like matrix.

- Finally, we assume $\sigma_1(M)$ and $\sigma_n(M)$ (the largest and smallest singular values of M) are available.

Let $M_0 = I$, $M_i = (1 - a_i)I + a_i M$, $Q_i = M_{i-1}^{-1} M_i$ for $i = 1, 2, \ldots, m$, where $0 < a_1 < \ldots < a_m = 1$ are parameters that we compute from $\sigma_1(M)$ and $\sigma_n(M)$ such that $2 \leq \kappa(Q_i) \leq 3$ for all i. Note that all M_i's and Q_i's are positive definite and $\kappa(Q_i) = \kappa(M_i)/\kappa(M_{i-1})$, so $m = O(\log \kappa(M))$. The following preconditioned conjugate gradient method computes $M_i^{-1}\mathbf{b}$, provided that M_{i-1}^{-1} is already computed. It is essentially the same as to compute $Q_i^{-1}(M_{i-1}^{-1}\mathbf{b})$ by using the conjugate gradient method.

Remark 5.7. When $\sigma_1(M)$ and $\sigma_n(M)$ are not avaliable, we still can compute a_i's from an upper bound σ of $\sigma_1(M)$, because

$$\kappa(Q_i) \leq \frac{1 - a_i}{1 - a_{i+1}} \cdot \frac{(1 - a_{i+1}) + a_{i+1}\sigma}{(1 - a_i) + a_i \sigma}.$$

Algorithm 5.8 (Preconditioned Conjugate Gradient).

Input: Matrices M, $N \approx M^{-1}$, vectors $\mathbf{b}$, $\mathbf{x}_0$, scalar $\epsilon > 0$.

Output: vector $\mathbf{x}_k$ such that $\|\mathbf{b} - M\mathbf{x}_k\| < \epsilon$.

Computation: Let $\mathbf{p}_0 = \mathbf{0}$. For $i = 0, 1, \ldots, k$, compute

$$\mathbf{r}_i = \mathbf{b} - M\mathbf{x}_i$$

$$\mathbf{p}_{i+1} = N\mathbf{r}_i + \frac{\|N\mathbf{r}_i\|}{\|N\mathbf{r}_{i-1}\|}\mathbf{p}_i$$

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\|N\mathbf{r}_i\|}{\mathbf{p}_{i+1}^T NM\mathbf{p}_{i+1}}\mathbf{p}_{i+1}$$

until $\|\mathbf{r}_k\| < \epsilon$.

Algorithm 5.7 outputs the generators of a Toeplitz-like matrix $N_i \approx M_i^{-1}$ such that $\operatorname{rank} \nabla_{B,A}(N_i) = r_i = \operatorname{rank} \nabla_{B,A}(M_i)$ and

$$\|M_i \nabla_{B,A}(N_i) M_i + \nabla_{A,B}(M_i)\| \leq 2\epsilon\sqrt{r_i \|\nabla_{A,B}(M_i)\|}.$$

Before the last homotopic step, we need to keep $N_i M_{i+1}$ well-conditioned, say, $\kappa(N_i M_{i+1}) \leq 4$ for all $i = 1, 2, \ldots, m-1$. Then it suffices to have $\|I - N_i M_i\| \leq \frac{1}{7}$ because

$$\kappa(N_i M_{i+1}) \leq \kappa(N_i M_i)\kappa(M_i^{-1} M_{i+1}) \leq 3\frac{1 + \|I - N_i M_i\|}{1 - \|I - N_i M_i\|}.$$

Furthermore, because

$$\begin{aligned}
\|I - N_i M_i\| &\leq \|M_i\| \cdot \|N_i - M_i^{-1}\| \\
&\leq \|M_i\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|\nabla_{B,A}(N_i) + M^{-1}\nabla_{A,B}(M_i)M^{-1}\| \\
&\leq \|M_i\| \cdot \|\nabla_{B,A}^{-1}\| \cdot \|M_i^{-1}\|^2 \cdot \|M_i \nabla_{B,A}(N_i) M_i + \nabla_{A,B}(M_i)\|,
\end{aligned}$$

it suffices to choose

$$\epsilon \leq \left(14\|M_i\| \cdot \|M_i^{-1}\|^2 \cdot \|\nabla_{B,A}^{-1}\| \sqrt{r_i \|\nabla_{A,B}(M_i)\|}\right)^{-1}.$$

In the last homotopic step, the output $X_0 = N_m$ should satisfy condition (5.1) or (5.2). Then it suffices to choose

$$\epsilon \leq \left(8\kappa(M)^2 \cdot \|M^{-1}\| \cdot \|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\|^2 \sqrt{r\|\nabla_{A,B}(M)\|}\right)^{-1},$$

or

$$\epsilon \leq \left(4\kappa(M)^4 \cdot \|M^{-1}\| \cdot \|\nabla_{B,A}\| \cdot \|\nabla_{B,A}^{-1}\|^2 \sqrt{r\|\nabla_{A,B}(M)\|}\right)^{-1}.$$

respectively.

Remark 5.9. Suppose $\|M\| = n^{O(1)}$, then it takes

$$O\Big(n \log n \log \kappa(M)(\log n + \log \kappa(M))\Big)$$

flops to compute X_0 satisfying (5.1) or (5.2).

Chapter 6

Inversion of Integer Toeplitz-like Matrices

In this chapter, we combines the current fastest MBA algorithm and p-adic lifting algorithms for the exact inversion of an $n \times n$ integer Toeplitz-like matrix M. The algorithm can be outlined as follows.

Algorithm 6.1.

1. Compute $M^{-1} \bmod p$ for a random prime p, $\det(M) \neq 0 \bmod p$.

2. Compute $M^{-1} \bmod q$ for a $q = p^m$ for a sufficiently large m.

3. Compute $s_n = s_n(M)$, the largest invariant factor of M.

4. Output the displacement generator of M^{-1}.

In the first step, we compute $M^{-1} \bmod p$ by first applying the MBA algorithm for computing the balanced complete recursive triangular factorization of M and M^{-1}. If $\det(M) = 0 \bmod p$, the process will stop and return FAILURE. Then we may try another random prime p until SUCCESS. In the second step, we use p-adic lifting to compute $M^{-1} \bmod q$ for $q = p^m$ large enough (see (6.4) in Section 6.3). Then we recover $\frac{\eta_i}{\delta_i} = \mathbf{u}_i^T M^{-1} \mathbf{v}_i$ from $\mathbf{u}_i^T M^{-1} \mathbf{v}_i \bmod q$ for random integer vectors $\mathbf{u}_i, \mathbf{v}_i$, $i = 1, 2, \ldots, k$. In this case, $\mathrm{lcm}(\delta_1, \delta_2, \ldots, k) = s_n(M)$ with high probability. Finally, we output the displacement generator $(\frac{sM^{-1}G}{s}, \frac{sM^{-T}H}{s})$ of M^{-1}.

