Computing specified generators of structured matrix inverses

Claude-Pierre Jeannerod
LIP - ENS de Lyon
INRIA
claude-pierre.jeannerod@ens-lyon.fr

Christophe Mouilleron
LIP - ENS de Lyon
Université de Lyon
christophe.mouilleron@ens-lyon.org

ABSTRACT

The asymptotically fastest known divide-and-conquer methods for inverting dense structured matrices are essentially variations or extensions of the Morf/Bitmead-Anderson algorithm. Most of them must deal with the growth in length of intermediate generators, and this is done by incorporating various generator compression techniques into the algorithms. One exception is an algorithm by Cardinal, which in the particular case of Cauchy-like matrices avoids such growth by focusing on well-specified, already compressed generators of the inverse. In this paper, we extend Cardinal's method to a broader class of structured matrices including those of Vandermonde, Hankel, and Toeplitz types. Besides, some first experimental results illustrate the practical interest of the approach.

Categories and Subject Descriptors

I.1.2 [**Computing Methodologies**]: Symbolic and Algebraic Manipulation—*Algebraic Algorithms*

General Terms

Algorithms, Theory

Keywords

Structured linear algebra, matrix inversion

1. INTRODUCTION

Since [10], a classical way of exploiting the structure of dense matrices is via the displacement rank approach: typically, $n \times n$ matrices are represented by pairs (G, H) of $n \times \alpha$ matrices such that $\mathcal{L}(\mathsf{A}) = \mathsf{G}\mathsf{H}^T$ for some linear operator $\mathcal{L}$ called a displacement. Classical choices for $\mathcal{L}$ are Stein's displacement $\Delta[\mathsf{M}, \mathsf{N}] : \mathsf{A} \mapsto \mathsf{A} - \mathsf{M}\mathsf{A}\mathsf{N}$ and Sylvester's displacement $\nabla[\mathsf{M}, \mathsf{N}] : \mathsf{A} \mapsto \mathsf{M}\mathsf{A} - \mathsf{A}\mathsf{N}$. With respect to a given displacement, (G, H) is called a *generator* of length α for A, and A is considered to be structured when α is "small" (in a sense that depends on the context) compared to n. According to the unified treatment [19], many of the structures encountered in practice are covered by the following operator matrices: for a field $\mathbb{K}$ and a positive integer n, let

$$\mathsf{M}, \mathsf{N} \in \{\mathbb{D}(\mathsf{x}),\ \mathbb{Z}_{n,\varphi},\ \mathbb{Z}_{n,\psi}^T\}, \quad \mathsf{x} \in \mathbb{K}^n, \quad \varphi, \psi \in \mathbb{K}, \tag{1}$$

with $\mathbb{D}(\mathsf{x})$ the diagonal matrix whose entry (i, i) is the ith coefficient x_i of vector x, and $\mathbb{Z}_{n,\varphi}$ the $n \times n$ unit φ-circulant matrix having a φ in position $(1, n)$, ones in positions $(i + 1, i)$, and zeros everywhere else.

When a structured matrix $\mathsf{A} \in \mathbb{K}^{n \times n}$ is invertible, its inverse A^{-1} is known to be structured too, and some asymptotically fast algorithms are available for computing length-α generators for A^{-1} and linear system solutions, whose costs in terms of operations in $\mathbb{K}$ are in $O\tilde{~}(\alpha^2\, n)$ (see [19] and the references therein) and, since more recently, in $O\tilde{~}(\alpha^{\omega-1}\, n)$ (see [2, 3]). (Here and hereafter the $O\tilde{~}$ notation hides all logarithmic factors.) Such algorithms are essentially variations or extensions of the Morf/Bitmead-Anderson (MBA) divide-and-conquer approach [13, 1]. In practice, they apply to important types of structures like those of (1). However, most of these algorithms must deal with the growth in length of intermediate generators, and this is done by recursively using a generator compression stage which, given matrices $\mathsf{G}, \mathsf{H} \in \mathbb{K}^{n \times \beta}$ such that $\mathsf{G}\mathsf{H}^T$ has rank $\alpha \le \beta$, computes matrices $\mathsf{G}_c, \mathsf{H}_c$ that satisfy $\mathsf{G}_c\mathsf{H}_c^T = \mathsf{G}\mathsf{H}^T$ but now have exactly α columns; see [17, 15, 16, 11, 12] and [19, §4.6].

One exception is a variant of MBA due to Cardinal [4, 5]: assuming Sylvester's displacement equation

$$\nabla[\mathsf{M}, \mathsf{N}](\mathsf{A}) = \mathsf{G}\mathsf{H}^T \tag{2}$$

and in the particular case where both M and N are diagonal (Cauchy-like structure), Cardinal's algorithm completely avoids generator compression by directly computing

$$\mathsf{Y} = -\mathsf{A}^{-1}\mathsf{G}, \qquad \mathsf{Z} = \mathsf{A}^{-T}\mathsf{H}. \tag{3}$$

As already noted in [9] —and this is readily verified by pre- and postmultiplying (2) with the inverse of A—, the matrix pair (Y, Z) is a $\nabla[\mathsf{N}, \mathsf{M}]$-generator of length α for A^{-1}. Due to its very special form, we shall call it a *specified generator* for the inverse of A.

