ELSEVIER SCIENCE B.V. [DTD 4.1.0]
JOURNAL **LAA** ARTICLE No. **7028**
PAGES **1-33** DISPATCH **May 7, 2001**
in all correspondence concerning this paper refer to: **LAA 7028**
PROD. TYPE: **DISK**
A

ELSEVIER
Linear Algebra and its Applications 00 (2001) 000–000
LINEAR ALGEBRA AND ITS APPLICATIONS
www.elsevier.com/locate/laa

Structured matrices and Newton's iteration: unified approach☆

Victor Y. Pan [a,*], Youssef Rami [b], Xinmao Wang [b]

[a] *Department of Mathematics and Computer Science, Lehman College, CUNY, Bronx, NY 10468, USA*
[b] *Ph.D. Program in Mathematics, Graduate School, CUNY, New York, NY 10036, USA*

Received 21 September 2000; accepted 26 March 2001

Submitted by A. Sayed

Abstract

Recent progress in the study of structured matrices shows advantages of unifying the treatment of various classes of such matrices. We recall some fundamental techniques for such a unification and then specify it in full details for Newton's iteration, which rapidly improves an initial approximation to the inverse matrix by performing two matrix multiplications per recursive step. The iteration is particularly suitable for $n \times n$ structured matrices, represented with $\mathrm{O}(n)$ entries of their short generators rather than with their own n^2 entries. Based on such a representation, matrix operations are performed much more rapidly and use much less memory space. A major problem is to control the length of the generators, which tends to grow quite rapidly in the iterative process. Two known methods solve this problem for Toeplitz-like and Cauchy-like matrices. We extend both methods to a more general class of structured matrices and estimate the convergence rate as well as the computational complexity. Some novel techniques are introduced in this study, in particular for the estimation of the norms of the inverse displacement operators. © 2001 Elsevier Science Inc. All rights reserved.

AMS classification: 65F10; 65F30; 47N40; 47B35; 15A09

Keywords: Structured matrices; Displacement rank; Newton's iteration for matrix inversion; Inverse displacement operators

☆ Supported by NFS Grant CCR 9732206 and PSC CUNY Awards 61393-0030 and 62435-0031.

[*] Corresponding author.

E-mail addresses: vpan@lehman.cuny.edu (V.Y. Pan), ryoussef.geo@yahoo.com (Y. Rami), xwang2@gc.cuny.edu (X. Wang).

PII: S 0 0 2 4 - 3 7 9 5 (0 1) 0 0 3 3 6 - 6

1. Introduction

1.1. *Four basic classes of structured matrices and their four basic properties*

In Table 1, we display four basic classes of structured matrices, which themselves are highly important in numerous applications to sciences, engineering, and communication and also have been naturally extended in terms of the associated operators to cover several other popular and important classes of structured matrices.

The matrices of the four classes of Table 1:

(1) are represented with a few parameters (from m to $m + n$ for an $m \times n$ matrix),
(2) can be multiplied by vectors much faster than general matrices,
(3) are closely related to some operations with polynomials, and
(4) can be naturally associated with some linear operators of shift and scaling.

We refer the reader to [4,32] on property (3), will specify properties (1) and (2) in Table 2, will extend them to more general classes of structured matrices in Section 1.4, and will comment below on property (4). The latter property as well as properties (1) and (2) characterizes the more general class of structured matrices.

1.2. *The displacement rank approach and our main subject*

The modern study of structured matrices was largely motivated by the seminal paper [18] and, in particular, by the basic concept of the *displacement rank* introduced there. The idea was to measure the Toeplitz-like (or Hankel-like) structure of a matrix M by the rank of its displacement, that is, of the image matrix of some linear shift operators applied to the matrix M.

The rank of the displacement (called the displacement rank) of M is at most 2 for Toeplitz (and Hankel) matrices, and an $m \times n$ matrix M is said to be of *Toeplitz* (or *Hankel*) *type* or, alternatively, to be *Toeplitz-like* (or *Hankel-like*) if the rank r of its displacement is small (say, bounded by a small constant independent of m and n). In this case, the matrix can be represented by $(m + n)r$ entries of its short displacement generators rather than by its own mn entries. This enables more efficient storage of such matrices in computer memory as well as much faster computations with them [22].

Toeplitz-like and Hankel-like matrices are omnipresent in scientific and engineering computations, but there are other popular matrix structures too.

Several important classes of structured matrices can be defined and treated similarly in a unified way based on their association with other linear operators, in particular the scaling operators of multiplication by diagonal matrices and the operators that combine scaling and shifts. There are conceptual and computational benefits of

Table 1
Four classes of structured matrices

Toeplitz matrices, $T = [t_{i-j}]_{i,j=0}^{n-1}$	Hankel matrices, $H = [h_{i+j}]_{i,j=0}^{n-1}$
$\begin{bmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{bmatrix}$	$\begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & \iddots & \iddots & h_n \\ \vdots & \iddots & \iddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix}$
Vandermonde matrices, $V = [t_i^j]_{i,j=0}^{n-1}$	Cauchy matrices, $C = [\frac{1}{s_i - t_j}]_{i,j=0}^{n-1}$
$\begin{bmatrix} 1 & t_0 & \cdots & t_0^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{n-1} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{s_0 - t_0} & \cdots & \frac{1}{s_0 - t_{n-1}} \\ \vdots & & \vdots \\ \frac{1}{s_{n-1} - t_0} & \cdots & \frac{1}{s_{n-1} - t_{n-1}} \end{bmatrix}$

Table 2
Parameters and flops counts for matrix representation and its multiplication by a vector

Matrix class	Number of parameters in the $m \times n$ matrix	Number of flops required for multiplication by a vector
General	mn	$2mn - m - n$
Toeplitz / Hankel	$m + n - 1$	$\mathrm{O}((m + n) \log n)$
Vandermonde	n	$\mathrm{O}((m + n) \log^2 n)$
Cauchy	$m + n$	$\mathrm{O}((m + n) \log^2 n)$

unified treatment of various classes of structured matrices, where the operators are unspecified and viewed as symbolic and where the computations with matrices are performed with their displacements, that is, with the images of the operators applied to the matrices (see [4,24,26,31,32]). This will be our approach in the present paper.

The approach can be applied to various algorithms for various computational problems [26,32]. Presently, however, we will narrow our goal to the detail study of Newton's iteration for the computation of the inverse of a structured input matrix and will analyze the resulting algorithms.

Strong numerical stability of Newton's iteration is well-known (see, e.g., [38]); furthermore, the iteration becomes particularly effective where the input matrix is structured. In this case, the main basic operation of matrix multiplication is simplified dramatically and the entire computation uses much smaller memory and computer time than for general matrices provided that the matrix structure is preserved throughout the computation. The preservation of matrix structure, however, is a nontrival problem, which will be our main subject.

1.3. Our results

We present Newton's iteration for the computation of the inverses or generalized inverses of various structured input matrices. Each iteration step is reduced to two multiplications of structured matrices. We elaborate two methods that preserve matrix structure throughout the computation, based either on truncation of the smallest singular values of the displacement or on the substitution of approximate inverses for the inverse matrix into its displacement expression. With each of the techniques, every iteration step is performed by using nearly linear time and nearly optimal memory space, in line with the estimates of Table 2. We also prove superlinear convergence with each of these techniques. The algorithms and their analysis are presented in terms of operations with symbolic displacement operators and symbolic displacements of the matrices involved, which makes the presentation unified for various matrix structures. The study of the convergence rate requires estimates for the norms of the inverse displacement operators. We obtain such estimates by using three distinct techniques of some independent interest.

1.4. Related work

Newton's iteration for matrix inversion was proposed by Schultz in 1933 and was well studied (see [38] and bibliography therein). The Toeplitz-like case was studied in [2,4,27–30,33] and the Cauchy-like case in [40]. The unified approach was treated so far only in the two proceedings papers [36] (confined to the variant with the truncation of the singular values for both the iteration itself and the estimate of the inverse operator norms) and [37]. In Section 17, we will comment on the further ongoing research on our subjects.

1.5. Organization of our paper

We organize the rest of our paper as follows. In the following three sections, we state some basic definitions and assumptions and recall some auxiliary results for the unified study of structured matrices represented by their associated operators and displacements. In Section 5, we recall the definitions and basic facts for the matrix and operator norms. In Section 6, we brielfly recall Newton's iteration for general matrices and outline its modification where the input matrix is structured. The modification involves a subroutine for the compression of generators of the computed approximate inverses. We propose two variants of such a subroutine in Sections 7 and 9 (as Subalgorithms 7.1 and 9.1). In Sections 8 and 10, we analyze the resulting variants of our Newton-structured iteration and estimate its computational complexity and convergence rate. Our study in Sections 7–10 is elaborated for the displacement operator of Sylvester type. In Section 11, we extend the algorithms to the case of the operators of Stein type. In Sections 12–15, we estimate the norm

ν^- of the inverse displacement operators, required in our estimates for the output errors and the convergence rate of our algorithms. In Section 16, we breifly compare the unification approach with the transformation approach, which reduces to each other the inversion of structured matrix of various classes. In Section 17, we briefly comment on the extension of our algorithms to the cases where the displacement operator is singular and/or where no initial approximation to the inverse is available as well as to some other important computations with structured matrices.

2. Displacement operators and compressed displacement representation of structured matrices

The four classes of matrices of Table 1 are naturally associated with various linear *displacement operators L* of *Sylvester type* (also called *Lyapunov type* when they are applied over the functional spaces),

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

and/or *Stein type*,

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

Here A and B are fixed $n \times n$ matrices and are said to be *operator matrices*; the image matrices $L(M)$ are said to be the *displacements* of M. The operators of the two types are closely related to each other.

Theorem 2.1. $\nabla_{A,B} = A\Delta_{A^{-1},B}$ *if the operator matrix A is nonsingular*, *and* $\nabla_{A,B} = -\Delta_{A,B^{-1}}B$ *if the operator matrix B is nonsingular.*

Among the customary choices for A and B are the matrices of the scaling and shifts operators, $D(\mathbf{s})$, Z_f, Z_f^{T}, which we will define next.

Definition 2.1. For a fixed vector $\mathbf{s} = (s_i)_{i=0}^{k-1}$, $D(\mathbf{s}) = \mathrm{diag}(s_0, \ldots, s_{k-1})$ denotes a $k \times k$ diagonal matrix. $\mathbf{e}_j$ is the $(j+1)$th coordinate vector. $\mathbf{v}^{\mathrm{T}}$ and W^{T} are the transposes of a vector $\mathbf{v}$ and a matrix W, respectively. $\mathbf{v}^*$ and W^* are the Hermitian (conjugate) transposes. $Z = Z_0 = \sum_i \mathbf{e}_{i+1}\mathbf{e}_i^{\mathrm{T}}$ is the unit lower triangular Toeplitz matrix, defined by its first column $\mathbf{e}_1 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$.

$$J = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}$$

is the *reflection matrix*, and $\mathbf{t}^n = (t_i^n)$ for a vector $\mathbf{t} = (t_i)$. $Z_f = Z + f\mathbf{e}_0\mathbf{e}_{n-1}^{\mathrm{T}}$ denotes the unit *f*-circulant matrices (for a fixed scalar *f*).

We immediately observe that

Table 3
Operator matrices A, B of the operators $\Delta_{A,B}$ associated with the four basic classes of structured matrices

Matrix class	Matrix pair (A, B) for the operator $\Delta_{A,B}$	The $\Delta_{A,B}$-rank
Toeplitz	(Z_e, Z_f^{T}) or (Z_e^{T}, Z_f)	At most 2
Hankel	(Z_e, Z_f) or $(Z_e^{\mathrm{T}}, Z_f^{\mathrm{T}})$	At most 2
Vandermonde	$(D(\mathbf{t}), Z_e^{\mathrm{T}})$ or $(D(\mathbf{t}), Z_e)$	1
Cauchy	$(D^{-1}(\mathbf{s}), D(\mathbf{t}))$	1

$$Z_e^{\mathrm{T}} = JZ_eJ, \quad Z_{1/f}^{\mathrm{T}} = Z_f^{-1} \quad \text{for any scalars } e \text{ and } f \neq 0. \tag{2.3}$$

Definition 2.2. $Z(\mathbf{v})$ is the *lower triangular Toeplitz matrix* $\sum_{i=0}^{n-1} v_i Z^i$. $Z_f(\mathbf{v})$ is an *f-circulant matrix* $\sum_{i=0}^{n-1} v_i Z_f^i$; in particular $Z_1(\mathbf{v})$ is a circulant matrix.