6.1 The MBA Algorithm

Let M be partitioned as a 2×2 block matrix,

$$M = \begin{pmatrix} M_1 & M_2 \\ M_3 & M_4 \end{pmatrix}$$

where M_1 has size $\lceil \frac{n}{2} \rceil \times \lceil \frac{n}{2} \rceil$. Let $r = \mathrm{rank}(\Delta_{Z,Z^T}(M))$, we have

$$\mathrm{rank}(\Delta_{Z,Z^T}(M_1)) \leq r, \qquad \mathrm{rank}(\Delta_{Z,Z^T}(M_2)) \leq r+1,$$

$$\mathrm{rank}(\Delta_{Z,Z^T}(M_3)) \leq r+1, \qquad \mathrm{rank}(\Delta_{Z,Z^T}(S)) \leq r.$$

Suppose M_1 is nonsingular, then

$$M^{-1} = \begin{pmatrix} M_1^{-1} + M_1^{-1}M_2S^{-1}M_3M_1^{-1} & -M_1^{-1}M_2S^{-1} \\ -S^{-1}M_3M_1^{-1} & S^{-1} \end{pmatrix} \tag{6.1}$$

where $S = M_4 - M_3M_1^{-1}M_2$ is the Schur complement of M_1. Therefore, when M has generic rank profile we may compute M^{-1} recursively by computing M_1^{-1} and S^{-1} first.

Denote by $F_q(n)$ the bit cost of computing M^{-1} mod q, then we have

$$F_q(n) = 2F_q(n/2) + O(r^2\nu_q(n/2)),$$

where $\nu_q(n)$ is the bit cost of multiplying modulo q a Toeplitz matrix by a vector. Thus, $F_p(n) = O(r^2\nu_p(n)\log n)$.

Remark 6.2.

$$\nu_q(n) = O(\mu(n\log q)) \tag{6.2}$$

where $\mu(d)$ is the bit cost of multiplying two integers modulo 2^d, and [SS71]

$$\mu(d) = O((d\log d)\log\log d). \tag{6.3}$$

Remark 6.3. In the case M does not has generic rank profile, we may precondition M by multiplying two Toeplitz matrices X, Y on both sides such that XMY has generic rank profile. See [P01, Section 5.6] for details.

6.2 p-adic Lifting

Suppose we have already computed $M^{-1} \bmod p$ by recursively using (6.1). $A = Z_0$, $B = Z_1$, a $\Delta_{A,B}$-generator of $M \bmod p$ of length r and a $\Delta_{B,A}$-generator of $M^{-1} \bmod p$ of length $l = O(r)$ are available.

Algorithm 6.4 (Newton's Lifting).

Input: $X_0 = M^{-1} \bmod p$.

Output: $X_k = M^{-1} \bmod q$, $q = p^{2^k}$.

Computation: For $i = 1, 2, \ldots, k$, compute

$$Y_i = 2X_{i-1} - X_{i-1}MX_{i-1} \mod p^{2^i},$$

$$\nabla_{B,A}(X_i) = -Y_i \nabla_{A,B}(M) Y_i \mod p^{2^i}.$$

Complexity: In each iteration steps, the bit cost is $O(r^2\mu(n2^i \log p))$ ops. The total bit cost is $O(r^2\mu(n \log q))$ ops.

Algorithm 6.5 (Hensel's Lifting).

Input: $X_0 = M^{-1} \bmod p$, $\mathbf{v}_0 \in \mathbb{Z}^{n\times 1}$ such that $\|\mathbf{v}_0\| \leq n\|M\|$.

Output: $M^{-1}\mathbf{v}_0 \bmod q$, $q = p^k$.

Computation: For $i = 1, 2, \ldots, k-1$, compute

$$\mathbf{u}_i = X_0\mathbf{v}_i \bmod p,$$

$$\mathbf{v}_{i+1} = (\mathbf{v}_i - M\mathbf{u}_i)/p.$$

Finally, output the formal sum $\mathbf{u}_0 + \mathbf{u}_1 p + \ldots + \mathbf{u}_{k-1}p^{k-1}$.

Complexity: In each iteration steps, the bit cost is $O(r\mu(n\log(n^2p\|M\|)))$ ops. The total bit cost is $O(rk\mu(n\log(n^2p\|M\|)))$ ops.

6.3 The Largest Invariant Factor

The known best algorithm for computing s_n is the algorithm **Largest Invariant Factor** in [EGV00]. It is based on the approach proposed in [P87, Appendix], [P88], [ABM99]. The approach reduces the problem to solving linear systems $M\mathbf{x} = \mathbf{v}$ for random vectors $\mathbf{v}$ and to subsequent randomized recovery of s_n as the least common denominator of all components of $\mathbf{x}$. But unlike [EGV00], we recover $s_n(M)$ from the denominators of a scalar $\mathbf{u}^T\mathbf{x} = \mathbf{u}^T M^{-1}\mathbf{v}$ for a random vector $\mathbf{u}$. This saves us the factor of roughly n in the estimated bit cost of the recovery. The acceleration relies on a simple

trick of obtaining the LCM as the denominator of a random linear combination of reciprocals already used in [P92], [BP94], [CFGH99], and [MSa] although in a different context.

Definition 6.6. The greatest common divisor (GCD) $d_k = d_k(M)$ of all $k \times k$ minors (subdeterminants) of a matrix $M \in \mathbb{Z}^{n\times n}$ is called the k-th *determinant divisor* of M, for $k = 1, \ldots, n$. We write $s_0 = d_0 = 1$ and define the k-th *Smith invariant factor* of M as $s_k = s_k(M) = d_k/d_{k-1}$ for $k = 1, \ldots, n$.