The goal of this paper is to extend Cardinal's algorithm beyond the Cauchy-like structure and to show that, in MBA and for Sylvester's displacement, generator compression can be systematically avoided by targeting a specified generator for the inverse, rather than just an arbitrary one of length α. More precisely, our three main contributions can be summarized as follows:

First, we propose a recursive formula that allows to factor a specified generator of the inverse for A in terms of specified generators for the inverse of its upper-left block A_{11} and for the inverse of the Schur complement of A_{11} in A.

Second, we show how to reduce the computation of specified inverse generators for the structures defined in (1) to the computation of specified inverse generators for the three basic cases below:

$$(\mathsf{M},\mathsf{N}) \in \left\{ \big(\mathbb{D}(\mathsf{x}),\mathbb{D}(\mathsf{y})\big), \big(\mathbb{D}(\mathsf{x}),\mathbb{Z}_{n,0}^T\big), \big(\mathbb{Z}_{n,0},\mathbb{Z}_{n,0}^T\big) \right\}. \quad (4)$$

For each of those three structures, which are of the Cauchy-like, Vandermonde-like, and Hankel-like types, respectively, we further give and analyze explicit algorithms for computing a specified generator of the inverse. These algorithms are compression-free and thus, in that sense, simpler to analyze and implement than traditional MBA variants. Moreover, although removing generator compression does not affect the overall asymptotic costs, it yields smaller dominant terms.

Third, we report on a first set of experiments done with our C++ implementation of MBA and of several of the new compression-free algorithms. For the Cauchy-like structure, for example, the speed-ups compared to MBA are by a factor from 4.6 to 6.7. This suggests that our extension of Cardinal's compression-free approach may yield algorithms that are not only simpler but also significantly faster in practice.

Outline of the paper. After some notation and preliminaries in §2, some properties of specified generators are studied in §3. Then §4 gives a compression-free algorithm for the case where M and N^T are lower triangular. The algorithm is specialized to the Cauchy-like and Vandermonde-like structures in §4.1 and §4.2, and then extended in §4.3 to the irregular Hankel-like case $(\mathsf{M},\mathsf{N}) = (\mathbb{Z}_{n,0},\mathbb{Z}_{n,0}^T)$. Experiments are reported in §5 and we conclude in §6.

2. NOTATION AND PRELIMINARIES

Here and hereafter, $\mathbb{I}_n$ is the identity matrix of order n, $\mathsf{e}_{n,i}$ is the ith unit vector of $\mathbb{K}^n$, and $\mathbb{J}_n$ is the reflexion matrix of order n, whose (i,j) entry is 1 if $i+j=n+1$, and 0 otherwise. For $\mathsf{A} \in \mathbb{K}^{n\times m}$, a_{ij} denotes its (i,j) entry and a_j its jth column. Also, for $\alpha \le m$, we write $\mathsf{A}^{\mapsto\alpha}$ for the matrix $[\mathsf{a}_1,\dots,\mathsf{a}_\alpha] \in \mathbb{K}^{n\times\alpha}$.

Given positive integers n_1 and n_2 such that $n_1+n_2=n$, we will often partition A, G, H, M, N into $n_i \times n_j$ blocks as

$$\mathsf{A} = \begin{bmatrix} \mathsf{A}_{11} & \mathsf{A}_{12} \\ \mathsf{A}_{21} & \mathsf{A}_{22} \end{bmatrix}, \quad \mathsf{G} = \begin{bmatrix} \mathsf{G}_1 \\ \mathsf{G}_2 \end{bmatrix}, \quad \mathsf{H} = \begin{bmatrix} \mathsf{H}_1 \\ \mathsf{H}_2 \end{bmatrix},$$
$$\mathsf{M} = \begin{bmatrix} \mathsf{M}_{11} & \mathsf{M}_{12} \\ \mathsf{M}_{21} & \mathsf{M}_{22} \end{bmatrix}, \quad \mathsf{N} = \begin{bmatrix} \mathsf{N}_{11} & \mathsf{N}_{12} \\ \mathsf{N}_{21} & \mathsf{N}_{22} \end{bmatrix}. \quad (5)$$

We shall write μ and ν for the rank of, respectively, M_{12} and N_{21}. Consequently, those two matrices can be written

$$\mathsf{M}_{12} = \mathsf{U}_1\mathsf{V}_2^T, \qquad \mathsf{N}_{21} = \mathsf{U}_2\mathsf{V}_1^T \quad (6)$$

for some full column rank matrices $\mathsf{U}_1 \in \mathbb{K}^{n_1\times\mu}$, $\mathsf{V}_2 \in \mathbb{K}^{n_2\times\mu}$, $\mathsf{U}_2 \in \mathbb{K}^{n_2\times\nu}$, and $\mathsf{V}_1 \in \mathbb{K}^{n_1\times\nu}$.