To each linear operator $L = \nabla_{A,B}$ or $L = \Delta_{A,B}$, a class of structured matrices *M* is associated such that the rank ρ of the displacements $\nabla_{A,B}(M)$ and/or $\Delta_{A,B}(M)$ of the application of this operator to a matrix *M* is small relatively to the matrix size. Or equivalently,

$$L(M) = GH^{\mathrm{T}}, \tag{2.4}$$

G and *H* are $n \times \rho$ matrices, ρ is minimal for the three given matrices *M*, *A* and *B* and is small relatively to the matrix size.

We will call the matrix pair (G, H) by both *generator* for the displacement $L(M)$ and *L-generator* for the matrix *M*, and we will call the rank of $L(M)$ by the *L-rank* of *M* (for the operators *L* of Eqs. (2.1) and (2.2)) (see [4,18,31,32]). We will also say *displacement rank* and *displacement generator*, where *L* is unspecified or is known by default.

Table 3 represents some examples of operator matrices associated naturally with the matrices *M* of Table 1 and with the ones having similar structure. The matrices whose *L*-rank is small for the operator *L* of the respective lines of Table 3 will be called the matrices of *Toeplitz*, *Hankel*, *Vandermonde*, and *Cauchy* types, respectively. We will call them *Toeplitz-like*, *Hankel-like*, *Vandermonde-like*, and *Cauchy-like* matrices, respectively if their short *L*-generators are available. (Some authors use the nomenclature Vandermonde-like for a distinct class of matrices [13,23].) The above definitions cover several important classes of structured matrices such as Sylvester (resultant), Frobenius, Loewner, and Pick matrices [4,24]. Furthermore, the block submatrices, the products and the inverses of structured matrices inherit their structure [4,18,26,31,32], that is, have small *L*-rank for appropriate operators *L* too. There are also some natural extensions of the above matrix classes, such as polynomial Vandermonde-like matrices [21], Toeplitz + Hankel-like matrices [3,4,16] and block Toeplitz matrices.

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

In this paper, we will deal with nonsingular displacement operators L that are *readily and* (*bi*)*linearly invertible*, that is, we have simple expressions of matrices M via the generator matrices G and H and the operator matrices A and B, which are *linear* in the displacement $L(M)$ and *bilinear* in the entries of the generator matrices G and H. Here are some examples from [32,39], where such expressions are presented also for several other operators (see [1,4,9,10,16,18] on some earlier works and see [32,39] on the extensions to singular operators).

Theorem 2.2. *Let G and H be a pair of $n \times \alpha$ matrices*,

$$G = (\mathbf{g}_1, \ldots, \mathbf{g}_\alpha), \quad H = (\mathbf{h}_1, \ldots, \mathbf{h}_\alpha), \tag{2.5}$$

and let $L(M) = GH^{\mathrm{T}}$ (see (2.1), (2.2), and (2.4)). Then we have

(a)

$$(e - f)M = \sum_{j=1}^{\alpha} Z_e(\mathbf{g}_j) Z_f(J\mathbf{h}_j), \quad \textit{where } L = \nabla_{Z_e, Z_f}, \ e \neq f,$$

$$(1 - ef)M = \sum_{j=1}^{\alpha} Z_e(\mathbf{g}_j) Z_f^{\mathrm{T}}(\mathbf{h}_j), \quad \textit{where } L = \Delta_{Z_e, Z_f^{\mathrm{T}}}, \ ef \neq 1,$$

$$(e - f)M = \sum_{j=1}^{\alpha} Z_e(\mathbf{g}_j) Z_f(\mathbf{h}_j) J, \quad \textit{where } L = \nabla_{Z_e, Z_f^{\mathrm{T}}}, \ e \neq f,$$

$$(1 - ef)M = \sum_{j=1}^{\alpha} Z_e(\mathbf{g}_j) Z_f^{\mathrm{T}}(J\mathbf{h}_j) J, \quad \textit{where } L = \Delta_{Z_e, Z_f}, \ ef \neq 1.$$

(b)

$$(1 - fD^n(\mathbf{t}))M = \sum_{j=1}^{\alpha} D(\mathbf{g}_j) V(\mathbf{t}) Z_f^{\mathrm{T}}(\mathbf{h}_j),$$

where $L = \Delta_{D(\mathbf{t}), Z_f^{\mathrm{T}}}$, $t_i^n f \neq 1$ for all i,

$$(1 - fD^n(\mathbf{t}))M = \sum_{j=1}^{\alpha} D(\mathbf{g}_j) V(\mathbf{t}) J Z_f(J\mathbf{h}_j),$$

where $L = \Delta_{D(\mathbf{t}), Z_f}$, $t_i^n f \neq 1$ for all i,

$$(1 - fD^n(\mathbf{t}))M = D(\mathbf{t}) \sum_{j=1}^{\alpha} D(\mathbf{g}_j) V(\mathbf{t}) Z_f(J\mathbf{h}_j),$$

where $L = \nabla_{D^{-1}(\mathbf{t}), Z_f}$, $t_i^n f \neq 1$, $t_i \neq 0$ for all i,

$$(1 - fD^n(\mathbf{t}))M = D(\mathbf{t}) \sum_{j=1}^{\alpha} D(\mathbf{g}_j) V(\mathbf{t}) Z_f^{\mathrm{T}}(J\mathbf{h}_j),$$

where $L = \nabla_{D^{-1}(\mathbf{t}),Z_f^{\mathrm{T}}}$, $t_i^n f \neq 1$, $t_i \neq 0$ *for all i.*

(c) *For operators L of Cauchy type, we have*

$$M = \sum_{j=1}^{\alpha} D(\mathbf{g}_j)C(\mathbf{s},\mathbf{t})D(\mathbf{h}_j),$$

where $L = \nabla_{D(\mathbf{s},D(\mathbf{t}))}$, $s_i \neq t_k$ *for all* i, k,

$$M = \sum_{j=1}^{\alpha} D(\mathbf{g}_j)\left(\frac{1}{1-s_it_k}\right)_{i,k} D(\mathbf{h}_j),$$

where $L = \Delta_{D(\mathbf{s}),D(\mathbf{t})}$, $s_i \neq t_k$ *for all* i, k.

Remark 2.1. The above expressions can be immediately extended to some other operators L based on the equations

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

Theorem 2.2 enables immediate extension of the computational cost bounds of Table 2 to more general classes of structured matrices.

Definition 2.3. $v_\alpha = v_{\alpha,n}(L)$ denotes the arithmetic cost (in flops) of multiplication by a vector of an $n \times n$ matrix M represented by its L-generator of length α. $m_{\alpha,n} = m_{\alpha,n}(L, L_1)$ denotes the arithmetic cost of multiplication of a pair of $n \times n$ matrices, where the input matrices are represented by their L- and L_1-generators of length $\mathrm{O}(\alpha)$ for nonsingular operators L and L_1, respectively.

Theorem 2.3. *We have* $v_{\alpha,n}(L) = \mathrm{O}(\alpha n \log n)$ *for* $L = \nabla_{A,B}$, $L = \Delta_{A,B}$ *for any pair of matrices* A, B *from the set* $\{Z_e, Z_e^{\mathrm{T}}, Z_f, Z_f^{\mathrm{T}}\}$ *and for any pair of scalars e and f*; $v_{\alpha,n}(L) = \mathrm{O}(\alpha n \log^2 n)$ *for* $L = \nabla_{A,B}$, $L = \Delta_{A,B}$, *where* $A = D(\mathbf{s})$, $B = D(\mathbf{t})$, *or* $A = D(\mathbf{s})$, $B \in \{Z_f, Z_f^{\mathrm{T}}\}$, *or* $A \in \{Z_f, Z_f^{\mathrm{T}}\}$, $B = D(\mathbf{s})$ *for any pair of vectors* $\mathbf{s}$ *and* $\mathbf{t}$ *and any scalar f.*

The displacement rank approach can be represented by the following flowchart:

COMPRESS, OPERATE, RECOVER

To take advantage of the matrix structure, we will COMPRESS the structured input matrices M via their short L-generators based on Theorem 2.2 or its generalization, then OPERATE with L-generators rather than with the matrices themselves, and finally RECOVER the output from the computed short L-generators.

3. Basic matrix operation performed with displacements

The following three theorems express the operator and generator matrices for the linear combinations, products and inverses of matrices. They are well known [18,31] and can be easily verified.

Theorem 3.1. *For any linear operator L, any pair of $m \times n$ matrices M and N, and any pair of scalars a and b, we have $L(aM + bN) = aL(M) + bL(N)$.*

Theorem 3.2. *For any 5-tuple (A, B, C, M, N) of matrices of compatible sizes we have*

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

Furthermore,

$$\Delta_{A,C}(MN) = \Delta_{A,B}(M)N + AMB\Delta_{B^{-1},C}(N)$$

if B is a nonsingular matrix, whereas

$$\Delta_{A,C}(MN) = \Delta_{A,B}(M)N - AM\Delta_{B,C^{-1}}(N)C$$

if C is a nonsingular matrix.

Theorem 3.3. *Let a pair of $n \times \alpha$ matrices G and H form a $\Delta_{A,B}$-generator of length α for a nonsingular matrix M. Write $M^{-1}G = -U$ and $H^{\mathrm{T}}M^{-1} = W^{\mathrm{T}}$. Then $\nabla_{B,A}(M^{-1}) = UW^{\mathrm{T}}$. Furthermore,*

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

if B is a nonsingular matrix, whereas

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

if A is a nonsingular matrix.

Theorem 3.2 motivates the following definition [32].

Definition 3.1. Operator pairs $(\nabla_{A,B}, \nabla_{B,C})$, $(\Delta_{A,B}, \nabla_{B,C})$, $(\Delta_{A,B}, \Delta_{B^{-1},C})$, and $(\Delta_{A,B}, \Delta_{B,C^{-1}})$ are called *compatible*.

Theorem 3.4. *For a pair of compatible operators L and L_1 associated with operator matrix pairs of Theorem 2.3, we have $m_{\alpha,n}(L, L_1) = \mathrm{O}(\alpha v_{\alpha,n}(L) + \alpha v_{\alpha,n}(L_1))$.*

Our next assumption is motivated by Theorems 2.3 and 3.4.

Assumption 3.1. Hereafter, we will always deal with nonsingular operators L and L_1 having linear inverse operators and such that

$$\begin{aligned} &m_{\alpha,n}(L, L_1) = \mathrm{O}(\alpha v_{\alpha,n}(L) + \alpha v_{\alpha,n}(L_1)), \\ &v_{\alpha,n}(L) = \mathrm{O}(\alpha n \log^d n), \quad d \leqslant 2. \end{aligned}$$

We will call such operators *strongly regular*.

4. Orthogonal displacement representation of structured matrices

For a fixed pair of operator L and matrix M, we may choose the *orthogonal* (*SVD-based*) *L-generator matrices* to achieve better numerical stability [2,27,29,33]). That is, we first compute the SVD of the displacement $W = L(M)$,

$$W = U\Sigma^2 V^{\mathrm{T}}, \tag{4.1}$$

$$\begin{aligned} &U^*U = V^*V = I_\rho, \quad \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_\rho), \\ &\sigma_1 \geqslant \cdots \geqslant \sigma_\rho > 0, \quad \rho = \mathrm{rank}(W), \end{aligned} \tag{4.2}$$

where U and V are $m \times \rho$ and $\rho \times n$ matrices, respectively, and $\sigma_1^2, \ldots, \sigma_\rho^2$ denote the singular values of the matrix W, and then we write

$$G = U\Sigma, \quad H = V\Sigma. \tag{4.3}$$

Remark 4.1 (*see* [29]). The SVD computation is quite inexpensive in this case, involving $\mathrm{O}(n\alpha^2 + \alpha(\log\log(1/\delta)\log\alpha)$ flops for an $n \times n$ matrix M given with its L-generators of length α and for δ denoting the output approximation error bound for the SVD; we will ignore the latter term assuming realistically that $(\log\log(1/\delta))\log\alpha = \mathrm{O}(n\alpha)$. The computation of the SVD of the displacement $L(M)$ given with its longer L-generator of length α enables us to achieve maximal compression of a matrix M, that is, to obtain its shortest L-generator. An alternative algorithm of Proposition A.6 of [28] for the same compression task uses $\mathrm{O}(n\alpha^2)$ flops and involves no SVDs. Thus, we will usually assume that our structured matrices are given with their shortest L-generators.

5. Matrix and operator norms

We will need some further definitions in addition to the ones of Section 1.

Definition 5.1. $\|M\|$ denotes any fixed operator norm of a matrix M. $\|M\|_l$ is the l-norm, $l = 1, 2, \infty$ (see [4,7]). $\kappa(M) = \mathrm{cond}_2(M) = \sigma_1^2(M)/\sigma_\rho^2(M)$, where $\sigma_i^2(M)$ is the ith singular value of M (see (4.1) and (4.2)), $i = 1, \ldots, \rho$, $\rho = \mathrm{rank}(M)$.