It is well known that for a given matrix $M \in \mathbb{Z}^{n\times n}$ there exist two matrices $P \in \mathbb{Z}^{n\times n}$ and $Q \in \mathbb{Z}^{n\times n}$ such that $|\det(P)| = |\det(Q)| = 1$ and

$$M = P \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{pmatrix} Q.$$

Therefore, if $M \in \mathbb{Z}^{n\times n}$ is nonsingular then $s_n M^{-1} \in \mathbb{Z}^{n\times n}$ too. Due to this, reconstruction of M^{-1} from $M^{-1} \bmod q$ as well as $M^{-1}\mathbf{v}$ from $M^{-1}\mathbf{v} \bmod q$ (for larger q) is trivial if $s_n = s_n(M)$ is available. It is also well know that (see Chapter 2 for the definition of $|M|$)

$$s_n \le |\det(M)| \le (\sqrt{n}|M|)^n.$$

Let $\mathbf{u}_i$ and $\mathbf{v}_i$ be n-dimensional random integer vectors with entries independently uniformly chosen from $\{-K, 1-K, \ldots, K-1\}$, $\mathbf{u}_i^T M^{-1}\mathbf{v}_i = \frac{\eta_i}{\delta_i}$

where $\delta_i > 0$ and $\gcd(\eta_i, \delta_i) = 1$, for $i = 1, 2, 3, \ldots$.

Remark 6.7. (η_i, δ_i) can be reconstructed from $\mathbf{u}_i^T M^{-1} \mathbf{v}_i \bmod q$ for

$$\gcd(q, s_n) = 1 \quad \text{and} \quad q \geq 2K^2 n^{n+2} |M|^{2n} \tag{6.4}$$

at the bit cost $O(\mu(\log q) \log\log q)$. Here q can be a prime, prime power, or product of primes. See Chapter 7 for the choice of q.

Definition 6.8. $\mathrm{ord}_p(n)$ is the largest integer k such that $p^k \mid n$.

Theorem 6.9. *For any prime p and any i, we have*

$$\text{Probability}\{\mathrm{ord}_p(\delta_i) = \mathrm{ord}_p(s_n)\} \geq (1 - \tfrac{1}{2K}\lceil \tfrac{2K}{p} \rceil)^2.$$

Proof. First we observe that $\delta_i \mid s_n$. Then, by [ABM99, Theorem 2], we have

$$\text{Probability}\{s_n M^{-1} \mathbf{v}_i = \mathbf{0} \bmod p\} \leq \tfrac{1}{2K}\lceil \tfrac{2K}{p} \rceil.$$

In addition, for given $\mathbf{w}_i \neq \mathbf{0} \bmod p$, we have

$$\text{Probability}\{\mathbf{u}_i^T \mathbf{w}_i = 0 \bmod p\} \leq \tfrac{1}{2K}\lceil \tfrac{K}{p} \rceil.$$

Therefore,

$$\begin{aligned}
& \text{Probability}\{\mathrm{ord}_p(\delta_i) = \mathrm{ord}_p(s_n)\} \\
\geq\ & \text{Probability}\{\tfrac{s_n \eta_i}{\delta_i} \neq 0 \bmod p\} \\
=\ & \text{Probability}\{\mathbf{u}_i^T s_n M^{-1} \mathbf{v}_i \neq 0 \bmod p\} \ \geq\ (1 - \tfrac{1}{2K}\lceil \tfrac{K}{p} \rceil)^2. \qquad \square
\end{aligned}$$

Corollary 6.10. *If* $K \geq \max(2^{31}, 2^{-56}\log_2 s_n)$ *then*

$$\text{Probability}\{\text{lcm}(\delta_1,\delta_2,\delta_3,\delta_4) = s_n\} > \tfrac{1}{5}.$$

Proof.

$$\begin{aligned}
&\text{Probability}\{\text{lcm}(\delta_1,\delta_2,\delta_3,\delta_4) \neq s_n\} \\
&\leq \sum_{\text{prime } p|s_n} \text{Probability}\{\text{ord}_p(\delta_i) \neq \text{ord}_p(s_n),\ \forall i = 1,2,3,4\} \\
&\leq \sum_{\text{prime } p|s_n} \{1-(1-\tfrac{1}{2K}\lceil\tfrac{2K}{p}\rceil)^2\}^4 = \sum_{\text{prime } p|s_n} \{(2-\tfrac{1}{2K}\lceil\tfrac{2K}{p}\rceil)\tfrac{1}{2K}\lceil\tfrac{2K}{p}\rceil\}^4 \\
&\leq \tfrac{81}{256} + \sum_{\text{odd prime } p|s_n} (\tfrac{1}{K}\lceil\tfrac{2K}{p}\rceil)^4 \leq \tfrac{81}{256} + \sum_{\text{odd prime } p|s_n} (\tfrac{1}{K}+\tfrac{2}{p})^4 \\
&\leq \tfrac{81}{256} + \sum_{\text{odd prime } p|s_n} (\tfrac{1}{K^4}+\tfrac{8}{3K^3}) + (\tfrac{24}{K^2}+\tfrac{32}{3K}+\tfrac{16}{9})\tfrac{1}{p^2} \\
&\leq \tfrac{81}{256} + (\tfrac{1}{K^4}+\tfrac{8}{3K^3})\log_3 s_n + (\tfrac{24}{K^2}+\tfrac{32}{3K}+\tfrac{16}{9})\tfrac{1}{4} < \tfrac{4}{5}.
\end{aligned}$$

□

Chapter 7

Rational Number Reconstruction

A customary approach in computer algebra is to perform computations with rational numbers modulo a large integer q (a prime, prime power, or product of several selected primes) and then to reconstruct the rational output from its value modulo q [GG99]. In particular, the *modular rational number reconstruction* is the final stage of the solution of a nonsingular linear system of n equations by means of p-adic lifting [MC79], [D82], [P02] (see [GG99], [S86], [UP83], [Z93], for other important applications).

Problem 7.1. Compute a pair of integers (η, δ) from three positive integers

m, n, k such that

$$|\eta| < k < m, \quad 1 \leq \delta \leq m/k, \quad \eta = n\delta \bmod m. \tag{7.1}$$

There always exists a solution to Problem 7.1. There are at most two solutions such that $\gcd(\eta, \delta) = 1$, and at most one of them satisfies $|\eta| < k/2$ [GG99, Theorem 5.26]. To ensure correct reconstruction of coprimed η and δ with some upper bounds, say, $|\eta| < M$ and $\delta < N$, we may choose $m \geq 2MN$ and $\gcd(m, \eta) = \gcd(m, \delta) = 1$. Then η and δ can be uniquely recovered from $n = \eta/\delta \bmod m$ by solving the following problem.