The Schur complement of A_{11} in A is written S:

$$\mathsf{S} = \mathsf{A}_{22} - \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{A}_{12}.$$

Recall that S is nonsingular if A_{11} and A are nonsingular; if A is strongly regular then so are A_{11} and S. Finally, let

$$\mathsf{E} = \begin{bmatrix} \mathbb{I}_{n_1} & \\ -\mathsf{A}_{21}\mathsf{A}_{11}^{-1} & \mathbb{I}_{n_2} \end{bmatrix}, \quad \mathsf{F} = \begin{bmatrix} \mathbb{I}_{n_1} & -\mathsf{A}_{11}^{-1}\mathsf{A}_{12} \\ & \mathbb{I}_{n_2} \end{bmatrix}. \quad (7)$$

From $\mathsf{EAF} = \operatorname{diag}(\mathsf{A}_{11},\mathsf{S})$, we deduce the following classical recursive factorization of the inverse of A [19, p. 157]:

$$\mathsf{A}^{-1} = \mathsf{F} \begin{bmatrix} \mathsf{A}_{11}^{-1} & \\ & \mathsf{S}^{-1} \end{bmatrix} \mathsf{E}. \quad (8)$$

2.1 Properties of Sylvester's displacement

The properties below show how to deduce, given a $\nabla[\mathsf{M},\mathsf{N}]$-generator for A, a generator for various matrices related to A. All of them appear in/follow immediately from [18, 19].

Generation of the transpose. Let $\mathsf{A},\mathsf{G},\mathsf{H}$ be as in (2). By transposing the identity $\mathsf{MA}-\mathsf{AN} = \mathsf{GH}^T$, we obtain

$$\nabla[\mathsf{N}^T,\mathsf{M}^T](\mathsf{A}^T) = -\mathsf{HG}^T, \quad (9)$$

so that the pair $(-\mathsf{H},\mathsf{G})$ is a $\nabla[\mathsf{N}^T,\mathsf{M}^T]$-generator of A^T.

Generation of products. One has the following classical rule for generating matrix products [19, p. 10]:

$$\nabla[\mathsf{M},\mathsf{N}](\mathsf{AB}) = \nabla[\mathsf{M},*](\mathsf{A})\,\mathsf{B} + \mathsf{A}\,\nabla[*,\mathsf{N}](\mathsf{B}), \quad (10)$$

for any matrix $*$ of conforming dimensions. Applying this rule twice, we can straightforwardly deduce explicit formulas for generating products of three matrices:

Lemma 1. *Let $\mathsf{A},\mathsf{G},\mathsf{H}$ be as in (2) and, for two matrices P_1 and P_2, let*

$$\widetilde{\mathsf{A}} = \mathsf{P}_1\mathsf{A}\mathsf{P}_2. \quad (11)$$

If $\nabla[\widetilde{\mathsf{M}},\mathsf{M}](\mathsf{P}_1) = \mathsf{G}_{\mathsf{P}_1}\mathsf{H}_{\mathsf{P}_1}^T$ and $\nabla[\mathsf{N},\widetilde{\mathsf{N}}](\mathsf{P}_2) = \mathsf{G}_{\mathsf{P}_2}\mathsf{H}_{\mathsf{P}_2}^T$ then

$$\nabla[\widetilde{\mathsf{M}},\widetilde{\mathsf{N}}](\widetilde{\mathsf{A}}) = \widetilde{\mathsf{G}}\widetilde{\mathsf{H}}^T,$$

$$\widetilde{\mathsf{G}} = [\mathsf{P}_1\mathsf{G}|\mathsf{G}_{\mathsf{P}_1}|\mathsf{P}_1\mathsf{A}\mathsf{G}_{\mathsf{P}_2}], \quad \widetilde{\mathsf{H}} = [\mathsf{P}_2^T\mathsf{H}|\mathsf{P}_2^T\mathsf{A}^T\mathsf{H}_{\mathsf{P}_1}|\mathsf{H}_{\mathsf{P}_2}]. \quad (12)$$

As an example, let us mention three special cases which we will use later: assuming $\mathsf{M},\mathsf{N} \in \{\mathbb{Z}_{n,\varphi},\mathbb{Z}_{n,\varphi}^T\}$, let first

$$(\mathsf{P}_1,\mathsf{P}_2) = (\mathbb{I}_n,\mathbb{J}_n) \quad \text{and} \quad (\widetilde{\mathsf{M}},\widetilde{\mathsf{N}}) = (\mathsf{M},\mathsf{N}^T).$$

Then obviously $\nabla[\widetilde{\mathsf{M}},\mathsf{M}](\mathsf{P}_1)$ is zero and, using the facts that $\mathbb{J}_n^2 = \mathbb{I}_n$ and $\mathbb{J}_n\mathbb{Z}_{n,\varphi}\mathbb{J}_n = \mathbb{Z}_{n,\varphi}^T$ (see [19, p. 24]), we deduce that $\nabla[\mathsf{N},\widetilde{\mathsf{N}}](\mathsf{P}_2)$ is zero as well. Consequently, since $\mathbb{J}_n$ is symmetric, applying (12) yields

$$\nabla[\mathsf{M},\mathsf{N}^T](\mathsf{A}\,\mathbb{J}_n) = \mathsf{G}(\mathbb{J}_n\mathsf{H})^T. \quad (13a)$$