Theorem 5.1 (see [4,7]). *$\|M\|_2 = \sigma_1^2(M)$ for every matrix M, and $\kappa(M) = \|M\|_2 \|M^{-1}\|_2$ for an $n \times n$ nonsingular matrix $M = [m_{i,j}]$. Furthermore, we have $\|M\|_l/\sqrt{n} \leqslant \|M\|_2 \leqslant |M\|_l\sqrt{n}$, $l = 1, \infty$; $\|M\|_1 = \|M^{\mathrm{T}}\|_\infty = \max_j \sum_i |m_{i,j}|$, $\|M\|_2^2 \leqslant \|M\|_1\|M\|_\infty$.*

Definition 5.2. We define the *norms* of a nonsingular linear operator L and its inverse L^{-1}:

$$\nu = \nu_{\rho,l}(L) = \sup_M(\|L(M)\|_l/\|M\|_l),$$
$$\nu^- = \nu^-_{\rho,l}(L) = \nu_{\rho,l}(L^{-1}) = \sup_M(\|M\|_l/\|L(M)\|_l),$$

where $l = 1, 2, \infty$ and the supremum is over all matrices M having positive L-rank of at most ρ. We also define the *condition number* of the operator L:

$$\kappa = \kappa(L) = \mathrm{cond}(L) = \nu\nu^- = \nu_{\rho,l}(L)\nu^-_{\rho,l}(L).$$

6. The Newton-structured iteration

Let us assume that a crude initial approximation to $\nabla_{A,B}(M^{-1})$ is available, supplied, say by the preconditioned conjugate gradient method, which converges to the output rather slowly, with linear rate [5], or by a direct solution algorithm performed with rounding. The approximations can be rapidly refined by means of Newton's iteration for matrix inversion:

$$X_{i+1} = 2X_i - X_i M X_i, \quad i = 0, 1, \ldots \tag{6.1}$$

Matrix equation (5.1) implies that

$$I - MX_{i+1} = (I - MX_i)^2, \quad \|I - MX_{i+1}\| \leqslant \|I - MX_i\|^2$$

for all i. That is, we have quadratic convergence if $\|I - MX_0\| < 1$. This is a special case of the residual correction process [17, p. 86]. The iteration is numerically stable even where M is a singular matrix (in this case the iteration converges to the Moore–Penrose generalized inverse M^+) and can be accelerated based on various policies of scaling X_{i+1} for all i and choosing an initial approximation X_0 [38]. We will only study unscaled Newton's iteration (see our comments on scaling in Section 17). Furthermore, to make our analysis more transparent, we will work with iteration (6.1) though practically it is slightly simpler to perform the computations with the matrices $-X_i$ and $-X_{i+1}$ and to rely on the equations $-X_{i+1} = (-X_i)(2I + M(-X_i))$, $i = 0, 1, \ldots$ [33].

Each step (6.1) is essentially two matrix multiplications, which use $\mathrm{O}(\alpha^2 n \log^d n)$ flops, $d \leqslant 2$, for structured matrices of Theorems 2.3 and 3.4. In particular, for structured matrices, M and X_0, having short $\nabla_{A,B}$- and $\nabla_{B,A}$-generators, respectively, the iteration can be performed efficiently by operating with short ∇-generators of the matrices M, X_i, and MX_i (or X_iM). This, however, requires some special techniques for controlling the length of the $\nabla_{B,A}$-generators of X_i, which tends to be tripled at every iterative step. Similar comments apply where $\Delta_{A,B}$- and $\Delta_{B,A}$-generators are used.

Two methods proposed in [27,30,33,40] counter such a mishap in the case of Toeplitz-like and Cauchy-like matrices. Our main goal in the present paper is to extend them to various other classes of structured matrices in a unified way and to analyze the resulting algorithms.

Here is the basic observation of [27,30,33,40]. By assumption, rank$(\nabla_{B,A}(M^{-1}))$ $= \rho$. Therefore, the matrices X_i, which approximate M^{-1} closely for larger i, have a nearby matrix M^{-1} of $\nabla_{B,A}$-rank ρ. Thus, our strategy is to replace X_i in (6.1) by a nearby matrix Y_i having $\nabla_{B,A}$-rank at most ρ and then to restart the iteration with Y_i instead of X_i.

Let us next formally describe this approach for Sylvester type operators. (On the extension to Stein type operators, see Section 11.)

Algorithm 6.1 (*The Newton-structured matrix iteration for the Sylvester type operators*).

Input. A positive integer ρ, a pair of $n \times n$ matrices A and B, an $n \times n$ nonsingular matrix M having $\nabla_{A,B}$-rank ρ and defined by its $\nabla_{A,B}$-generator (G, H) of length ρ, a matrix Y_0 (an initial approximation to the matrix M^{-1}) given with its $\nabla_{B,A}$-generator of length at most ρ, a bound on the number N of Newton's iteration steps, and a subroutine **R** for the transition from a $\nabla_{B,A}$-generator of length at most 3ρ for an $n \times n$ matrix approximating M^{-1} to an $\nabla_{A,B}$-generator of length at most ρ for a nearby matrix.

Output. A $\nabla_{B,A}$-generator of length at most ρ for a matrix Y_{l+1} approximating M^{-1}.

Computations. Recursively compute $\nabla_{B,A}$-generators of length at most 3ρ for the matrices

$$X_{i+1} = Y_i(2I - MY_i), \quad i = 0, 1, \ldots, N-1, \tag{6.2}$$

and $\nabla_{B,A}$-generators of length at most ρ for the matrices Y_{i+1} defined by a transformation from X_{i+1} by means of the subroutine **R**.

Theorem 6.1 (see [40] or [32]). *Let the assumptions of Algorithm* 6.1 *hold. Then for any* $i = 0, 1, \ldots,$ *a* $\nabla_{B,A}$*-generator of length at most* 3ρ *for the matrix* $X_{i+1} = 2Y_i - Y_iMY_i$ *can be computed at the cost of performing* $\mathrm{O}(\rho v_{\rho,n}(\nabla_{B,A}) + \rho v_{\rho,n}(\nabla_{A,B}))$ *flops*, *which is* $\mathrm{O}(\rho^2 n \log^d n)$ *flops for* $d \leqslant 2$ *under Assumption* 3.1.

To complete the description of the Newton-structured iteration, it remains to specify the subroutine **R**, which controls the length of the computed L-generators. We will do this in two ways, to be specified in Sections 7 and 9.

7. Newton-structured iteration I: compression by the truncation of singular values

The following result enables us to compress a matrix Y_i lying near X_i and M^{-1}.

Theorem 7.1 [7, pp. 72, 230]. *Given a matrix W of rank ρ and a non-negative integer β, $\beta \leqslant \rho$, it holds that*

$$\sigma_{\beta+1}^2 = \min_{B:\mathrm{rank}(B)\leqslant\beta} \|W - B\|_2.$$

We will represent the displacement $\nabla_{B,A}(X_i)$ via its SVD, truncate all its singular values except for the ρ largest of them, and thus obtain a $\nabla_{B,A}$-generator of length at most ρ for a nearby matrix Y_i. The matrices X_i and M lie near each other. Furthermore, we have $\|\nabla_{B,A}(X_i) - \nabla_{B,A}(Y_i)\|_2 \leqslant \|\nabla_{B,A}(X_i) - \nabla_{B,A}(M^{-1})\|_2$ by Theorem 7.1 because $\mathrm{rank}(\nabla_{B,A}(M^{-1})) \leqslant \rho$. For invertible operators $\nabla_{B,A}$, this implies that also Y_i lies near X_i.

To specify and to analyze formally the transition from the matrices X_i to Y_i, we will use some further definitions and simple preliminary results.

Hereafter, we will write $\beta = \beta_i = \mathrm{rank}(\nabla_{B,A}(X_i))$. ($\beta \leqslant 3\rho$ for all i, by Theorem 6.1.) Let us also write

$$e_{l,i} = \|X_i - M^{-1}\|_l, \quad l = 1, 2, \infty; \quad e_i = \|X_i - M^{-1}\|, \tag{7.1}$$

$$\hat{e}_{l,i} = \|Y_i - M^{-1}\|_l, \quad l = 1, 2, \infty; \quad \hat{e}_i = \|Y_i - M^{-1}\|, \tag{7.2}$$

$$r_i = \|I - Y_i M\|_2. \tag{7.3}$$

Now, we are ready to describe variant I of subroutine **R** for Algorithm 6.1.

Subalgorithm 7.1 (*Compression of a displacement by truncation of its smallest singular values*).

Input. A positive integer ρ, operator matrices A and B, a $\nabla_{A,B}$-generator of length ρ for a nonsingular $n \times n$ matrix M, where $\rho = \mathrm{rank}(\nabla_{A,B}(M)) = \mathrm{rank}(\nabla_{B,A}(M^{-1}))$, and a $\nabla_{B,A}$-generator (G_i, H_i) of length at most $\beta = \beta_i$ for a matrix X_i such that $\rho \leqslant \beta$, $\nabla_{B,A}(X_i) = G_i H_i^{\mathrm{T}}$.

Output. A $\nabla_{B,A}$-generator of length at most ρ for a matrix Y_i such that

$$\|Y_i - M^{-1}\|_2 \leqslant (1 + (\|A\|_2 + \|B\|_2)\nu^-)e_{2,i} \tag{7.4}$$

for $e_{2,i}$ of (7.1) and $\nu^- = \nu_{\rho,2}(\nabla_{A,B}^{-1})$ of Definition 5.2.

Computations.

(a) Compute the SVD of the displacement $\nabla_{B,A}(X_i) = U_i \Sigma_i^2 V_i^{\mathrm{T}}$.

(b) Set to zero the diagonal entries $\sigma_{\rho+1}^2, \ldots, \sigma_\beta^2$ of the matrix Σ_i^2, thus turning Σ_i^2 into a diagonal matrix of rank at most ρ. ($\sigma_{\rho+1}^2, \ldots, \sigma_\beta^2$ are the $\beta - \rho$ smallest singular values of the matrix $\nabla_{B,A}(X_i)$.)

(c) Compute and output the matrices G_i^* and H_i^* obtained from the matrices $U_i\Sigma_i$ and $V_i\Sigma_i$, respectively, by deleting their last $\beta - \rho$ columns.

Correctness of Subalgorithm 7.1 is implied by the following result, which shows that bound (7.4) holds under our assumptions on the input of Algorithm 6.1 and Subalgorithm 7.1.

Theorem 7.2. *Let the structured matrices M^{-1}, X_i, and Y_i be defined as above and let a positive scalar $e_{2,i}$ be defined by Eq.* (7.1). *Let $\nabla_{B,A}$ be a nonsingular linear operator. Then bound* (7.4) *holds.*

Theorem 7.2 generalizes a result proved in [27,29,30] for the Toeplitz-like case. To prove Theorem 7.2, we need the following two lemmas.

Lemma 7.1. *Under the notation of Algorithm* 6.1, *we have*

$$\|\nabla_{B,A}(X_i) - \nabla_{B,A}(Y_i)\|_2 = \sigma_{\rho+1}^2(\nabla_{B,A}(X_i)), \tag{7.5}$$

$$\|\nabla_{B,A}(M^{-1}) - \nabla_{B,A}(X_i)\| \leqslant (\|A\| + \|B\|)e_i, \tag{7.6}$$

for e_i of (7.1).

Proof. Eq. (7.5) follows immediately from Theorem 5.1. To prove bound (7.6), recall that

$$\nabla_{B,A}(M^{-1}) = AM^{-1} - M^{-1}B,$$
$$\nabla_{B,A}(X_i) = AX_i - X_iB.$$

Therefore,

$$\begin{aligned}
&\|\nabla_{B,A}(M^{-1}) - \nabla_{B,A}(X_i)\| \\
&\quad = \|X_iB - AX_i - M^{-1}B + AM^{-1}\| \\
&\quad = \|(X_i - M^{-1})B - A(X_i - M^{-1})\| \\
&\quad \leqslant \|X_i - M^{-1}\| \cdot \|B\| + \|A\| \cdot \|X_i - M^{-1}\| \\
&\quad \leqslant (\|A\| + \|B\|)\|X_i - M^{-1}\| \\
&\quad = (\|A\| + \|B\|)e_i. \qquad \square
\end{aligned}$$

Lemma 7.2. $\|\nabla_{B,A}(X_i) - \nabla_{B,A}(Y_i)\|_2 \leqslant (\|A\|_2 + \|B\|_2)e_{2,i}$.