Problem 7.2. Compute a pair of integers (η, δ) from four positive integers M, N, m, n, $m \geq 2MN$, such that

$$\gcd(\eta, \delta) = 1, \quad |\eta| < M, \quad 1 \leq \delta \leq N, \quad \eta = n\delta \bmod m. \tag{7.2}$$

The common approach to the solution of the problems of rational number reconstruction is by applying the extended Euclidean algorithm to m and n. Hereafter, we refer to this algorithm as the *EEA*, and we seek faster solution algorithms based on accelerating the EEA. The algorithm produces a sequence of triples (r_j, s_j, t_j), $j = 1, \ldots, l$ (notation used in [GG99]). In our case, we need only the triples $(r_{j-1}, s_{j-1}, t_{j-1})$ and (r_j, s_j, t_j) for a specially

selected j. Extension from computing these triples to the solution to Problems 7.1 and 7.2 is shown in full detail in [GG99, Theorem 5.26]. We show an alternative approach, which is more directly related to our modification of the EEA. Our goal is the acceleration of the EEA and consequently the solution of Problems 7.1 and 7.2. The known algorithms compute the desired pair of the EEA triples and thus solve Problems 7.1 and 7.2 by using $O(d^2)$ bit operations, where $d = \lfloor \log_2 m \rfloor$, $m \geq n$. We speed up the computation by the factor of almost d; that is, we decrease the above bit cost bound to the level

$$\rho(d) = O(\mu(d) \log d). \tag{7.3}$$

A similar acceleration is known for the Euclidean algorithm applied to polynomials [M73], [AHU74], [BGY80], but in the integer case a well known additional difficulty is due to the carries. Among the known methods, only the Knuth-Schönhage algorithm [S71] has settled the problem for integers but only in the special case in which $j = l$ and the triple (r_l, s_l, t_l) terminates the Euclidean algorithm, that is, where r_l is the GCD.

In the next sections, we overcome the difficulty and come out with a desired algorithm, which solves the GCD problem as a special case. Our construction relies on computing a matrix sequence $\{Q_i,\ i = 0, 1, \ldots\}$, which

represents the quotients and cofactors computed in the EEA rather than on computing just the remainder sequence $\{r_i,\ i = 0, 1, \ldots\}$. This enables a simpler control over the growth of the magnitude of the entries of the Q_i than we would have had over the decrease of the r_i.

7.1 Extended Euclidean Algorithm

Algorithm 7.3 (Euclidean Algorithm).

Input: A pair of natural numbers (m, n), $m \geq n$.

Output: $\gcd(m, n)$.

Computation: Write $r_0 = m$, $r_1 = n$. Compute

$$r_{i+1} = r_{i-1} \bmod r_i$$

for $i = 1, 2, \ldots, l$ until $r_{l+1} = 0$. Output r_l.

For $i = 1, 2, \ldots, l$, let

$$\begin{pmatrix} r_{i-1} \\ r_i \end{pmatrix} = P_i \begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix}$$

where

$$P_i = \begin{pmatrix} q_i & 1 \\ 1 & 0 \end{pmatrix}, \quad q_i = \lfloor r_{i-1}/r_i \rfloor,$$

$$Q_i = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} = P_1 P_2 \cdots P_i.$$

$$Q_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad Q_{l+1} = \begin{pmatrix} \infty & \infty \\ \infty & \infty \end{pmatrix}.$$

The sequence $\{r_i\}_{i=0}^{l}$ is called the *remainder sequence*, and the sequence $\{Q_i\}_{i=0}^{l}$ is called the *matrix sequence*. The *extended Euclidean algorithm (EEA)* outputs both sequences $\{r_i\}_{i=0}^{l}$ and $\{Q_i\}_{i=0}^{l}$.

For a given pair (m, n) and the sequence $\{Q_i\}$, we can immediately compute the sequence $\{r_i\}$ because

$$\det(P_i) = -1, \quad \det(Q_i) = (-1)^i, \tag{7.4}$$

$$\begin{pmatrix} m \\ n \end{pmatrix} = Q_i \begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix}, \quad Q_i^{-1} = (-1)^i \begin{pmatrix} d_i & -b_i \\ -c_i & a_i \end{pmatrix} \tag{7.5}$$

for all $i = 1, 2, \ldots, l$.

Our main task is to solve the following problem.

Problem 7.4 (Selected Output of the EEA).

Input: Integers m, n, h such that $m \geq n \geq 1$, $h \geq 0$.

Output: The unique Q_i such that $|Q_i| \leq 2^h < |Q_{i+1}|$.

In the remaining part of this section we state some simple auxiliary properties of the remainders r_i and the matrices Q_i.

Theorem 7.5. *$r_i > r_{i+1} > 0$, $r_i \geq r_{i+1} + r_{i+2}$ for $i = 0, 1, \ldots, l-1$.*

Theorem 7.6.

(i) $b_i = a_{i-1}$, $d_i = c_{i-1}$ *for* $i = 1, 2, \ldots, l$.

(ii) $a_i = a_{i-1}q_i + a_{i-2} > a_{i-1}$, $c_i = c_{i-1}q_i + c_{i-2} > c_{i-1}$ *for* $i = 2, 3, \ldots, l$.

(iii) $a_{i-2} = a_i \bmod a_{i-1}$, $c_{i-2} = c_i \bmod c_{i-1}$ *for* $i = 3, 4, \ldots, l$.

(iv) $a_0 > c_0$, $a_1 \geq c_1$, $a_i > c_i$ *for* $i = 2, 3, \ldots, l$.

Corollary 7.7. *Q_{i-1} can be computed from Q_i by Theorem 7.6 (i), (iii).*

Corollary 7.8.

(i) $|Q_i| = a_i$ *for* $i = 0, 1, \ldots, l$.

(ii) $|Q_i| \geq |Q_{i-1}| + |Q_{i-2}|$ *for* $i = 2, 3, \ldots, l$.

Corollary 7.9. *$m/2 < r_i|Q_i| \leq m$ for $i = 0, 1, \ldots, l$.*

Remark 7.10. Note an equivalent customary representation of the EEA's output by the sequences $\{r_i\}$, $\{s_i\}$, $\{t_i\}$ (with the notation in [GG99]), where $s_i = (-1)^i d_i$, $t_i = (-1)^{i-1} b_i$.

7.2 EEA for Modified Input

To accelerate the solution of Problem 7.4, we apply the divide-and-conquer techniques. Roughly, the idea is to solve Problem 7.4 in two steps. In each step, Problem 7.4 is solved for h replaced by $\lfloor h/2 \rfloor$, and the output of the first step is used as the input of the second step. We are going to show that

(i) this leads to the same desired output, and

(ii) the computational cost of the reduction to the pair of half-size problems is small.