Similarly, exchanging the roles of P_1 and P_2 yields

$$\nabla[\mathsf{M}^T,\mathsf{N}](\mathbb{J}_n\mathsf{A}) = (\mathbb{J}_n\mathsf{G})\mathsf{H}^T, \quad (13b)$$

while taking $\mathsf{P}_1 = \mathsf{P}_2 = \mathbb{J}_n$ gives

$$\nabla[\mathsf{M}^T,\mathsf{N}^T](\mathbb{J}_n\mathsf{A}\,\mathbb{J}_n) = (\mathbb{J}_n\mathsf{G})(\mathbb{J}_n\mathsf{H})^T. \quad (13c)$$

Generation of submatrices. From (2) and (6) and the partitioning into blocks we deduce that, for $i,j \in \{1,2\}$, submatrix A_{ij} satisfies the following matrix equation

$$\nabla[\mathsf{M}_{ij},\mathsf{N}_{ij}](\mathsf{A}_{ij}) = \mathsf{G}_{ij}\mathsf{H}_{ij}^T, \quad (14)$$

where, in particular (see for example [18, Proposition 4.4]),

$$\mathsf{G}_{11} = [\mathsf{G}_1| -\mathsf{U}_1|\mathsf{A}_{12}\mathsf{U}_2] \in \mathbb{K}^{n_1\times(\alpha+\mu+\nu)}, \quad (15a)$$
$$\mathsf{H}_{11} = [\mathsf{H}_1|\mathsf{A}_{21}^T\mathsf{V}_2|\mathsf{V}_1] \in \mathbb{K}^{n_1\times(\alpha+\mu+\nu)}. \quad (15b)$$

Generation of Schur complements. By combining [18, Proposition 4.5] with (6), we have the following description of the structure of the Schur complement S of A_{11} in A:

$$\nabla[\mathsf{M}_{22},\mathsf{N}_{22}](\mathsf{S}) = \mathsf{G}_S\mathsf{H}_S^T, \quad (16)$$

with G_S and H_S the two matrices in $\mathbb{K}^{n_2\times(\alpha+\mu+\nu)}$ given by

$$\mathsf{G}_\mathsf{S} = [\mathsf{G}_2 - \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{G}_1 | \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{U}_1 | -\mathsf{S}\mathsf{U}_2], \quad (17a)$$

$$\mathsf{H}_\mathsf{S} = [\mathsf{H}_2 - \mathsf{A}_{12}^T\mathsf{A}_{11}^{-T}\mathsf{H}_1 | \mathsf{S}^T\mathsf{V}_2 | \mathsf{A}_{12}^T\mathsf{A}_{11}^{-T}\mathsf{V}_1]. \quad (17b)$$

When the operator matrices M and N^T are lower triangular, one has $\mu=\nu=0$ and the above formulas for generating the Schur complement can thus be simplified as follows (see [8, Theorem 2.3], [14, Lemma 3.1], [19, §5.4]):

$$\mathsf{G}_\mathsf{S} = \mathsf{G}_2 - \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{G}_1, \quad \mathsf{H}_\mathsf{S} = \mathsf{H}_2 - \mathsf{A}_{12}^T\mathsf{A}_{11}^{-T}\mathsf{H}_1. \quad (18)$$

2.2 Computing with basic structures

We conclude our preliminaries by reviewing three basic invertible displacement operators that we shall repeatedly use in the sequel, as well as some associated cost functions. Here we assume that (2) holds in the rectangular case, that is, for $\mathsf{A}\in\mathbb{K}^{n\times m}$, $\mathsf{G}\in\mathbb{K}^{n\times\alpha}$, and $\mathsf{H}\in\mathbb{K}^{m\times\alpha}$; this assumption will allow us to handle off-diagonal blocks in Section 4. Recall also that $\nabla[\mathsf{M},\mathsf{N}]$ is invertible if and only if the spectra of M and N are disjoint [19, p. 123].

Cauchy-like structure. For $\mathsf{x}\in\mathbb{K}^n$ and $\mathsf{y}\in\mathbb{K}^m$, assume

$$\mathsf{M}=\mathbb{D}(\mathsf{x}), \quad \mathsf{N}=\mathbb{D}(\mathsf{y}), \quad x_i\neq y_j \text{ for all } (i,j). \quad (19a)$$

Then $\nabla[\mathsf{M},\mathsf{N}]$ is invertible and it is known [7] (see also [19, p. 8] and [20, Lemma 2.1]) that (2) is equivalent to

$$\mathsf{A}=\textstyle\sum_{j=1}^{\alpha}\mathbb{D}(\mathsf{g}_j)\,\mathbb{C}(\mathsf{x},\mathsf{y})\,\mathbb{D}(\mathsf{h}_j), \quad (19b)$$

with $\mathbb{C}(\mathsf{x},\mathsf{y})$ the n by m Cauchy matrix $[1/(x_i-y_j)]_{i,j}$.