Proof. Apply the well-known estimate of [7, p. 428] and deduce that

$$|\sigma_j^2(\nabla_{B,A}(X_i)) - \sigma_j^2(\nabla_{B,A}(M^{-1}))| \leqslant \|\nabla_{B,A}(X_i) - \nabla_{B,A}(M^{-1})\|_2$$

for all j, where $\sigma_j^2(W)$ are defined by (4.1) and (4.2). For all $j > \rho$, recall that $\sigma_j^2(\nabla_{B,A}(M^{-1})) = 0$ and obtain

$$\sigma_j^2(\nabla_{B,A}(X_i)) \leqslant \|\nabla_{B,A}(X_i) - \nabla_{B,A}(M^{-1})\|_2.$$

Now, substitute inequality (7.6) and deduce that

$$\sigma_j^2(\nabla_{B,A}(X_i)) \leqslant (\|A\|_2 + \|B\|_2)e_{2,i} \quad \text{for } j > \rho.$$

Combine this bound for $j = \rho + 1$ with Eq. (7.5) and deduce Lemma 7.2. □

Now, we are prepared to prove Theorem 7.2.

Proof of Theorem 7.2. By first applying Definition 5.2 for $l = 2$ and $L = \nabla_{B,A}$ and then applying the linearity of the operator $\nabla_{B,A}$, we obtain that

$$\|X_i - Y_i\|_2 \leqslant \nu^- \|\nabla_{B,A}(X_i - Y_i)\|_2 = \nu^- \|\nabla_{B,A}(X_i) - \nabla_{B,A}(Y_i)\|_2.$$

On the right-hand side of the inequality

$$\|M^{-1} - Y_i\|_2 \leqslant \|M^{-1} - X_i\|_2 + \|X_i - Y_i\|_2,$$

substitute Eq. (7.1) for $l = 2$, that is, $e_{2,i} = \|X_i - M^{-1}\|_2$, substitute the above bound on $\|X_i - Y_i\|_2$ and the one of Lemma 7.2, and obtain that

$$\begin{aligned}\|M^{-1} - Y_i\|_2 &\leqslant e_{2,i} + \|\nabla_{B,A}(X_i) - \nabla_{B,A}(Y_i)\|_2 \nu^- \\ &\leqslant e_{2,i} + (\|A\|_2 + \|B\|_2)e_{2,i}\nu^-. \qquad \square\end{aligned}$$

8. Newton-structured iteration I: convergence rate and computational complexity estimates

Combining Algorithm 6.1 with Subalgorithm 7.1 applied as a subroutine **R** defines *Newton-structured iteration* I. Next, we will estimate its convergence rate and computational complexity. Estimating the computational cost, we will rely on Definition 5.2 and the bound $\mathrm{O}(n\rho^2)$ of Remark 4.1 on the cost of computing the SVD. This immediately implies:

Theorem 8.1. *Newton-structured iteration* I *produces the matrices* $X_1, Y_1, X_2, Y_2, \ldots, X_i, Y_i$ *by performing* $\mathrm{O}((v_{\rho,n}(\nabla_{B,A}) + v_{\rho,n}(\nabla_{A,B}) + \rho n)i\rho)$ *flops, which is* $\mathrm{O}(i\rho^2 n \log^d n)$ *flops for* $d \leqslant 2$ *under Assumption* 3.1.

Let us next estimate the convergence rate of the iteration. We have

$$I - X_{i+1}M = (I - Y_iM)^2, \quad \|I - X_{i+1}M\|_2 \leqslant r_i^2$$

for $r_i = \|I - Y_iM\|_2$ of (7.3); therefore, $\|M^{-1} - X_{i+1}\|_2 \leqslant r_i^2\|M^{-1}\|_2$. By Theorem 7.2, we have

$$\|M^{-1} - Y_{i+1}\|_2 \leqslant (1 + (\|A\|_2 + \|B\|_2)\nu^-)\|M^{-1} - X_{i+1}\|_2.$$

Consequently, we have

$$\|M^{-1} - Y_{i+1}\|_2 \leqslant (1 + (\|A\|_2 + \|B\|_2)\nu^-)r_i^2\|M^{-1}\|_2.$$

Therefore,

$$\begin{aligned} r_{i+1} = \|I - Y_{i+1}M\|_2 &\leqslant \|M^{-1} - Y_{i+1}\|_2\|M\|_2 \\ &\leqslant (1 + (\|A\|_2 + \|B\|_2)\nu^-)r_i^2\|M^{-1}\|_2\|M\|_2 \\ &\leqslant (1 + (\|A\|_2 + \|B\|_2)\nu^-)r_i^2\kappa(M), \end{aligned}$$

where $\kappa(M) = \mathrm{cond}_2(M) = \|M^{-1}\|_2\|M\|_2$ (see Theorem 5.1). Let us rewrite the latter bound as follows:

$$\mu r_{i+1} \leqslant (\mu r_i)^2, \quad \mu = (1 + (\|A\|_2 + \|B\|_2)\nu^-)\kappa(M) \text{ for } i = 0, 1, \ldots \tag{8.1}$$

Relations (8.1) imply that

$$\mu r_i \leqslant (\mu r_0)^{2^i}, \quad i = 0, 1, \ldots$$

The following theorem summarizes our analysis.

Theorem 8.2. *Let the matrices X_0 and M be given with their $\nabla_{B,A}$- and $\nabla_{A,B}$-generators of length β_0 and ρ, respectively. Furthermore, let*

$$\mu r_0 \leqslant \theta < 1, \quad \mu = (1 + (\|A\|_2 + \|B\|_2)\nu^-)\kappa(M), \tag{8.2}$$

for $r_0 = \|I - Y_0M\|_2$ of (7.3), *$\nu^- = \nu^{-1}_{\rho,2}(\nabla_{A,B})$ of Definition* 5.2, *μ of* (8.1), *and some fixed real θ. Then for all positive i, we have $\hat{e}_{2,i} = \|Y_i - M^{-1}\|_2 \leqslant r_i\|M^{-1}\|_2 \leqslant (\mu r_0)^{2^i}\|M^{-1}\|_2/\mu \leqslant \theta^{2^i}\|M^{-1}\|_2/\mu$.*

Corollary 8.1 (see also Remark 4.1). *Under the assumptions of Theorem* 8.2, *the residual norm bound $r_l = \|I - Y_lM\|_2 \leqslant \epsilon/\mu$ is ensured in*

$$l = \lceil \log_2(\log \epsilon / \log \theta) \rceil$$

steps of Newton-structured iteration I. *These steps require* $\mathrm{O}((v_{\rho,n}(\nabla_{B,A}) + v_{\rho,n}(\nabla_{A,B}))\rho l)$ *flops, which is* $\mathrm{O}(l\rho^2 n \log^d n)$ *for $d \leqslant 2$ under Assumption* 3.1.

9. Newton-structured iteration II: compression by means of substitution

Let us describe an SVD-free method for the compression of approximate inverses. First recall that $\nabla_{B,A}(M^{-1}) = -M^{-1}GH^{\mathrm{T}}M^{-1}$, by Theorem 3.3. Now substitute X_i for M^{-1} on the right-hand side and define a short $\nabla_{B,A}$-generator for the matrix Y_i:

$$\nabla_{B,A}(Y_i) = U_i W_i^{\mathrm{T}}, \quad U_i = -X_i G \in \mathbb{C}^{n\times\rho}, \quad W_i^{\mathrm{T}} = H^{\mathrm{T}} X_i \in \mathbb{C}^{\rho\times n}. \tag{9.1}$$

(We expect that $Y_i \approx M^{-1}$ because $\nabla_{B,A}(Y_i) \approx \nabla_{B,A}(M^{-1})$, which should hold because $X_i \approx M^{-1}$.) This leads us to the following variant of subroutine **R**.

Subalgorithm 9.1 (*Compression of the displacement by substitution of an approximate inverse for the inverse*).

Input. A positive integer ρ, a pair of $n \times n$ operator matrices A and B defining a strongly regular operator $\nabla_{B,A}$, a $\nabla_{A,B}$-generator of length ρ for a nonsingular $n \times n$ matrix M where $\rho = \operatorname{rank}(\nabla_{A,B}(M)) = \operatorname{rank}(\nabla_{B,A}(M^{-1}))$, and a $\nabla_{B,A}$-generator (G_{i+1}, H_{i+1}) of length at most 3ρ for a matrix X_{i+1} of Eq. (6.1).

Output. A $\nabla_{B,A}$-generator (U_{i+1}, W_{i+1}) of length at most ρ for a matrix Y_{i+1} such that

$$\hat{e}_{i+1} = \|Y_{i+1} - M^{-1}\| \leqslant C_i e_i \tag{9.2}$$

for $\hat{e}_{i+1}$ of (7.2), e_i of (7.1), $C_i = \nu^{-}\|GH^{\mathrm{T}}\|(e_i + 2\|M^{-1}\|)$, and $\nu^{-} = \nu_\rho(\nabla_{B,A}^{-1})$ of Definition 5.2.

Computations. Compute and output the matrix products $U_{i+1} = -X_{i+1}G$, $W_{i+1}^{\mathrm{T}} = H^{\mathrm{T}}X_{i+1}$.

Under Assumption 3.1 about strong regularity of the operator $\nabla_{B,A}$, the matrix pair (U_{i+1}, W_{i+1}) is a $\nabla_{B,A}$-generator of length at most ρ for a matrix Y_{i+1}, which is a unique solution to the following equation (see (9.1)):

$$\nabla_{B,A}(Y_{i+1}) = U_{i+1} W_{i+1}^{\mathrm{T}}.$$

The computation of the $n \times \rho$ matrices U_{i+1}, W_{i+1} of (9.1) is reduced to multiplication of the matrix X_{i+1} by the $n \times (2\rho)$ matrix $(-G, H)$. This requires $\mathrm{O}((\nu_{\rho,n}(\nabla_{B,A}) + \nu_{\rho,n}(\nabla_{A,B})\rho))$ flops, which is $\mathrm{O}(\rho^2 n \log^d n)$ flops under Assumption 3.1.

To prove correctness of the subalgorithm, that is, to prove bound (9.2), we need some auxiliary results. Recall the matrix equations $-U = M^{-1}G$ and $W^{\mathrm{T}} = H^{\mathrm{T}}M^{-1}$ of Theorem 3.3 and deduce that

$$\begin{aligned} -U_j &= X_j G = (X_j - M^{-1})G + M^{-1}G, \\ W_j^{\mathrm{T}} &= H^{\mathrm{T}} X_j = H^{\mathrm{T}}(X_j - M^{-1}) + H^{\mathrm{T}}M^{-1}. \end{aligned}$$

Now, write $E_j = UW^{\mathrm{T}} - U_j W_j^{\mathrm{T}}$ and obtain the following matrix equation:

$$\begin{aligned} E_j = {} & (X_j - M^{-1})GH^{\mathrm{T}}(X_j - M^{-1}) \\ & + M^{-1}GH^{\mathrm{T}}(X_j - M^{-1}) \\ & + (X_j - M^{-1})GH^{\mathrm{T}}M^{-1}. \end{aligned}$$

Lemma 9.1. *For the matrices U_j, W_j^{T}, and E_j defined above and for $e_j = \|X_j - M^{-1}\|$ of* (7.1), *we have*

$$\|E_j\| = \|U_j W_j^{\mathrm{T}} - UW^{\mathrm{T}}\| \leqslant \|GH^{\mathrm{T}}\| e_j(e_j + 2\|M^{-1}\|).$$

Proof. The lemma follows from the above expression for E_j. □

Theorem 9.1. *For $i = 0, 1, \ldots,$ we have $\hat{e}_{i+1} \leqslant C_{i+1}e_{i+1}$ for e_i of* (7.1), *$\hat{e}_i$ of* (7.2), *$C_i = \nu^- \|GH^{\mathrm{T}}\|(e_i + 2\|M^{-1}\|)$, and the norm $\nu^- = \nu(\nabla_{B,A}^{-1})$ of Definition* 5.2.

Proof. Recall that $\hat{e}_{i+1} = \|Y_{i+1} - M^{-1}\| \leqslant \nu^- \|\nabla_{B,A}(Y_{i+1} - M^{-1})\|$. Since the operator $\nabla_{B,A}$ is linear, we have $\hat{e}_{i+1} \leqslant \nu^- \|\nabla_{B,A}(Y_{i+1}) - \nabla_{B,A}(M^{-1})\| \leqslant \nu^- \| U_{i+1}W_{i+1}^{\mathrm{T}} - UW^{\mathrm{T}}\| \leqslant \nu^- \|E_{i+1}\|$. At this point, apply Lemma 9.1 for $j = i + 1$ and obtain that $\hat{e}_{i+1} \leqslant C_{i+1}e_{i+1}$. □

10. Newton-structured iteration II: its convergence rate and computational complexity estimates

Combining Algorithm 6.1 with Subalgorithm 9.1 (applied as subroutine **R**) defines Newton-structured iteration II. Our next goal is to estimate its convergence rate and its computational complexity.