A basic observation is that the matrix sequence $\{Q_i\}$ depends only on the quotient m/n. That is, for another input values m^* and n^* such that $m^*/n^* = m/n$, the Euclidean algorithm computes the same matrices $Q_i^* = Q_i$ for all i. A relatively small perturbation of the quotient m/n should not affect the first several terms of the sequence $\{Q_i\}$, using which is enough to solve the problem for smaller h. That is, we may replace m and n by smaller integers m^* and n^* provided that $m^*/n^* \approx m/n$. For the input values m^* and n^*, we denote by $\{r_i^*\}$ the remainder sequence and by $\{Q_i^*\}$ the matrix sequence. Next, we specify some bounds on the allowed perturbations of m/n for which $Q_i = Q_i^*$ and then state our main theorem.

Theorem 7.11. *Suppose* $m^* = \lfloor m/\lambda \rfloor$ *and* $n^* = \lfloor n/\lambda \rfloor$ *for a positive integer* λ. *For any given integer* i, *if*

$$r^*_{i+2} \geq |Q^*_{i+1}| \quad or \quad r_{i+2} \geq \lambda |Q_{i+1}|$$

then $Q_i = Q^*_i$.

Proof. (i) Suppose $r^*_{i+2} \geq |Q^*_{i+1}|$. Write $\binom{u_j}{v_j} = Q^{*-1}_j \binom{m}{n}$ for $j = 0, 1, \ldots, i+1$. Then we have

$$\begin{pmatrix} u_{j+1} \\ v_{j+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q^*_{j+1} \end{pmatrix} \begin{pmatrix} u_j \\ v_j \end{pmatrix}.$$

Therefore, $u_{j+1} = v_j$ for $j = 0, 1, \ldots, i$. Furthermore, extending (7.5) to (m^*, n^*), we obtain that

$$\begin{pmatrix} r^*_j \\ r^*_{j+1} \end{pmatrix} = Q^{*-1}_j \begin{pmatrix} m^* \\ n^* \end{pmatrix},$$

$$\begin{pmatrix} u_j \\ v_j \end{pmatrix} = \begin{pmatrix} r^*_j \\ r^*_{j+1} \end{pmatrix} \lambda + Q^{*-1}_j \begin{pmatrix} m - m^*\lambda \\ n - n^*\lambda \end{pmatrix}.$$

By (7.5) we also know that, in each row of Q^{*-1}_j, one of the entries is non-negative, and another is non-positive, and their absolute values are bounded by $|Q^*_{j-1}|$ in the first row and by $|Q^*_j|$ in the second row. Therefore, we have

$$v_j > (r^*_{j+1} - |Q^*_j|)\lambda$$

and

$$u_j - v_j > (r^*_j - |Q^*_{j-1}|)\lambda - (r^*_{j+1} + |Q^*_j|)\lambda \geq (r^*_{j+2} - |Q^*_{j+1}|)\lambda.$$

So, by assumption, $u_j > v_j > 0$ for $j = 1, 2, \ldots, i$. Now we have $u_0 = m$, $u_1 = n$, $u_{j+1} = u_{j-1} \bmod u_j$ for $j = 1, 2, \ldots, i$. So $u_j = r_j$ and $Q_j = Q_j^*$ for $j = 0, 1, \ldots, i$.

(ii) Suppose $r_{i+2} \geq \lambda |Q_{i+1}|$. Write $\binom{x_j}{y_j} = Q_j^{-1}\binom{m^*}{n^*}$ for $j = 0, 1, \ldots, i+1$. Then we have

$$\begin{pmatrix} x_{j+1} \\ y_{j+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & -q_{j+1} \end{pmatrix} \begin{pmatrix} x_j \\ y_j \end{pmatrix}.$$

Therefore, $x_{j+1} = y_j$ for $j = 0, 1, \ldots, i$. Furthermore, by (7.5), we extend the above expression for x_j and y_j as follows:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} r_j \\ r_{j+1} \end{pmatrix} \lambda^{-1} - Q_j^{-1} \begin{pmatrix} m/\lambda - m^* \\ n/\lambda - n^* \end{pmatrix}.$$

Now, similarly to in part (i), we deduce that $y_j > r_{j+1}\lambda^{-1} - |Q_j|$ and $x_j - y_j > (r_j\lambda^{-1} - |Q_{j-1}|) - (r_{j+1}\lambda^{-1} + |Q_j|) \geq (r_{j+2}\lambda^{-1} - |Q_{j+1}|)$. So, by assumption, $x_j > y_j > 0$ for $j = 1, 2, \ldots, i$. Now we have $x_0 = m^*$, $x_1 = n^*$, $x_{j+1} = x_{j-1} \bmod x_j$ for $j = 1, 2, \ldots, i$. So $x_j = r_j^*$ and $Q_j = Q_j^*$ for $j = 0, 1, \ldots, i$.

□

Corollary 7.12. *Suppose $m^* = \lfloor m/\lambda \rfloor$, $n^* = \lfloor n/\lambda \rfloor$ for a positive integer λ. For any given integer i, if*

$$m^* \geq 2|Q_{i+2}^*| \cdot |Q_{i+1}^*| \quad \text{or} \quad m \geq 2\lambda |Q_{i+2}| \cdot |Q_{i+1}|,$$

then $Q_i = Q_i^*$.

Proof. Combine the assumed bound on m^* and m with the first inequality of Corollary 7.9 extended also to m^*, r_i^*, Q_i^* and arrive at the bounds on r_{i+2}^* and r_{i+2} in Theorem 7.11. □

Theorem 7.13. *Suppose* $m^* = \lfloor m/\lambda \rfloor$, $n^* = \lfloor n/\lambda \rfloor$ *for a positive integer* λ, *and* K *is a given positive integer such that* $m^* \geq 2K^2$. *If* $|Q_i^*| \leq K < |Q_{i+1}^*|$, *then* $Q_j = Q_j^*$ *for all* $j \leq i-2$ *and* $|Q_j| \leq K < |Q_{j+1}|$ *for some* j *such that* $i-2 \leq j \leq i+2$.

Proof. By Corollary 7.12, we have $Q_j = Q_j^*$ for $j \leq i-2$. If $|Q_{i+3}| > K$, then we are done. Otherwise, we have $m \geq \lambda m^* \geq 2\lambda K^2 \geq 2\lambda |Q_{i+3}|^2$. By applying Corollary 7.12 again, we obtain $Q_{i+1} = Q_{i+1}^*$, $Q_i = Q_i$. □

Theorem 7.13 leads to the following algorithm.

Algorithm 7.14 (Selected Output of the EEA).

Input: A triple of integers (m, n, h) such that $m \geq n > 0, h \geq 0$.

Output: The unique matrix Q_k such that $|Q_k| \leq 2^h < |Q_{k+1}|$.

Computation: Let $d = \lfloor \log m \rfloor$.