Vandermonde-like structure. For $\mathsf{x}\in\mathbb{K}^n$, assume now

$$\mathsf{M}=\mathbb{D}(\mathsf{x}), \quad \mathsf{N}=\mathbb{Z}_{m,0}^T, \quad x_i\neq 0 \text{ for all } i. \quad (20a)$$

Then $\nabla[\mathsf{M},\mathsf{N}]$ is invertible and, in this case, A can be recovered as follows (see [19, Example 4.4.6(d)]):

$$\mathsf{A}=\textstyle\sum_{j=1}^{\alpha}\,\mathbb{D}(\mathsf{x}^{-1}\cdot\mathsf{g}_j)\,\mathbb{V}(\mathsf{x}^{-1},m)\,\mathbb{U}(\mathsf{h}_j), \quad (20b)$$

where, for $\mathsf{x}\in\mathbb{K}^n$, $\mathbb{V}(\mathsf{x},m)$ is the n by m Vandermonde matrix whose (i,j) entry equals x_i^{j-1}, and $\mathbb{U}(\mathsf{h}_j)$ is the m by m upper triangular Toeplitz matrix whose first row is h_j^T.

Hankel-like structure. Assume finally that

$$\mathsf{M}=\mathbb{Z}_{n,1}, \quad \mathsf{N}=\mathbb{Z}_{m,0}^T. \quad (21a)$$

Since $\mathbb{Z}_{n,1}$ and $\mathbb{Z}_{m,0}^T$ have disjoint spectra, $\nabla[\mathsf{M},\mathsf{N}]$ is invertible. In addition, we can recover A as follows:

$$\mathsf{A}=\textstyle\sum_{j=1}^{\alpha}\,\mathbb{T}^{n\times m}(\mathsf{g}_j)\,\mathbb{L}(\mathsf{h}_j)\,\mathbb{J}_m, \quad (21b)$$

where, for $\mathsf{x}\in\mathbb{K}^n$, $\mathbb{T}^{n\times m}(\mathsf{x})$ is the n by m Toeplitz matrix $[\mathsf{x}_{1+(i-j+m) \bmod n}]_{i,j}$, and where $\mathbb{L}(\mathsf{h}_j)$ is the m by m lower triangular Toeplitz matrix whose first column is h_j. (A proof of (21b) is given in Appendix A.)

Cost functions. Our algorithms in the next sections will essentially require the ability to efficiently evaluate products of the form $\mathsf{A}\mathsf{v}$ and $\mathsf{A}^T\mathsf{v}$, where A has one of the three basic structures above, m is of the same order of n, and v consists of one or several vectors.

In order to relate the costs of our algorithms in Section 4 to the costs of such products, we introduce the following functions. For the Cauchy-like structure (19a), let $\mathsf{MM}_\mathsf{C}:\mathbb{N}_{>0}\times\mathbb{N}_{>0}\times\mathbb{N}_{>0}\to\mathbb{R}_{\geq 0}$ be such that, for $\mathsf{A}\in\mathbb{K}^{n\times n}$ given by the right hand-side of (19b) and $\mathsf{v}\in\mathbb{K}^{n\times\beta}$, the products $\mathsf{A}\mathsf{v}$ and $\mathsf{A}^T\mathsf{v}$ can be computed using at most $\mathsf{MM}_\mathsf{C}(\alpha,n,\beta)$ operations in $\mathbb{K}$. We define the functions MM_V and MM_H in a similar way for, respectively, the Vandermonde-like and Hankel-like structures. Also, when $\beta=\alpha$ we shall simply write $\mathsf{MM}_*(\alpha,n)$, for $*=\mathsf{C},\mathsf{V},\mathsf{H}$.

Following [6, p. 242], we write $\mathsf{M}(n)$ for the cost of multiplying two polynomials of degree less than n over $\mathbb{K}[x]$, and we assume that $\mathsf{M}(n)$ is "superlinear", that is, $\mathsf{M}(n)/n$ is nondecreasing.

It is known (see for example [19]) that $\mathbb{C}(\mathsf{x},\mathsf{y})$ is $-\mathbb{C}(\mathsf{y},\mathsf{x})$ and that multiplying $\mathbb{C}(\mathsf{x},y)$, $\mathbb{V}(\mathsf{x},n)$, or $\mathbb{V}(\mathsf{x},n)^T$ by a vector can be done in time $O(\mathsf{M}(n)\log(n))$ via (transposed) multipoint evaluation. Hence by a straightforward application of the summation formulas (19b), (20b), and (21b), one has

$$\mathsf{MM}_*(\alpha,n,1)\in O(\alpha\,\mathsf{M}(n)\log(n)) \quad \text{for } *=\mathsf{C},\mathsf{V},$$

$$\mathsf{MM}_\mathsf{H}(\alpha,n,1)\in O(\alpha\,\mathsf{M}(n)).$$

We shall also use the three basic properties given below:

LEMMA 2. *Let $k,\ell\in O(1)$. Then*

$$\mathsf{MM}_\mathsf{V}(\alpha+k,n,\alpha)\in\mathsf{MM}_\mathsf{V}(\alpha,n)+O(\alpha\,\mathsf{M}(n)\log(n)), \quad (22a)$$

$$\mathsf{MM}_\mathsf{H}(\alpha+k,n,\alpha)\in\mathsf{MM}_\mathsf{H}(\alpha,n)+O(\alpha\,\mathsf{M}(n)), \quad (22b)$$

and, for $*=\mathsf{C},\mathsf{V},\mathsf{H}$,

$$\mathsf{MM}_*(k\alpha,n,\ell\alpha)\in k\ell\,\mathsf{MM}_*(\alpha,n)+O(\alpha\,n). \quad (23)$$

PROOF. To get (22) note that, for all $*$, $\mathsf{MM}_*(\alpha+k,n,\alpha)$ is in $\mathsf{MM}_*(\alpha,n,\alpha)+\mathsf{MM}_*(k,n,\alpha)+O(\alpha n)$. Indeed, one can evaluate our sum of $\alpha+k$ products by adding the first α terms and the last k terms separately, and then combining the two intermediate results. Since moreover $\mathsf{MM}_*(k,n,\alpha)\leq\alpha\mathsf{MM}_*(k,n,1)$, (22a) and (22b) follow from the complexities of $\mathsf{MM}_\mathsf{V}(\alpha,n)$ and $\mathsf{MM}_\mathsf{H}(\alpha,n)$ mentioned above. To establish (23), notice that a sum of $k\alpha$ terms for $\ell\alpha$ vectors can be evaluated via k sums of α terms for α vectors plus a final sum in $O(\alpha\,n)$, repeated ℓ times. $\square$

Finally, we assume as for $\mathsf{M}(n)$ that the function $\mathsf{M}(\cdot,n)$ is "superlinear," that is, $\mathsf{MM}(\cdot,n)/n$ is nondecreasing. This assumption will allow us to simplify the cost bounds of the algorithms of Section 4 and can be easily supported by "naive" implementations in $O\tilde{}(\alpha^2 n)$ as those used in Section 5.

3. PROPERTIES OF SPECIFIED GENERATORS OF THE MATRIX INVERSE

3.1 Recovery after matrix transformations

We recalled in Section 2 some formulas for generating the matrix $\widetilde{\mathsf{A}}\in\{\mathsf{A}^T,\mathsf{P}_1\mathsf{A}\mathsf{P}_2\}$ from some generators of the matrix A. Conversely, we give in the theorem below some formulas for recovering specified generators of the inverse of A from specified generators of the inverse of $\widetilde{\mathsf{A}}$.

THEOREM 1. *Let $\mathsf{A}\in\mathbb{K}^{n\times n}$ be invertible and let $\mathsf{G},\mathsf{H}\in\mathbb{K}^{n\times\alpha}$ and $\mathsf{Y},\mathsf{Z}\in\mathbb{K}^{n\times\alpha}$ be as in (2) and (3). Let $\widetilde{\mathsf{A}}\in\mathbb{K}^{n\times n}$ be invertible and, for $\widetilde{\mathsf{G}},\widetilde{\mathsf{H}}\in\mathbb{K}^{n\times\beta}$, $\beta\geq\alpha$, define*

$$\widetilde{\mathsf{Y}}=-\widetilde{\mathsf{A}}^{-1}\widetilde{\mathsf{G}}, \qquad \widetilde{\mathsf{Z}}=\widetilde{\mathsf{A}}^{-T}\widetilde{\mathsf{H}}.$$

Then

- *for $\widetilde{\mathsf{A}}=\mathsf{A}^T$ and $(\widetilde{\mathsf{G}},\widetilde{\mathsf{H}})=(-\mathsf{H},\mathsf{G})$, one has*

$$\mathsf{Y}=-\widetilde{\mathsf{Z}}, \qquad \mathsf{Z}=\widetilde{\mathsf{Y}};$$

- *for* $\widetilde{\mathsf{A}} = \mathsf{P}_1\mathsf{A}\mathsf{P}_2$ *with* $\mathsf{P}_1, \mathsf{P}_2 \in \mathbb{K}^{n\times n}$ *invertible, and for* $\widetilde{\mathsf{G}}, \widetilde{\mathsf{H}}$ *as in (12), one has*

$$\mathsf{Y} = \mathsf{P}_2\widetilde{\mathsf{Y}}^{\mapsto\alpha}, \qquad \mathsf{Z} = \mathsf{P}_1^T\widetilde{\mathsf{Z}}^{\mapsto\alpha}.$$