Lemma 10.1. *For a nonsingular matrix M, the matrix X_{i+1} defined by Eq.* (6.1), *the matrix Y_{i+1} of Subalgorithm* 9.1, *and the scalars C_i, $e_{i+1} = \|X_{i+1} - M^{-1}\|$ and $\hat{e}_i = \|Y_i - M^{-1}\|$ (of Theorem* 9.1 *and Eqs.* (7.1) *and* (7.2)), *we have $e_{i+1} \leqslant \|M\|\hat{e}_i^2 \leqslant (C_i e_i)^2\|M\|$ for $i = 0, 1, \ldots$*

Proof. By (6.1), we have $I - MX_{i+1} = (I - MY_i)^2$, $i = 0, 1, \ldots$ It follows that $e_{i+1} = \|X_{i+1} - M^{-1}\| = \|M^{-1}(I - MX_{i+1})\| = \|M^{-1}(I - MY_i)^2\| = \|(M^{-1} - Y_i)M(M^{-1} - Y_i)\| \leqslant \|M\|\hat{e}_i^2$. Finally, substitute the bound of Theorem 9.1. □

Let us restate this lemma in a more constructive way, that is, let us replace the values e_0, $\|M^{-1}\|$, and C_i by more readily available values. Write

$$e_0^* = r_0\|Y_0\|/(1 - r_0) \tag{10.1}$$

for $r_0 = \|I - MY_0\|$ of (7.3) and assume realistically that

$$r_0 \leqslant 1, \tag{10.2}$$

$$e_i \leqslant \|M^{-1}\|, \tag{10.3}$$

for e_i of (7.1) and for all i.

Lemma 10.2. *Assuming relations* (10.1)–(10.3), *we have*

$$\|M^{-1}\| \leqslant \|Y_0\|/(1-r_0), \tag{10.4}$$

$$\hat{e}_0 \leqslant e_0^*, \tag{10.5}$$

$$C_i \leqslant C = 3\nu^- \|GH^{\mathrm{T}}\| \cdot \|Y_0\|/(1-r_0) \quad \textit{for all } i. \tag{10.6}$$

Proof. We have

$$|\|M^{-1}\| - \|Y_0\|| \leqslant \hat{e}_0 = \|M^{-1} - Y_0\| \leqslant \|M^{-1}\| r_0,$$

and (10.4) follows. Substitute (10.4) into the bound $\hat{e}_0 \leqslant \|M^{-1}\| r_0$ and obtain (10.5). Substitute (10.3) into the expression of Theorem 9.1 for C_i, then substitute (10.4), and obtain (10.6). □

By combining Theorem 9.1, Lemmas 10.1 and 10.2, we obtain:

Theorem 10.1. *Assume relations* (10.1)–(10.3). *Then* $\hat{e}_i \leqslant Ce_i$, $e_{i+1} \leqslant (Ce_i)^2\|M\|$ *for* C *of* (10.6), e_i *of* (7.1), $\hat{e}_i$ *of* (7.2), *and* $i = 1, 2, \ldots$

Corollary 10.1. *Assume relations* (10.1)–(10.3) *and write* $\bar{\mu} = C^2\|M\|$. *Then we have*

$$\bar{\mu} e_{i+1} \leqslant (\bar{\mu} e_i)^2 \leqslant (\bar{\mu} e_1)^{2^i} \quad \textit{for } i = 1, 2, \ldots \tag{10.7}$$

By applying Lemma 10.1 and then bound (10.5), we obtain that $e_1 \leqslant \hat{e}_0^2\|M\| \leqslant (e_0^*)^2\|M\|$. Substitute the latter bound into Corollary 10.1 and obtain:

Corollary 10.2. *Assume relations* (10.1)–(10.3) *and the bound*

$$(\bar{\mu} e_1)^{1/2} \leqslant Ce_0^*\|M\| \leqslant \theta < 1, \tag{10.8}$$

for $\bar{\mu} = C^2\|M\|$, e_0^* *of* (10.1), C *of* (10.6), *and a real* θ. *Then we have* $\bar{\mu} e_{i+1} \leqslant (\bar{\mu} e_1)^{2^i} \leqslant (Ce_0^*\|M\|)^{2^{i+1}} \leqslant \theta^{2^{i+1}}$ *and* $\hat{e}_{i+1} \leqslant Ce_{i+1}$, $i = 0, 1, \ldots$

Corollary 10.3. *Write*

$$i^* + 1 = \lceil \log_2((\log \epsilon^*)/\log \theta) \rceil,$$

and assume relations (10.1)–(10.3) *and* (10.8). *Then we have* $e_{i+1} = \|X_{i+1} - M^{-1}\| \leqslant \epsilon^*/\bar{\mu}$ *and* $\hat{e}_{i+1} = \|Y_{i+1} - M^{-1}\| \leqslant C\epsilon^*/\bar{\mu}$ *for* $i + 1 \geqslant i^* + 1$; *furthermore, the matrices* X_{i+1} *and* Y_{i+1} *are computed in* $i^* + 1$ *steps* (9.1) *by using* $\mathrm{O}((i^*+1)(v_{\rho,n}(\nabla_{B,A}) + v_{\rho,n}(\nabla_{A,B}))\rho)$ *flops*; *this is* $\mathrm{O}((i^*+1)\rho^2 n \log^d n)$ *flops for* $d \leqslant 2$ *under Assumption* 3.1.

11. Extension to the case of the Stein-type operators

We may extend our algorithms by replacing the Sylvester type operators $\nabla_{A,B}$ by the Stein type operators $\Delta_{A,B}$ (see Theorem 2.1). This involves some minor changes. First, the formula for the recovery of a matrix W from its image $\Delta_{A,B}(W)$ changes versus the recovery from $\nabla_{A,B}(W)$, and all the algorithms change respectively. Second, minor changes appear in the computation of the $\Delta_{B,A}$-generators of the matrices $X_{i+1} = 2Y_i - Y_i M Y_i$ because of the changes of the expressions for the matrix products and inverses. Let us specify.

Assume that the matrices M and A are nonsingular and write $\Delta_{A,B}(M) = GH^{\mathrm{T}}$. Then we have the following expression for the inverse:

$$\Delta_{B,A}(M^{-1}) = M^{-1} - BM^{-1}A = M^{-1}A^{-1}\Delta_{A,B}(M)M^{-1}A = G_{-}H_{-}^{\mathrm{T}},$$

where $G_{-} = M^{-1}A^{-1}G$ and $H_{-}^{\mathrm{T}} = H^{\mathrm{T}}M^{-1}A$. Similarly, if M and B are nonsingular, we have

$$\Delta_{B,A}(M^{-1}) = M^{-1} - BM^{-1}A = BM^{-1}\Delta_{A,B}(M)B^{-1}M^{-1} = G_{+}H_{+}^{\mathrm{T}},$$

where $G_{+} = BM^{-1}G$ and $H_{+}^{\mathrm{T}} = H^{\mathrm{T}}B^{-1}M^{-1}$. In both cases, the length of the $\Delta_{A,B}$-generator G, H for M equals the length of the respective $\Delta_{B,A}$-generator for M^{-1}.

Likewise, for the product YMY we deduce the following expression without any nonsingularity assumptions:

$$YMY - BYMYA = (Y - BYA)MY + BYAM(Y - BYA) - BY(M - AMB)YA.$$

This expression furnishes us with $\Delta_{B,A}$-generators (of Stein type) of length at most 3ρ for Y_iMY_i and, consequently, for $X_{i+1} = 2Y_i - Y_iMY_i$, provided that M and Y_i are given with their $\Delta_{A,B}$- and $\Delta_{B,A}$-generators of length at most ρ, respectively.

The resulting changes of our algorithms will be further specified in the following two subsections. On some more elaborate techniques that enable extension of our algorithms to some operators $\Delta_{A,B}$ where both matrices A and B are singular, see, e.g., Theorem 11.2 of Chapter 2 in [4].

11.1. Specific changes for Subalgorithm 7.1

We change the requirements to the output of Subalgorithm 7.1 and its computation as follows:

New output. A $\Delta_{B,A}$-generator of a length at most ρ for a matrix Y_i satisfying the bound

$$\|Y_i - M^{-1}\|_2 \leqslant (1 + (1 + \|A\|_2\|B\|_2)\nu^{-})e_{2,i} \tag{11.1}$$

for $e_{2,i}$ of (7.1) and ν^{-} of Definition 5.2. The latter change is motivated by the following argument extending the proof of Lemma 7.1:

$$\|X_i - BX_iA - M^{-1} + BM^{-1}A\|_2$$

$$\begin{aligned}
&= \|(X_i - M^{-1}) - B(X_i - M^{-1})A\|_2 \\
&\leqslant \|(X_i - M^{-1})\|_2 + \|B\|_2\|(X_i - M^{-1})\|_2\|A\|_2 \\
&\leqslant (1 + \|A\|_2\|B\|_2)\|(X_i - M^{-1})\|_2 \\
&\leqslant (1 + \|A\|_2\|B\|_2)e_i.
\end{aligned}$$

Assumption (8.2) for $i = 0$, which ensures rapid convergence of Algorithm 6.1, turns into the following one in the Stein type case:

$$(1 + (1 + \|A\|_2\|B\|_2)\nu^-)\kappa(M)r_0 \leqslant \theta < 1 \tag{11.2}$$

for $\nu^- = \nu_{\rho,2}(\varDelta_{A,B})$ of Definition 5.2.

11.2. Specific changes for Subalgorithm 9.1

We change Subalgorithm 9.1 as follows:

New input. A positive integer ρ, a pair of $n \times n$ operator matrices A and B, which define a strongly regular operator $\varDelta_{A,B}$, A being nonsingular, a $\varDelta_{A,B}$-generator of length at most ρ for a nonsingular matrix M, where $\rho = \mathrm{rank}(\varDelta_{A,B}(M)) = \mathrm{rank}(\varDelta_{B,A}(M^{-1}))$, and a $\varDelta_{B,A}$-generator (G_{i+1}, H_{i+1}) of length at most 3ρ for matrix X_{i+1} of Eq. (6.1).

New output. A $\varDelta_{B,A}$-generator of a length at most ρ for matrix Y_{i+1} satisfying the bound

$$\hat{e}_{i+1} = \|Y_{i+1} - M^{-1}\| \leqslant \bar{C}_i e_i \tag{11.3}$$

for e_i of (7.1), $\hat{e}_i$ of (7.2),

$$\bar{C}_i = \nu^-\|GH^{\mathrm{T}}\| \cdot \|A\| \cdot \|A^{-1}\|(e_i + 2\|M^{-1}\|), \tag{11.4}$$

and ν^- of Definition 5.2.

New computations. Recall Theorem 3.3, compute and output the matrices $U_{i+1} = X_{i+1}A^{-1}G$, $W_{i+1}^{\mathrm{T}} = H^{\mathrm{T}}X_{i+1}A$.

The latter changes are motivated by the following argument extending Lemma 9.1. Express the matrix

$$E_j = U_jW_j^{\mathrm{T}} - UW^{\mathrm{T}} = X_jA^{-1}GH^{\mathrm{T}}X_jA - M^{-1}A^{-1}GH^{\mathrm{T}}M^{-1}A$$

as follows:

$$\begin{aligned}
E_j = {}& U_jH^{\mathrm{T}}(X_j - M^{-1})A + (X_j - M^{-1})A^{-1}GW_j^{\mathrm{T}} \\
&-(X_j - M^{-1})A^{-1}GH^{\mathrm{T}}(X_j - M^{-1})A.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\|E_i\| &\leqslant \|GH^{\mathrm{T}}\| \cdot \|A\| \cdot \|A^{-1}\|e_j(e_j + 2\|M^{-1}\|) \\
&= \|GH^{\mathrm{T}}\|\kappa(A)e_j(e_j + 2\|M^{-1}\|).
\end{aligned}$$

Corollaries 10.1–10.3, which specify the convergence rate of Algorithm 6.1 combined with Subalgorithm 9.1 and the computational cost of the resulting algorithm, are extended to the Stein type case too. Here is the respective extension of Corollary 10.1, which immediately implies appropriate extension of Corollaries 10.2 and 10.3.