1. When $h \leq \lfloor d/2 \rfloor - 1$, let $\lambda = 2^{d-2h-1}$, $m^* = \lfloor m/\lambda \rfloor$, and $n^* = \lfloor n/\lambda \rfloor$; then $2^{2h+1} \leq m^* \leq m/2$. We first apply the algorithm to the input (m^*, n^*, h) and have the output Q_i^*. Theorem 7.13 for $K = 2^h$ implies that $Q_{i-2} = Q_{i-2}^*$ and $|Q_k| \leq 2^h < |Q_{k+1}|$ for some $i-2 \leq k \leq i+2$. We may compute $Q_{i-2} = Q_{i-2}^*$ from Q_i^* (cf. Corollary 7.7) and then find Q_k in a few Euclidean steps.

2. When $\lfloor d/2 \rfloor \leq h \leq d-1$, we first apply the algorithm to find $|Q_i| \leq 2^{\lfloor h/2 \rfloor} < |Q_{i+1}|$. Next we apply the algorithm again for the input $(r_i, r_{i+1}, \lfloor h/2 \rfloor)$ and have the output $\tilde{Q}_j$. Now we have $Q_{i+j} = Q_i \tilde{Q}_j$, $|Q_{i+j}| < 2^{h+1}$, and $|Q_{i+j+2}| > 2^{h-1}$. Then $|Q_k| \leq 2^h < |Q_{k+1}|$ for some $i+j-2 \leq k \leq i+j+2$, and we may find Q_k in a few Euclidean steps.

3. When $h \geq d$, we first apply the algorithm to find $|Q_i| \leq 2^{d-1} < |Q_{i+1}|$. Then $|Q_k| \leq 2^h < |Q_{k+1}|$ for some $i \leq k \leq i+4$, and we may find Q_k in a few Euclidean steps.

Theorem 7.15. *Let $f(d,h)$ be the bit cost of performing Algorithm 7.14 for the input (m,n,h), where $d = \lfloor \log m \rfloor$. Then we have*

$$f(d,h) = O(\mu(d) \log h)$$

Proof. By inspection of the algorithm, we have

$$f(d,h) = \begin{cases} f(2h+1,h) + O(\mu(d)) & \text{if } h \le \lfloor \frac{d}{2} \rfloor - 1, \\ f(d, \lfloor \frac{h}{2} \rfloor) + f(d - \lfloor \frac{h}{2} \rfloor, \lfloor \frac{h}{2} \rfloor) + O(\mu(d)) & \\ & \text{if } \lfloor \frac{d}{2} \rfloor \le h \le d-1, \\ f(d,d-1) + O(\mu(d)) & \text{if } h \ge d. \end{cases}$$

Let us write $F(h) = f(2h+1,h)$. Then

$$F(h) = 2F(\lfloor h/2 \rfloor) + O(\mu(2h)),$$

and we obtain that

$$F(h) = O(\mu(2h)\log h).$$

By recursively combining this bound with the above expressions for $f(d,h)$, we obtain

$$f(d,h) = \sum_{i=1}^{1+\lfloor \log h \rfloor} (F(\lfloor h/2^i \rfloor) + O(\mu(d))) = O(\mu(d)\log h). \quad \square$$

Remark 7.16.

(i) We may easily extend Algorithm 7.14 to compute the matrix Q_i (at the bit cost $O(\mu(d)\log\log K)$), such that $|Q_i| \le K < |Q_{i+1}|$ for any real $K \ge 1$, not just for $K = 2^h$.

(ii) We may also easily extend Algorithm 7.14 to find the remainder r_i (at the bit cost $O(\mu(d)\log\log(m/K))$, such that $r_i \geq K > r_{i+1}$ for any real $1 \leq K \leq m$. By choosing $K = 1$, we compute $r_i = \gcd(m, n)$.

7.3 Rational Number Reconstruction

Let us next extend Algorithm 7.14 to solve Problems 7.1 and 7.2. Note that (cf. (7.5))

$$\begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix} = Q_i^{-1} \begin{pmatrix} m \\ n \end{pmatrix} = (-1)^i \begin{pmatrix} d_i & -b_i \\ -c_i & a_i \end{pmatrix} \begin{pmatrix} m \\ n \end{pmatrix}.$$

Therefore,

$$(-1)^i r_{i+1} = na_i \bmod m \text{ for all } i.$$

Solution of Problem 7.2. Let i be such that $a_i \leq m/k < a_{i+1}$. Since $a_i \leq m/k$ and $r_{i+1} \leq \frac{m}{a_{i+1}} < k$, we obtain a solution $\big((-1)^i r_{i+1},\ a_i\big)$. □

Solution of Problem 7.2. Let i be such that $a_i \leq N < a_{i+1}$. If $r_{i+1} \leq M$, then $(\eta, \delta) = \big((-1)^i r_{i+1},\ a_i\big)$. Otherwise, $r_{i+1} > M$ and $\eta/\delta \neq (-1)^i r_{i+1}/a_i$. Therefore,

$$(\eta, \delta) = \big((-1)^{i-1}(r_i - t r_{i+1}),\ a_{i-1} + t a_i\big)$$

for some $t \in \mathbb{R}$. Note that $a_{i-1} + ta_i \in \mathbb{Z}$, $\frac{n\delta-\eta}{m} = c_{i-1} + tc_i \in \mathbb{Z}$, and $\gcd(a_i, c_i) = 1$, so $t \in \mathbb{Z}$. Therefore, (η, δ) is defined by the unique integer t such that $r_i - tr_{i+1} \leq M$ and $a_{i-1} + ta_i \leq N$. □

Corollary 7.17. *Problems 7.1 and 7.2 can be solved by using* $O(d(\log d)^2 \log\log d)$ *bit operations.*

Bibliography

[A87] S. D. Ashby, *Polynomial preconditioning for conjugate gradient methods*, Ph.D. thesis, Department of Computer Science, University of Illinois Urbana-Champaign, 1987.

[ABM99] J. Abbott, M. Bronstein, T. Mulders, Fast Deterministic Computations of the Determinants of Dense Matrices, *Proc. of International Symposium on Symbolic and Algebraic Computation (ISSAC'99)*, pp.197–204, ACM Press, New York, 1999.

[AG91] G. S. Ammar, P. Gader, A Variant of the Gohberg-Semencul Formula Involving Circulant Matrices, *SIAM Journal on Matrix Analysis and Applications*, 12, 3 (1991), pp.534–541.

[AHU74] A. V. Aho, J. E. Hopcroft, J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison–Wesley, Reading, MA, 1974.