PROOF. In the first case, $\widetilde{\mathsf{Y}} = -(\mathsf{A}^{-T})(-\mathsf{H}) = \mathsf{A}^{-T}\mathsf{H} = \mathsf{Z}$ and $\widetilde{\mathsf{Z}} = (\mathsf{A}^T)^{-T}(\mathsf{G}) = \mathsf{A}^{-1}\mathsf{G} = -\mathsf{Y}$. Now, in the case where $\widetilde{\mathsf{A}} = \mathsf{P}_1\mathsf{A}\mathsf{P}_2$ Lemma 1 implies that the first α columns of $\widetilde{\mathsf{Y}}$ are $\widetilde{\mathsf{Y}}^{\mapsto\alpha} = -(\mathsf{P}_1\mathsf{A}\mathsf{P}_2)^{-1}\mathsf{P}_1\mathsf{G} = \mathsf{P}_2^{-1}\mathsf{Y}$. Similarly, the first α columns of $\widetilde{\mathsf{Z}}$ are $\widetilde{\mathsf{Z}}^{\mapsto\alpha} = (\mathsf{P}_1\mathsf{A}\mathsf{P}_2)^{-T}\mathsf{P}_2^T\mathsf{H} = \mathsf{P}_1^{-T}\mathsf{Z}$. $\square$

For example, when $\mathsf{P}_1, \mathsf{P}_2 \in \{\mathbb{I}_n, \mathbb{J}_n\}$, it follows from (12) that $\beta = \alpha$. Consequently, Theorem 1 yields

$$(\mathsf{Y}, \mathsf{Z}) = (\mathbb{J}_n\widetilde{\mathsf{Y}}, \widetilde{\mathsf{Z}}) \quad \text{if } \widetilde{\mathsf{A}} = \mathsf{A}\,\mathbb{J}_n, \tag{25a}$$

$$(\mathsf{Y}, \mathsf{Z}) = (\widetilde{\mathsf{Y}}, \mathbb{J}_n\widetilde{\mathsf{Z}}) \quad \text{if } \widetilde{\mathsf{A}} = \mathbb{J}_n\mathsf{A}, \tag{25b}$$

$$(\mathsf{Y}, \mathsf{Z}) = (\mathbb{J}_n\widetilde{\mathsf{Y}}, \mathbb{J}_n\widetilde{\mathsf{Z}}) \quad \text{if } \widetilde{\mathsf{A}} = \mathbb{J}_n\mathsf{A}\,\mathbb{J}_n. \tag{25c}$$

Reduction to basic displacements. A first consequence of Theorem 1, when it comes to computing specified inverse generators, is that the nine possible displacements defined in (1) can be reduced to the three basic ones shown in (4).

First, it follows from (13a) and (25a) that the case $\mathsf{N} = \mathbb{Z}_{n,\psi}$ reduces to the case $\mathsf{N} = \mathbb{Z}_{n,\psi}^T$. Similarly, (13b) and (25b) imply that the case $\mathsf{M} = \mathbb{Z}_{n,\varphi}^T$ reduces to the case $\mathsf{M} = \mathbb{Z}_{n,\varphi}$. We thus are left with the four cases defined by

$$\mathsf{M} \in \{\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\varphi}\} \quad \text{and} \quad \mathsf{N} \in \{\mathbb{D}(\mathsf{y}), \mathbb{Z}_{n,\psi}^T\}.$$

Using (9) allows to further reduce the case where $\mathsf{M} = \mathbb{Z}_{n,\varphi}$ and $\mathsf{N} = \mathbb{D}(\mathsf{y})$ to the case where $\mathsf{M} = \mathbb{D}(\mathsf{y})$ and $\mathsf{N} = \mathbb{Z}_{n,\varphi}^T$. Due to the nature of the transformations applied to the $n \times \alpha$ generators (sign changes, permutations), the three reductions done so far imply an extra cost of only $O(\alpha\, n)$ operations in $\mathbb{K}$.

To reach (4) it remains to zero out the scalars φ and ψ. This can be done without transforming A, but only its displacement: for example, by combining the obvious identity

$$\mathbb{Z}_{n,\varphi} = \mathbb{Z}_{n,0} + \varphi\, \mathsf{e}_{n,1}\, \mathsf{e}_{n,n}^T, \tag{26}$$

with $\nabla[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,\psi}^T](\mathsf{A}) = \mathsf{G}\mathsf{H}^T$ and $\nabla[\mathbb{Z}_{n,\varphi}, \mathbb{Z}_{n,\psi}^T](\mathsf{A}) = \mathsf{G}\mathsf{H}^T$, we arrive at, respectively,

$$\nabla[\mathbb{D}(\mathsf{x}), \mathbb{Z}_{n,0}^T](\mathsf{A}) = \widetilde{\mathsf{G}}\widetilde{\mathsf{H}}^T \tag{i}$$

with $\widetilde{\mathsf{G}} = [\mathsf{G}|\psi\, \mathsf{e}_{n,1}]$ and $\widetilde{\mathsf{H}} = [\mathsf{H}|\mathsf{A}\mathsf{e}_{n,n}]$, and

$$\nabla[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^T](\mathsf{A}) = \widetilde{\mathsf{G}}\widetilde{\mathsf{H}}^T \tag{ii}$$