Corollary 11.1. *Assume relations* (7.1)–(7.3), (10.1)–(10.5) *and the bounds* $e_i \leqslant \|M^{-1}\|$ *for all i. Write* $\hat{C} = 3\nu^{-}\|GH^{\mathrm{T}}\|\kappa(M)\|Y_0\|/(1-r_0)$, $\hat{\mu} = \hat{C}^2\|M\|$. *Then we have* $\hat{\mu}e_{i+1} < (\hat{\mu}e_i)^2 < (\hat{\mu}e_1)^{2^i}$ *for* $i = 1, 2, \dots$

12. Norm estimates via truncation of singular values

To complete our analysis presented in the previous sections, we must estimate the norms ν^{-} of the inverse displacement operators $\nabla^{-1}_{A,B}$ or $\varDelta^{-1}_{A,B}$ that we associate with the input matrices of our Newton-structured iteration (see Definition 5.2). In this and the following three sections, we will apply three approaches to the solution of this problem (see yet alternative techniques in [39]).

In this section, we will estimate the norms ν^{-} for the operators associated with the four basic classes of structured matrices, that is, Toeplitz-like, Hankel-like, Vandermonde-like, and Cauchy-like matrices. The estimates will depend on the choice of the basic bilinear representation of such matrices). Technically, we will follow the line of the Appendix of [27]. In particular we will rely on the truncation of singular values of the displacement $L(M)$ and will use the two following simple auxiliary facts.

Fact 12.1. *We have* $\|Z_f(\mathbf{v})\|_l = \|\mathbf{v}\|_1$ *for any scalar* f, $|f| \leqslant 1$, *any vector* $\mathbf{v}$ *and* $l = 1, \infty$; *furthermore*, $\|D(\mathbf{v})\|_l \leqslant \|\mathbf{v}\|_l$.

Fact 12.2. *For an orthogonal L-generator* (G, H) *of a matrix* (*see* (4.1)–(4.3)), *we have* $\|\mathbf{g}_i\|_2 = \|\mathbf{h}_i\|_2 = \sigma_i(GH^{\mathrm{T}})$, $i = 1, \dots, \rho$, $\|GH^{\mathrm{T}}\|_2 = \sigma_1^2(GH^{\mathrm{T}})$.

Now we are ready to estimate the norms ν^{-}. We write $\mathbf{1} = (1)_{j=0}^{n-1}$, $\mathbf{t}^n = (t_j^n)_{j=0}^{n-1}$.

Theorem 12.1. *Let* $\mathbf{s} = (s_i)$ *and* $\mathbf{t} = (t_j)$ *be a pair of vectors of dimension n filled with* $2n$ *distinct coordinates*, *none of the* t_j *being zero. Let* $\nabla = \nabla_{A,B}$ *and* $\varDelta = \varDelta_{A,B}$ *be nonsingular operators of* (2.1) *and* (2.2). *Then we have the following bounds on the l-norm of the inverse operators* ∇^{-1} *and* $\varDelta^{-1}$ *over the* $n \times n$ *complex matrices*:

$$\nu_{\rho,l}(\varDelta^{-1}_{A,B}) \leqslant \rho n^{1.5}, \quad \nu_{\rho,l}(\nabla^{-1}_{A,B}) \leqslant \rho n^{1.5}, \tag{12.1}$$

where $A, B \in \{Z_f, Z_f^{\mathrm{T}} : |f| \leqslant 1\}$.

$$\begin{aligned}\nu_{\rho,l}(\varDelta^{-1}_{A,B}) &\leqslant \rho\sqrt{n}\|D^{-1}(\mathbf{1}-f\mathbf{t}^n)V(\mathbf{t})\|_l,\\ \nu_{\rho,l}(\nabla^{-1}_{A,B}) &\leqslant \rho\sqrt{n}\|D^{-1}(\mathbf{1}-f\mathbf{t}^n)V(\mathbf{t})\|_l,\end{aligned} \tag{12.2}$$

where $(A, B) \in \{(D(\mathbf{t}), Z_f), (D(\mathbf{t}), Z_f^{\mathrm{T}}), (Z_f, D(\mathbf{t})), (Z_f^{\mathrm{T}}, D(\mathbf{t}))\}$.

$$\nu_{\rho,l}\left(\nabla^{-1}_{D(\mathbf{t}),D(\mathbf{s})}\right) \leqslant \rho\sqrt{n}\|D(\mathbf{s})C(\mathbf{s},\mathbf{t})\|_l \tag{12.3}$$

for $l = 1, 2, \infty$, $1 \leqslant \rho \leqslant n$. *For* $l = 2$, *all these upper bounds are decreased by the factor of* $\sqrt{n}$.

Proof. The bounds of Theorem 12.1 are obtained based on the bilinear representation for each matrix M of $\varDelta$-rank (respectively, ∇-rank) at most ρ such that $\varDelta(M) = GH^{\mathrm{T}}$ (respectively, $\nabla(M) = GH^{\mathrm{T}}$) for the matrices G and H of (2.5), where $\alpha = \rho$. That is, we deduce bounds (12.1)–(12.3) based on the equations of Theorem 2.2 and Remark 2.1.

We first deduce from Theorem 2.2 (a) that $\|M\| \leqslant \sum_{i=1}^{\rho} \|Z_e(\mathbf{g}_i)Z_f^{\mathrm{T}}(\mathbf{h}_i)\|$ for M of part (a) of Theorem 2.2. By applying Facts 12.1, 12.2, and Theorem 5.1, we obtain that $\|Z(\mathbf{g}_i)\|_1 = \|\mathbf{g}_i\|_1 \leqslant \sigma_i\sqrt{n}$, $\|Z^{\mathrm{T}}(\mathbf{h}_i)\|_1 = \|Z(\mathbf{h}_i)\|_\infty \leqslant \|\mathbf{h}_i\|_1 \leqslant \sigma_i\sqrt{n}$, $\|Z(\mathbf{g}_i)Z^{\mathrm{T}}(\mathbf{h}_i)\|_l \leqslant \sigma_i^2 n$ for $l = 1, \infty$ and for all i. Therefore, $\|M\|_l \leqslant n\sum_{i=1}^{\rho}\sigma_i^2 \leqslant n\rho\sigma_1^2 = n\rho\|GH^{\mathrm{T}}\|_2$ for $l = 1$ and $l = \infty$.

By using Theorem 5.1, we reconcile the l-norm and the 2-norm on both sides of the latter inequality and arrive at bounds (12.1). Furthermore, we combine our bounds on $\|M\|_l$ for $l = 1, \infty$ with the bound $\|M\|_2^2 \leqslant \|M\|_1\|M\|_\infty$ of Theorem 5.1 and improve the bound of (12.1) for $l = 2$ by the factor of $\sqrt{n}$. Eqs. (12.2) and (12.3) are derived similarly, based on the expressions of Theorem 2.2 and on Remark 2.1. (We leave details to the reader.) □

Remark 12.1. The operators $\varDelta$ and ∇ are associated with Toeplitz-like and Hankel-like matrices (for (12.1)), Vandermonde-like matrices (for (12.2)), and Cauchy-like matrices (for (12.3)).

13. Norm estimates where operators matrices are *f*-potent

In this section, we will estimate the norm ν^- for the operators associated with Toeplitz-like, Hankel-like, Vandermonde-like, and Chebyshev–Vandermonde-like matrices where at least one of the operator matrices C ($C = A$ or $C = B$) is *f*-potent, that is, $C^n = fI$. This is the case for $C = Z_f$ and $C = Z_f^{\mathrm{T}}$.

We will explicitly estimate ν^- for the Stein type operators L, but we may extend the estimate immediately to the case of the operators (2.1) of Sylvester type provided that at least one of the operator matrices A and B is nonsingular. Indeed, recall Theorem 2.1 and observe that the matrix equation $\nabla_{A,B}(M) = A\varDelta_{A^{-1},B}(M)$ implies that $\nu^-_{\rho,1}(\Delta_{A^{-1},B}) \geqslant \|A^{-1}\|_1 \nu^-_{\rho,1}(\nabla_{A,B})$ and similarly $\nabla_{A,B}(M) = -\Delta_{A,B^{-1}}(M)B$ implies that $\nu^-_{\rho,1}(\Delta_{A,B^{-1}}) \geqslant \|B^{-1}\|_1 \nu^-_{\rho,1}(\nabla_{A,B})$.

We will start with auxiliary results (of independent interest), first of which will enable us to invert the operator $L = \Delta_{A,B}$ (bi)linearly where some annihilation polynomials for the matrices A and B are avaliable. This approach was used in [8,16,42] in order to express Toeplitz-like matrices via their displacements.

Theorem 13.1. *For all* $k \geqslant 1$, *we have*

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

Proof. Note that $A^i M B^i = A^{i+1} M B^{i+1} + A^i \Delta_{A,B}(M) B^i$, sum these matrix equations for $i = 0, 1, \ldots, k-1$, and cancel the identical terms that appear on both sides of the resulting equation. □

For $k = p$, we obtain the following corollary.

Corollary 13.1. *Suppose that* $A^p = aI$ *and/or* $B^q = bI$ (*that is*, A *is an a-potent matrix of order p and/or B is a b-potent matrix of order q*). *Then*

$$M = \left(\sum_{i=0}^{p-1} A^i \Delta_{A,B}(M) B^i \right) (I - aB^p)^{-1} \tag{13.1}$$

and/or

$$M = (I - bA^q)^{-1} \left(\sum_{i=0}^{q-1} A^i \Delta_{A,B}(M) B^i \right),$$

respectively.

Corollary 13.2. *Let* $L = \Delta_{A,B}$, *where* $A^k = fI$ *for some positive integer k. Then we have* (13.1) *for* $a = f$, $p = k$ *and*, *consequently*,

$$\nu^- \leqslant \left(1 + \|A\|\|B\| + \cdots + \|A^{k-1}\|\|B^{k-1}\|\right) \|(I - fB^k)^{-1}\|. \tag{13.2}$$

Likewise, *if* $B^k = fI$, *then we have*

$$\nu^- \leqslant \left(1 + \|A\|\|B\| + \cdots + \|A^{k-1}\|\|B^{k-1}\|\right) \|(I - fA^k)^{-1}\|. \tag{13.3}$$

Next, we will specialize Corollary 13.2 to some specific classes of structured matrices. We will use the following notation: **s** and **t** denote a pair of vectors of dimension n filled with $2n$ distinct coordinates s_i and t_j, none of the t_j being zero (as in

Theorem 12.1), and we write $t_- = \min_j |t_j|$, $t_+ = \max_j |t_j|$, $\hat{Z} = 2\sum_{i=1}^{\lfloor n/2 \rfloor}(-1)^{i-1} Z^{2i-1}$.

Theorem 13.2. *Let $a, b, e,$ and f be four scalars such that $|e| \leqslant 1,\ |f| \leqslant 1,\ a = 1/\max\{|1-e|, |1-f|\} \geqslant 1/2,\ b = 1/|1-f| \geqslant 1/2$. Then we have the following bounds*:

$$\nu_{\rho,l}\left(\Delta^{-1}_{A,B}\right) \leqslant na, \tag{13.4}$$

where $A, B \in \{Z_e, Z_f, Z_e^{\mathrm{T}}, Z_f^{\mathrm{T}}\},\ l = 1, \infty$,

$$\nu_{\rho,l}\left(\Delta^{-1}_{D^{-1}(\mathbf{t}),Z_f}\right) \leqslant \begin{cases} \frac{1-\left(\frac{1}{t_-}\right)^n}{1-\frac{1}{t_-}} b & \textit{if } t_- \neq 1, \\ nb & \textit{if } \frac{1}{t_-} = 1, \end{cases} \tag{13.5}$$

where $l = 1, 2, \infty$,

$$\nu_{\rho,l}\left(\Delta^{-1}_{D(\mathbf{t}),Z_f}\right) \leqslant \begin{cases} \frac{1-t_+^n}{1-t_+} b & \textit{if } t_+ \neq 1, \\ nb & \textit{if } t_+ = 1, \end{cases} \tag{13.6}$$

where $l = 1, 2, \infty$,

$$\nu_{\rho,1}\left(\Delta^{-1}_{D(\mathbf{t}),\hat{Z}}\right) \leqslant \begin{cases} 1 + \left(\frac{n}{t_+}\right)\left(\frac{1-\left(\frac{2}{t_+}\right)^{n-1}}{1-\left(\frac{2}{t_+}\right)}\right) & \textit{if } t_+ \neq 2, \\ 1 + \frac{n(n-1)}{2} & \textit{if } t_+ = 2. \end{cases} \tag{13.7}$$

Proof. The bounds of Theorem 13.2 are obtained based on bound (13.2) and (13.3) applied for $k = n$ and the operators Δ of (13.4)–(13.7). Bound (13.4) is immediate because $\|Z_c\|_l = \|Z_c^{\mathrm{T}}\|_l = \cdots = \|Z_c^{n-1}\|_l = \|(Z_c^{\mathrm{T}})^{n-1}\|_l = 1$ for $|c| \leqslant 1$ and $l = 1, \infty$. Bound (13.5) immediately follows from (13.3) for $k = n$ because $\|D^{-1}(\mathbf{t})\|_l = \frac{1}{t_-}$, and therefore, we have

$$\nu^-_{\rho,l} \leqslant 1 + \frac{1}{t_-} + \cdots + \frac{1}{t_-^{n-1}}.$$

The proof of (13.6) is similar to the proof of (13.5). Finally, let us prove (13.7). Recall that $\hat{Z} = 2\sum_{i=1}^{\lfloor n/2 \rfloor}(-1)^{i-1}Z^{2i-1}$ and deduced that

$$\hat{Z}^n = 0, \quad \|\hat{Z}\|_1 \leqslant n,$$

$$\|\hat{Z}^{n-1}\|_1 = 2^{n-1}\left\|\left(\sum_{i=1}^{\lfloor n/2 \rfloor}(-1)^{i-1}Z^{2i-1}\right)^{n-1}\right\|_1 \leqslant 2^{n-1}n/2 = 2^{n-2}n,$$

$$\nu^-_{\rho,1} \leqslant 1 + \frac{1}{t_+}n + \cdots + \frac{1}{t_+^{n-1}}2^{n-2}n$$

$$= 1 + \frac{n}{t_+}\left(1 + \frac{2}{t_+} + \cdots + \left(\frac{2}{t_+}\right)^{n-2}\right). \qquad \square$$

Remark 13.1. The operators Δ are associated with Toeplitz-like and Hankel-like matrices for Δ of (13.4), Vandermonde-like matrices for Δ of (13.5) and (13.6), and Chebyshev–Vandermonde-like matrices [21] for Δ of (13.7).