[AMS89] S. F. Ashby, T. A. Manteuffel, P. E. Saylor, Adaptive polynomial preconditioning for Hermitian indefinite linear systems, *BIT*, 29 (1989), pp.583–609.

[AMO92] S. F. Ashby, T. A. Manteuffel, J. S. Otto, A Comparison of Adaptive Chebyshev and Least Squares Polynomial Preconditioning for Hermitian Positive Definite Linear Systems, *SIAM J. Sci. Statist. Comput.*, 13 (1992), pp.1–29.

[BA80] R. R. Bitmead, B. D. O. Anderson, Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations, *Linear Algebra and Its Applications*, 34 (1980), pp.103–116.

[BGR90] J. A. Ball, I. Gohberg, L. Rodman, Interpolation of Rational Matrix Functions, *Operator Theory: Advances and Applications*, 45, Birkhäuser, Basel, 1990.

[BGY80] R. P. Brent, F. G. Gustavson, D. Y. Y. Yun, Fast solution of Toeplitz systems of equations and computation of Padé approximations, *J. Algorithms*, 1 (1980), pp.259–295.

[B-I66] A. Ben-Israel, A note on iterative method for generalized inversion of matrices, *Mathematics of Computation*, 20 (1966), pp.439–440.

[B-IC66] A. Ben-Israel, D. Cohen, On Iterative Computation of Generalized Inverses and Associated Projections, *SIAM Journal on Numerical Analysis*, 3 (1966), pp.410–419.

[BP93] D. A. Bini, V. Y. Pan, Improved Parallel Computations with Toeplitz-like and Hankel-like Matrices, *Linear Algebra and Its Applications*, 188/189 (1993), pp.3–29.

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

[BVB97] A. Bultheel, M. Van Barel, *Linear Algebra, Rational Approximation and Orthogonal Polynomials*, Vol.6 of *Studies in Computational Mathematics*, North-Holland, Elsevier Science, Amsterdam, 1997.

[CFGH99] G. Cooperman, S. Feisel, J. von zur Gathen, G. Havas, GCD of Many Integers, *Computing and Combinatorics, Lecture Notes in Computer Science*, 1627, pp.310–317, Springer, Berlin, 1999.

[CJL96] S. Cabay, A. Jones, G. Labahn, Computation of Numerical Padé-Hermite and Simultaneous Padé Systems, *SIAM Journal on Matrix Analysis and Applications*, 17 (1996), pp.248–297.

[CPW74] R. E. Cline, R. J. Plemmons, G. Worm, Generalized Inverses of Certain Toeplitz Matrices, *Linear Algebra and Its Applications*, 8 (1974), pp.25–33.

[D82] J. D. Dixon, Exact solution of linear equations using p-adic expansions, *Numer. Math.*, 40 (1982), pp.137–141.

[EGV00] W. Eberly, M. Giesbrecht, G. Villard, On Computing the Determinant and Smith Form of an Integer Matrix, *Proc. 41st Annual Symposium on Foundations of Computer Science (FOCS'2000)*, pp.675–685, IEEE Computer Society Press, Los Alamitos, California, 2000.

[FF63] D. K. Faddeev, V. N. Faddeeva, *Computational Methods of Linear Algebra*, W. H. Freeman, San Francisco, 1963.

[G98] M. Gu, Stable and Efficient Algorithms for Structured System of Linear Equations, *SIAM Journal on Matrix Analysis and Applications*, 19, 2 (1998), pp.279–306.

[G99] M. Gu, New Fast Algorithms for Structured Linear Least Squares Problems, *SIAM Journal on Matrix Analysis and Applications*, 20 (1999), pp.244–269.

[GG99] J. von zur Gathen, J. Gerhard, *Modern Computer Algebra*, Cambridge University Press, Cambridge, UK, 1999.

[GKKL87] I. Gohberg, T. Kailath, I. Koltracht, P. Lancaster, Linear Complexity Parallel Algorithms for Linear Systems of Equations with Recursive Structure, *Linear Algebra and Its Applications*, 88/89 (1987), pp.271–315.

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

[GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 1989 (2nd edition), 1996 (3rd edition).

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

[GO94] I. Gohberg, V. Olshevsky, Complexity of Multiplication with Vectors for Structured Matrices, *Linear Algebra and Its Applications*, 202 (1994), pp.163–192.

[GO94a] I. Gohberg, V. Olshevsky, Fast State-Space Algorithms for Matrix Nehari and Nehari-Takagi Interpolation Problems, *Integral Equations and Operator Theory*, 20, 1 (1994), pp.44–83.

[H82] L. K. Hua, *Introduction to Number Theory*, Springer-Verlag, Berlin, 1982.

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

[HR84] G. Heinig, K. Rost, *Algebraic Methods for Toeplitz-like Matrices and Operators* in *Operator Theory*, 13, Birkhäuser, 1984.

[HS52] M. R. Hestenes, E. Stiefel, Methods of Conjugate Gradients for Solving Linear Systems, *J. Res. Nat. Bur. Stand.*, 49 (1952), pp.409–436.

[JMP83] O. Johnson, C. Micchelli, G. Paul, Polynomial preconditioning for conjugate gradient calculation, *SIAM J. Numer. Anal.*, 20 (1983), pp.362–376.

[KKM79] T. Kailath, S. Y. Kung, M. Morf, Displacement Ranks of Matrices and Linear Equations, *Journal of Mathematical Analysis and Applications*, 68, 2 (1979), pp.395–407.

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

[KS95] T. Kailath, A. H. Sayed, Displacement Structure: Theory and Applications, *SIAM Review*, 37, 3 (1995), pp.297–386.

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

[KVB99] P. Kravanja, M. Van Barel, Algorithms for Solving Rational Interpolation Problems Related to Fast and Superfast Solvers for Toeplitz Systems, *Proceedings of Advanced Signal Processing Algorithms, Architecture and Implementation IX*, pp.359–370, Denver, Colorado, SPIE Publications, 1999.

[M73] R. Moenck, *Fast computation of GCDs*, in *Proceedings of 5th ACM Annual Symposium on Theory of Computing*, pp.142–171, ACM Press, New York, 1973.

[M74] M. Morf, Fast Algorithms for Multivariable Systems, *Ph.D. Thesis*, Department of Electrical Engineering, Stanford University, Stanford, CA, 1974.

[M80] M. Morf, Doubling Algorithms for Toeplitz and Related Equations, *Proceedings of IEEE International Conference on ASSP*, pp.954–959, IEEE Press, Piscataway, New Jersey, 1980.