14. Eigenvalue technique for the estimation of operator norms

In the following section, we will estimate the norm ν^- in the cases of the operators associated with the Cauchy-like and Toeplitz + Hankel-like matrices. Corollary 13.2 is not sufficient in these cases, but we will rely on the following result:

Theorem 14.1. *Let $\Delta = \Delta_{A,B}$ be a Stein type operator of* (2.2) *with $n \times n$ operator matrices A and B. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of the matrix A. Write $A_{\lambda_i} = A - \lambda_i I$, $B_{\lambda_i} = I - \lambda_i B$. Assume that the matrices B_{λ_i} are nonsingular for all i. Then we have*

$$\begin{aligned} M = {} & \Delta(M)B_{\lambda_1}^{-1} + A_{\lambda_1}\Delta(M)B_{\lambda_2}^{-1}BB_{\lambda_1}^{-1} \\ & + \cdots + A_{\lambda_1}\cdots A_{\lambda_{n-1}}\Delta(M)B_{\lambda_n}^{-1}\cdots BB_{\lambda_1}^{-1} \end{aligned} \tag{14.1}$$

and, *consequently*,

$$\begin{aligned} \nu_{\rho,1}(\Delta^{-1}) \leqslant {} & \|B_{\lambda_1}^{-1}\|_1 + \|A_{\lambda_1}\|_1\|B\|_1\|B_{\lambda_1}^{-1}\|_1\|B_{\lambda_2}^{-1}\|_1 \\ & + \cdots + \|A_{\lambda_1}\|_1\cdots\|A_{\lambda_{n-1}}\|_1\|B^{n-1}\|_1\|B_{\lambda_1}^{-1}\|_1\cdots\|B_{\lambda_n}^{-1}\|_1. \end{aligned} \tag{14.2}$$

Proof. Let λ be any eigenvalue of the matrix A. We have

$$\begin{aligned} \Delta(M) &= M - \lambda MB + \lambda MB - AMB \\ &= M(I - \lambda B) - (A - \lambda I)MB \\ &= MB_\lambda - A_\lambda MB, \end{aligned}$$

and, consequently,

$$\Delta(M)B_\lambda^{-1} = M - A_\lambda MBB_\lambda^{-1}. \tag{14.3}$$

For $\lambda = \lambda_1$, $\lambda = \lambda_2$ we obtain that

$$\Delta(M)B_{\lambda_1}^{-1} = M - A_{\lambda_1}MBB_{\lambda_1}^{-1}, \tag{14.4}$$

$$\Delta(M)B_{\lambda_2}^{-1} = M - A_{\lambda_2}MBB_{\lambda_2}^{-1}. \tag{14.5}$$

Pre-multiply (14.5) by A_{λ_1}, post-multiply by $BB_{\lambda_1}^{-1}$ and obtain that

$$A_{\lambda_1}\Delta(M)B_{\lambda_2}^{-1}BB_{\lambda_1}^{-1} = A_{\lambda_1}MBB_{\lambda_1}^{-1} - A_{\lambda_1}A_{\lambda_2}MBB_{\lambda_2}^{-1}BB_{\lambda_1}^{-1}. \tag{14.6}$$

Add (14.4) to (14.6) and obtain that

$$\Delta(M)B_{\lambda_1}^{-1} + A_{\lambda_1}\Delta(M)B_{\lambda_2}^{-1}BB_{\lambda_1}^{-1} = M - A_{\lambda_1}A_{\lambda_2}MBB_{\lambda_2}^{-1}BB_{\lambda_1}^{-1}. \tag{14.7}$$

Substitute $\lambda = \lambda_3$ into Eq. (14.3). Pre-multiply the resulting equation by the first term on the left-hand side of (14.7) and post-multiply it by the second term, then add (14.7) to the resulting equation. Repeat this process recursively and in n steps obtain the following equation:

$$\begin{aligned}&\Delta(M)B_{\lambda_1}^{-1} + A_{\lambda_1}\Delta(M)B_{\lambda_2}^{-1}BB_{\lambda_1}^{-1} + \cdots + A_{\lambda_1}\cdots A_{\lambda_{n-1}}\Delta(M)BB_{\lambda_n}^{-1}\cdots BB_{\lambda_1}^{-1}\\ &\quad = M - A_{\lambda_1}A_{\lambda_2}\cdots A_{\lambda_n}MBB_{\lambda_n}^{-1}\cdots BB_{\lambda_1}^{-1}.\end{aligned}$$

This implies (14.1) since $A_{\lambda_1}\cdots A_{\lambda_n} = 0$. □

15. Specific norm bounds based on the eigenvalue techniques

Let us apply Theorem 14.1 to the operators $\Delta_{D(\mathbf{s}),D(\mathbf{t})}$ associated with Cauchy-like matrices and $\Delta_{Y_{00},Y_{11}}$ associated with Toeplitz + Hankel-like matrices [6,21], where $Y_{00} = Z + Z^{\mathrm{T}}$, $Y_{11} = Y_{00} + \mathbf{e}_0\mathbf{e}_0^{\mathrm{T}} + \mathbf{e}_{n-1}\mathbf{e}_{n-1}^{\mathrm{T}}$. We have the following auxiliary results, which in particular show the diagonalization of the matrices Y_{00}, Y_{11} [20].

Theorem 15.1. *Let*

$$S = \left(\sqrt{\frac{2}{n+1}}\sin\frac{ij\pi}{n+1}\right)_{i,j=1}^{n},$$

$$Q = \left(\sqrt{\frac{2}{n}}q_j\cos\frac{(2i-1)(j-1)\pi}{2n}\right)_{i,j=1}^{n}$$

denote the (normalized) matrices of the Discrete Sine Transform I *and the Discrete Cosine Transform* II, *respectively, where* $q_1 = 1/\sqrt{2}$, $q_j = 1$ *for* $j > 1$. *Then we have* $S = S^{\mathrm{T}}$, $S^2 = Q^{\mathrm{T}}Q = T$, *so that* $\|S\|_2 = \|Q\|_2 = 1$, $\|S\|_l \leqslant \sqrt{n}$, $\|Q\|_l \leqslant \sqrt{n}$ *for* $l = 1, \infty$.

Theorem 15.2. $SY_{00}S = D_S$, $Q^{\mathrm{T}}Y_{11}Q = D_Q$, *where*

$$D_S = \operatorname{diag}\left(2\cos\frac{k\pi}{n+1}\right)_{k=1}^{n}, \quad \|D_S\|_l < 2,$$

$$D_Q = \operatorname{diag}\left(2\cos\frac{k\pi}{n}\right)_{k=0}^{n-1}, \quad \|D_Q\|_l = 2, \;\; l = 1, 2, \infty.$$

We will also use the following simple estimates:

$$\|Y_{11} - \lambda_i I\|_1 \leqslant 2 + |\lambda_i| \tag{15.1}$$

for any λ_i.

Fact 15.1. *For all scalars λ_i, we have $\|(Y_{00}^{-1})_{\lambda_i}^{-1}\|_1 \leqslant 2n/\psi_i$, where*

$$\psi_i = \min_j \left|1 - 2\lambda_i/\cos\left(\frac{j\pi}{n+1}\right)\right|.$$

Proof. By definition, $(Y_{00}^{-1})_{\lambda_i} = I - \lambda_i Y_{00}^{-1}$. Recall Theorems 15.1 and 15.2 and obtain that $(Y_{00}^{-1})_{\lambda_i} = S(I - \lambda_i D_S^{-1})S$. Therefore,

$$\|(Y_{00}^{-1})_{\lambda_i}^{-1}\|_1 = \|S(I - \lambda_i D_S^{-1})^{-1}S\|_1 \leqslant 2n/\psi_i. \qquad \square$$

Now, we are ready to state our next theorem.

Theorem 15.3. *As in Theorem* 12.1, *let* **s** *and* **t** *be a pair of vectors of dimension n filled with $2n$ distinct coordinates s_i and t_j, none of the t_j being zero. Let $\Delta = \Delta_{A,B}$ be an operator L of* (2.2). *Let λ_i denote the eigenvalues of the matrix A, $i = 1, \ldots, n$. Let us write $t_- = \min_j |t_j|$, $s_+ = \max_j |s_j|$, $\phi_i = \min_j |1 - s_i t_j|$, $\phi = \min_i \phi_i$, $\psi = \min_i \psi_i$, $\rho_i = 2 + |\lambda_i|$, and $\rho = \max_i \rho_i$ for ψ_i of Fact* 15.1. *Then we have*

$$\nu^- = \nu_{\rho,1}\left(\Delta^{-1}_{D(\mathbf{s}),D^{-1}(\mathbf{t})}\right) \leqslant \begin{cases} \dfrac{t_-}{\phi}\dfrac{1 - \left(\frac{2s_+t_-}{\phi}\right)^n}{1 - \frac{2s_+t_-}{\phi}} & \text{if } \frac{2s_+t_-}{\phi} \neq 1, \\ \dfrac{t_-n}{\phi} & \text{if } \frac{2s_+t_-}{\phi} = 1, \end{cases} \tag{15.2}$$

for $A = D(\mathbf{s})$, $B = D^{-1}(\mathbf{t})$,

$$\nu^- = \nu_{\rho,1}\left(\Delta^{-1}_{Y_{11},(Z+Z^{\mathrm{T}})^{-1}}\right) \leqslant \begin{cases} \dfrac{2n}{\psi}\dfrac{1 - \left(\frac{2n\rho}{\psi}\right)^n}{1 - \frac{2n\rho}{\psi}} & \text{if } \frac{2n\rho}{\psi} \neq 1, \\ \dfrac{2n^2}{\psi} & \text{if } \frac{2n\rho}{\psi} = 1, \end{cases} \tag{15.3}$$

for $A = Y_{11}$, $B = (Z + Z^{\mathrm{T}})^{-1}$.

Proof. Recall that $A_{\lambda_i} = A - \lambda_i I$ and deduce that $\|A_{\lambda_i}\|_1 = \|A - \lambda_i I\|_1 \leqslant \|A\|_1 + |\lambda_i|$. We have $A = D(\mathbf{s})$ in (15.2). Therefore, $\lambda_i = s_i$, $|\lambda_i| \leqslant \max_j |s_j| = s_+$, and $\|A_{\lambda_i}\|_1 \leqslant 2s_+$ for all i. Similarily, for $B = D^{-1}(\mathbf{t})$ of (15.2), we obtain that $\|B_{\lambda_i}\|_1 \leqslant t_-/\phi_i$. Substitute both norm bounds into (14.2) for $\Delta = \Delta_{D(\mathbf{s}),D^{-1}(\mathbf{t})}$ and obtain that

$$\nu^- \leqslant \frac{t_-}{\phi_1} + \frac{2s_+t_-^2}{\phi_1\phi_2} + \cdots + \frac{(2s_+)^{n-1}t_-^n}{\phi_1\cdots\phi_n}$$

for ν^- of (15.2). Since we have $\phi = \min_i \phi_i$, it follows that

$$\nu^- \leqslant \frac{t_-}{\phi} + 2s_+\frac{t_-^2}{\phi^2} + \cdots + (2s_+)^{n-1}\frac{t_-^n}{\phi^n},$$

and we obtain (15.2). By combining (14.2), (15.1), and Fact 15.1, we obtain that

$$\nu^- \leqslant \frac{2n}{\psi_1} + \rho_1\frac{(2n)^2}{\psi_1\psi_2} + \cdots + \rho_1\cdots\rho_{n-1}\frac{(2n)^n}{\psi_1\cdots\psi_n}$$

for ν^- of (15.3). Substitute $\psi = \min \psi_i$ and $\rho = \max \rho_i$, obtain that

$$\begin{aligned}\nu^- &\leqslant \frac{2n}{\psi} + \rho\frac{(2n)^2}{\psi^2} + \cdots + \rho^{n-1}\frac{(2n)^n}{\psi^n}\\ &= \frac{2n}{\psi}\left(1 + \frac{2n\rho}{\psi} + \cdots + \left(\frac{2n\rho}{\psi}\right)^{n-1}\right),\end{aligned}$$

and arrive at (15.3). □

16. The unification and transformation approaches

As an alternative to the unification of the study of Newton's iteration for various matrix structures, one may transform the problem to the Toeplitz-like or Cauchy-like cases to extend the cited successful algorithms of [27,30,33,40] to other classes of structured matrices. This is a special case of the general idea of extending successful algorithms from one class to other classes of structured matrices. The idea was proposed in [26] together with the sample transformations in all directions among Toeplitz-like, Hankel-like, Vandermonde-like, and Cauchy-like matrices. The approach turns out to be quite powerful. Some of the current best practical algorithms for solving Toeplitz and Toeplitz-like linear systems of equations reduce them to Cauchy-like linear systems. Furthermore, structured matrix transformations of this kind have been used for handling matrix singularities, for computational improvements of polynomial interpolation and multipoint evaluation as well as algebraic decoding, and in the computational complexity analysis of structured matrix operations (cf. [6,15,35,41], [25, Section 6] and [31,32]). The unification and transformation approaches may effectively complement each other.

As a rule, the unification approach enables a deeper insight into the subject and its more comprehensive treatment. In some cases, transformations are costly (in terms of extra flops and numerical stability problems involved), and the unification approach can be more effective. In other cases, transformations are inexpensive (e.g., from

Toeplitz/Hankel-like to Cauchy-like matrices [6,15]) and can enhance the domain where the algorithms are effective.

In particular, Theorem 2.2 and the bounds of Table 2 imply that $v_{\alpha,n}(L) = \mathrm{O}(\alpha n \log^d n)$, where $d = 1$ for operators L associated with Toeplitz-like and Hankel-like matrices versus $d = 2$ in the Vandermonde-like and Cauchy-like cases. The difference in the computational cost is extended to various other operations with these matrices, in particular to Newton's iteration. The standard transformations to Toeplitz-like or Hankel-like cases (cf. [26] and [4, Section 12 of Chapter 2]) enable respective decrease of the asymptotic upper bounds on the number of flops involved in the algorithms.

17. Conclusion

There are several interesting directions for the extension and further study of the Newton-structured iteration.

1. Some useful singular displacement operators L are strongly regular on the linear space of matrices that vanish on a fixed subset S of their entries having small cardinality. In particular, S may consist of the first and/or last column (and/or) row of a matrix (see ∇_{Z_f,Z_f}) or of its diagonal (see $\nabla_{D(\mathbf{s}),D(\mathbf{s})}$). In such a case the matrix is generated by its entries of the set S and by its displacement $L(M)$ together. (Bi)linear expressions of Section 2 as well as iteration (6.1) and its analysis can be modified and extended, respectively (see [32,33]). Alternatively, the problem can be reduced to the one with a strongly regular operator [32].
2. If no initial approximation Y_0 to the matrix M^{-1} is available, such an approximation can be generated in a homotopic process [27,31,32,34]. In this approach, the algorithm of [27] approximates a Toeplitz-like matrix M^{-1} within the output error norm bound ϵ by using $\mathrm{O}((\gamma + \log\log(1/\epsilon)) r v_{r,n}(L))$ flops with $\gamma = \mathrm{O}(\log\kappa(M))$. The study is extended to other well-known classes of input matrices in [31,32,34].
3. There are alternative recipes for choosing an initial approximation Y_0 such as $Y_0 = M^*/(\|M\|_1\|M\|_\infty)$ and of the convergence acceleration by scaling the iterates Y_i for all i as well as by shifting to higher order processes. These recipes rely on the observation that the singular vectors of the residuals $I - MY_i$ are invariant in i provided that $X_i = Y_i$ for all i (see [38] and references therein). Compression of the displacements of the computed approximations, however, perturbs the matrices X_i so that the singular vectors of the matrices Y_i vary with i. Therefore, the entire approach remains valid only to the extent to which the perturbation caused by the compression makes no significiant impact on the singular spaces. Estimation and restriction of such an impact is the subject of further theoretical and experimental study [32,34]. Its preliminary results are encouraging.
4. Newton's iteration for the computation of the inverse and Moore–Penrose generalized inverse of a matrix is a special case of residual correction methods [17,34,38],

yielding faster convergence (in particular with using scaling). Would application of such more general methods improve our algorithms?

5. Newton's iteration is a well-known tool for the solution of matrix equation, in particular for the computation of the polar decomposition, the square roots and the sign function for general matrices [11,12,14,19]. Could the known methods be improved where the input matrix is structured? Our methods would be immediately extended whenever the output matrices have short displacement generators.

Acknowledgement

The authors are grateful to a referee for helpful comments.

References

[1] G.S. Ammar, P. Gader, A variant of the Gohberg–Semencul formula involving circulant matrices, SIAM J. Matrix Anal. Appl. 12 (3) (1991) 534–541.

[2] D.A. Bini, B. Meini, Approximate displacement rank and applications, preprint.

[3] D. Bini, V.Y. Pan, Improved parallel computations with Toeplitz-like and Hankel-like matrices, Linear Algebra Appl. 188/189 (1993) 3–29.

[4] D. Bini, V.Y. Pan, Polynomial and Matrix Computations, Vol. 1, Fundamental algorithms, Birkhäuser, Boston, 1994.

[5] R.H. Chan, M.K. Ng, Conjugate gradient methods for toeplitz systems, SIAM Rev. 38 (1996) 427–482.

[6] I. Gohberg, T. Kailath, V. Olshevsky, Fast Gaussian elimination with partial pivoting for matrices with displacement structure, Math. Comput. 64 (1995) 1557–1576.

[7] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[8] I. Gohberg, V. Olshevsky, Circulants, displacements and decompositions of matrices, Integral Equations and Operator Theory 15 (5) (1992) 730–743.

[9] I. Gohberg, V. Olshevsky, Complexity of multiplication with vectors for structured matrices, Linear Algebra Appl. 202 (1994) 163–192.

[10] I. Gohberg, A. Semencul, On the inversion of finite Toeplitz matrices and their continous analogs, Mat. Issled. 2 (1972) 187–224.

[11] N.J. Higham, Newton's method for the matrix square root, Math. Comput. 46 (1986) 537–550.

[12] N.J. Higham, Computing the polar decomposition—with applications, SIAM J. Sci. Statist. Comput. 7 (4) (1986) 1160–1174.

[13] N.J. Higham, Fast solution of Vandermonde-like systems involving orthogonal polynomials, IMA J. Numer. Anal. 8 (1988) 473–486.

[14] N.J. Higham, The matrix sign decomposition and its relations to polar decomposition, Linear Algebra Appl. 212/213 (1994) 3–20.

[15] G. Heinig, Inversion of generalized Cauchy matrices and the other classes of structured matrices, in: Linear algebra for signal processing, IMA Volume in Mathematics and its Applications, Vol. 69, Springer, Berlin, 1995, pp. 95–114.

[16] G. Heinig, K. Rost, Algebraic methods for Toeplitz-like matrices and operators, in: I. Gohberg, (Ed.), Operator Theory: Advances and Applications, vol. 13, Birkhäuser, Basel, 1984.

[17] E. Issacson, H.B. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

[18] T. Kailath, S.Y. Kung, M. Morf, Displacement ranks of matrices and linear equations, J. Math. Anal. Appl. 68 (2) (1979) 395–407.

[19] C. Kenney, A.J. Laub, On scaling Newton's method for polar decomposition and the matrix sign function, SIAM J Matrix Anal. Appl. 13 (3) (1992) 698–706.

[20] T. Kailath, V. Olshevsky, Displacement structure approach to discrete transform based preconditioners of G. Strang type and of T. Chan type, Calcolo 33 (1996) 191–208.

[21] T. Kailath, V. Olshevsky, Displacement structure approach to polynomial Vandermonde and related matrices, Linear Algebra Appl. 285 (1997) 37–67.

[22] T. Kailath, A. Sayed (Eds.), Fast Reliable Algorithms for Matrices with Structure, SIAM, Philadelphia, PA, 1999.

[23] H. Lu, Solution of Vandermonde-like systems and confluent Vandermonde-like systems, SIAM J. Matrix Anal. Appl. 17 (1) (1996) 127–138.

[24] V. Olshevsky, V.Y. Pan, A unified superfast algorithm for boundary rational tangential interpolation problem and for inversion and factorization of dense structured matrices, in: Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, 1998, pp. 192–201.

[25] V. Olshevsky, M.A. Shokrollahi, A displacement approach to efficient decoding of algebraic-geometric codes, in: Proceedings of the 31st Annual Symposium on Theory of Computing, ACM Press, New York, 1999, pp. 235–244.

[26] V.Y. Pan, On computations with dense structured matrices, Math. Comput. 55 (191) (1990) 179–190.

[27] V.Y. Pan, Parallel solution of Toeplitz-like linear systems, J. Complexity 8 (1992) 1–21.

[28] V.Y. Pan, Parametrization of Newton's iteration for computations with structured matrices and applications, Comput. Math. Appl. 24 (3) (1992) 61–75.

[29] V.Y. Pan, Decreasing the displacement rank of a matrix, SIAM J. Matrix Anal. Appl. 14 (1) (1993) 118–121.

[30] V.Y. Pan, Concurrent iterative algorithm for Toepliz-like linear systems, IEEE Trans. Parallel and Distributed Systems 4 (5) (1993) 592–600.

[31] V.Y. Pan, Nearly optimal computations with structured matrices, in: Proceedings of the 11th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA'2000), ACM Press, New York, and SIAM, Philadephia, 2000, pp. 953–962.

[32] V.Y. Pan, Structured Matrices and Polynomials: Unified Superfast Algorithms, Birkhäuser/Springer, Basel/Berlin, 2001, to appear.

[33] V.Y. Pan, S. Branham, R. Rosholt, A. Zheng, Newton's Iteration for Structured Matrices and Linear Systems of Equations, SIAM volume on Fast Reliable Algorithms for Matrices with Structure, SIAM, Philadelphia, PA, 1999.

[34] V.Y. Pan, M. Kunin, R. Rosholt, Homotopic residual correction processes, to appear.

[35] V.Y. Pan, E. Landowne, A. Sadikou, O. Tiga, A new approach to fast polynomial interpolation and multipoint evaluation, Comput. Math. Appl. 25 (9) (1993) 25–30.

[36] V.Y. Pan, Y. Rami, Newton's iteration for the inversion of structured matrices, in: D. Bini, E. Tyrtyshnikov, P. Yalamov (Eds.), Structured Matrices: Recent Developments in Theory and Computation, Nova Science Publishers, USA, 2001.

[37] V.Y. Pan, Y. Rami, X. Wang, Newton's iteration for the inversion of structured matrices, in: Proceedings of the 14th International Symposium on Math. Theory of Network and Systems (MTNS'2000), June 2000.

[38] V.Y. Pan, R. Schreiber, An improved Newton iteration for the generalized inverse of a matrix, with applications, SIAM J. Sci. Statist. Comput. 12 (5) (1991) 1109–1131.

[39] V.Y. Pan, X. Wang, Inversion of displacement operators, to appear.

[40] V.Y. Pan, A. Zheng, X. Huang, O. Dias, Newton's iteration for inversion of Cauchy-like and other structured matrices, J. Complexity 13 (1997) 108–124.

[41] V.Y. Pan, A. Zheng, X. Huang, Y. Yu, Fast multipoint polynomial evaluation and interpolation via computations with structured matrices, Ann. Numer. Math. 4 (1997) 483–510.

[42] D.H. Wood, Product rules for the displacement of nearly-Toeplitz matrices, Linear Algebra Appl. 188 (1993) 641–663.