[MC79] R. T. Moenck, J. H. Carter, *Approximate algorithms to derive exact solutions to systems of linear equations*, in *Proceedings of EUROSAM, Lecture Notes in Comput. Sci.*, 72, pp.63–73, Springer-Verlag, Berlin, 1979.

[MSa] T. Mulders, A. Storjohann. Certified Dense Linear System Solving, preprint, 2001.

[OP98] V. Olshevsky, V. Y. Pan, A Unified Superfast Algorithm for Boundary Rational Tangential Interpolation Problem, *Proceedings of 39th Annual IEEE Symposium Foundations of Computer Science*, pp.192–201, IEEE Computer Society Press, 1998.

[OS99] V. Olshevsky, M. A. Shokrollahi, A Unified Superfast Algorithm for Confluent Tangential Interpolation Problem and for Structured Matrices, *Proceedings of Advanced Signal Processing Algorithms, Architecture and Implementation IX*, Denver, Colorado, SPIE Publications, 1999.

[OS00] V. Olshevsky, M. A. Shokrollahi, A Superfast Algorithm for Confluent Rational Tangential Interpolation Problem via Matrix-Vector Multiplication for Confluent Cauchy-like Matrices, *Proceedings of 14th International Symposium on Mathematical Theory of Networks and Systems (MTNS'2000)*, University of Perpignan, Perpignan, France, 2000.

[P87] V. Y. Pan, Complexity of Parallel Matrix Computations, *Theoretical Computer Science*, 54 (1987), pp.65–85.

[P88] V. Y. Pan, Computing the Determinant and the Characteristic Polynomials of a Matrix via Solving Linear System of Equations. *Information Processing Letters*, 28 (1988), pp.71–75.

[P90] V. Y. Pan, On Computations with Dense Structured Matrices, *Mathematics of Computation*, 55, 191 (1990), pp.179–190. Proceedings version: *Proceedings of International Symposium on Symbolic*

and Algebraic Computation (ISSAC'89), pp.34–42, ACM Press, New York, 1989.

[P91] V. Y. Pan, Complexity of Algorithms for Linear Systems of Equations, *Computer Algorithms for Solving Linear Algebraic Equations (The State of the Art)*, edited by E. Spedicato, *NATO ASI Series, Series F: Computer and Systems Sciences*, 77, pp.27–56, Springer, Berlin, 1991.

[P92] V. Y. Pan, Parallel Solution of Toeplitz-like Linear Systems, *Journal of Complexity*, 8 (1992), pp.1–21.

[P93] V. Y. Pan, Decreasing the Displacement Rank of a Matrix, *SIAM Journal on Matrix Analysis and Applications*, 14, 1 (1993), pp.118–121.

[P99] V. Y. Pan, A Unified Superfast Divide-and-Conquer Algorithm for Structured Matrices over Abstract Fields, *MSRI Preprint No.1999-033*, Mathematical Sciences Research Institute, Berkeley, California, 1999.

[P00] V. Y. Pan, Nearly Optimal Computations with Structured Matrices, *Proceedings of 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2000)*, pp.953–962, ACM Press, New York, and SIAM Publications, Philadelphia, 2000.

[P01] V. Y. Pan, *Structured Matrices and Polynomials: Unified Superfast Algorithms*, Birkhäuser/Springer, Boston/New York, 2001.

[P02] V. Y. Pan, *Can we optimize Toeplitz/Hankel computations?* in Proceedings of Annual Conference on Algebraic and Symbolic Computation (CASC'02), E. W. Mayr, V. G. Ganzha, E. V. Vorozhtzov, editors, Technische Universitat München, Germany, 2002, pp.253–264.

[PKRCa] V. Y. Pan, M. Kunin, R. E. Rosholt, H. Cebecioğlu. Residual correction algorithms for general and structured matrices, 2002. Preprint.

[PRW00] V. Y. Pan, Y. Rami, X. Wang, Newton's Iteration for Structured Matrices, *Proceedings of 14th International Symposium on Mathematical Theory of Networks and Systems (MTNS'2000)*, University of Perpignan Press, Perpignan, France, June 2000.

[PRW01] V. Y. Pan, Y. Rami, X. Wang, Structured Matrices and Newton's Iteration: Unified Approach, *Linear Algebra and Its Applications*, 343/344 (2001), pp.233–265.

[PS91] V. Y. Pan, R. Schreiber. An improved Newton iteration for the generalized inverse of a matrix, with applications. *SIAM Journal on Scientific and Statistical Computing*, 12, 5 (1991), pp.1109–1131.

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

[PW02] V. Y. Pan, X. Wang, Acceleration of Euclidean algorithm and extensions, *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation*, Teo Mara, ed., ACM, New York, 2002, pp.207–213.

[PW03] V. Y. Pan, X. Wang, Inversion of Displacement Operators, *SIAM Journal on Matrix Analysis and Applications*, 24, 3 (2003), pp.660–677.

[R71] J. K. Reid, On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations, *Large Sparse Sets of Linear Equations*, ed. J. K. Reid, Academic Press, New York, pp.231–254, 1971.

[S33] G. Schultz, Iterative Berechnung der Reciproken Matrix, *Z. Angew. Meth. Mech.*, 13 (1933), pp.57–59.

[S71] A. Schönhage, Schnelle Berechnung von Kettenbruchentwicklungen, *Acta Inform.*, 1 (1971), pp.139–144.

[S85] Y. Saad, Practical use of polynomial preconditionings for the conjugate gradient method, *SIAM J. Sci. Statist. Comput.*, 6, 4 (1985), pp.865–881.

[S86] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[SS71] A. Schönhage, V. Strassen, Schnelle Multiplikation grosse Zahlen, *Computing*, 7 (1971), pp.281–292.

[SS74] T. Söderström, G. W. Stewart, On the numerical properties of an iterative method for computing the Moore–Penrose generalized inverse, *SIAM Journal on Numerical Analysis*, 11 (1974), pp.61–74.

[UP83] S. Ursic, C. Patarra, *Exact solution of systems of linear equations with iterative methods*, SIAM J. Algebraic Discrete Methods, 4 (1983), pp.111–115.

[W93] D. H. Wood, Product Rules for the Displacement of Nearly-Toeplitz Matrices, *Linear Algebra and Its Applications*, 188/189 (1993), pp.641–663.

[WP03] X. Wang, V. Y. Pan, Acceleration of Euclidean Algorithm and Rational Number Reconstruction, *SIAM Journal on Computing*, 32, 2 (2003), pp.548556.

[Z93] R. E. Zippel, *Effective Polynomial Computation*, Kluwer Academic Publishes, Norwell, MA, 1993.