with $\widetilde{\mathsf{G}} = [\mathsf{G}| -\varphi\, \mathsf{e}_{n,1}|\mathsf{A}\mathsf{e}_{n,n}]$ and $\widetilde{\mathsf{H}} = [\mathsf{H}|\mathsf{A}^T\mathsf{e}_{n,n}|\psi\mathsf{e}_{n,1}]$. The last column or row of A needed to set up the matrices $\widetilde{\mathsf{G}}$ and $\widetilde{\mathsf{H}}$ can be computed in $O(\alpha\,\mathsf{M}(n)\log(n))$ —case (i)— or $O(\alpha\,\mathsf{M}(n))$ —case (ii)— field operations from the explicit bilinear expressions of A given in [19, Examples 4.4.4 and 4.4.6(d)]. Due to the shape of $\widetilde{\mathsf{G}}$, $\widetilde{\mathsf{H}}$ above, extracting the first α columns of $\widetilde{\mathsf{Y}} = \mathsf{A}^{-1}\widetilde{\mathsf{G}}$ and $\widetilde{\mathsf{Z}} = \mathsf{A}^{-T}\widetilde{\mathsf{H}}$ in time $O(\alpha\, n)$ then yields the desired specified inverse generator (Y, Z).

Reduction to strong regularity. Theorem 1 further allows to restrict to matrices that are not only invertible but strongly regular. Strong regularity, which is needed in order to apply Theorem 2 recursively, is classically obtained by preconditioning A into $\widetilde{\mathsf{A}} = \mathsf{P}_1\mathsf{A}\mathsf{P}_2$ with two random structured matrices P_1 and P_2 (see [19, §5.6]). Thus, one may generate $\widetilde{\mathsf{A}}$ as in Lemma 1, then compute an associated specified generator $(\widetilde{\mathsf{Y}}, \widetilde{\mathsf{Z}})$ of its inverse, and finally recover via Theorem 1 a specified generator (Y, Z) of the inverse of A.

Let r_1 and r_2 be two random vectors in $\mathbb{K}^n$ and whose first entry is one. Then, applying the rules of [19, p. 167], possible preconditioners for each of the three basic displacements of (4) are as follows (with $\widetilde{\mathsf{x}}, \widetilde{\mathsf{y}}$ in $\mathbb{K}^n$ and such that $\widetilde{x}_i \neq x_i$ and $\widetilde{y}_i \neq y_i$ for all i):

M, N	P_1	P_2
$\mathbb{D}(\mathsf{x}),\ \mathbb{D}(\mathsf{y})$	$\mathbb{C}(\widetilde{\mathsf{x}}, \mathsf{x})\,\mathbb{D}(\mathsf{r}_1)$	$\mathbb{C}(\mathsf{y}, \widetilde{\mathsf{y}})\,\mathbb{D}(\mathsf{r}_2)$
$\mathbb{D}(\mathsf{x}),\ \mathbb{Z}_{n,0}^T$	$\mathbb{C}(\widetilde{\mathsf{x}}, \mathsf{x})\,\mathbb{D}(\mathsf{r}_1)$	$\mathbb{L}(\mathsf{r}_2)$
$\mathbb{Z}_{n,0},\ \mathbb{Z}_{n,0}^T$	$\mathbb{U}(\mathsf{r}_1)$	$\mathbb{L}(\mathsf{r}_2)$

For all these cases, one may check that the structure of A, P_1, and P_2 allows to prepare $(\widetilde{\mathsf{G}}, \widetilde{\mathsf{H}})$ in Lemma 1 and to recover (Y, Z) in Theorem 1 in time $O(\alpha\,\mathsf{M}(n))$ or $O(\alpha\,\mathsf{M}(n)\log(n))$.

3.2 Recursive factorization formula

THEOREM 2. *Let $A \in \mathbb{K}^{n\times n}$ be nonsingular and generated by* G *and* H *as in (2). Assume that* A_{11} *is nonsingular as well, that it is generated by* G_{11} *and* H_{11} *as in (15), and let*

$$\mathsf{Y}_{11} = -\mathsf{A}_{11}^{-1}\mathsf{G}_{11}, \qquad \mathsf{Z}_{11} = \mathsf{A}_{11}^{-T}\mathsf{H}_{11}.$$

Assume further that the Schur complement S *of* A_{11} *in* A *is generated by* G_S *and* H_S *as in (17), and let*

$$\mathsf{Y}_\mathsf{S} = -\mathsf{A}_\mathsf{S}^{-1}\mathsf{G}_\mathsf{S}, \qquad \mathsf{Z}_\mathsf{S} = \mathsf{A}_\mathsf{S}^{-T}\mathsf{H}_\mathsf{S}.$$

Then the matrices Y *and* Z *in (3) satisfy*

$$\mathsf{Y} = \mathsf{F}\begin{bmatrix}\mathsf{Y}_{11}^{\mapsto\alpha}\\ \mathsf{Y}_\mathsf{S}^{\mapsto\alpha}\end{bmatrix}, \qquad \mathsf{Z} = \mathsf{E}^T\begin{bmatrix}\mathsf{Z}_{11}^{\mapsto\alpha}\\ \mathsf{Z}_\mathsf{S}^{\mapsto\alpha}\end{bmatrix},$$

where E *and* F *are the elimination matrices defined in (7).*

PROOF. Using (5) and (8), we obtain

$$-\mathsf{A}^{-1}\mathsf{G} = \mathsf{F}\begin{bmatrix}-\mathsf{A}_{11}^{-1}\mathsf{G}_1\\ -\mathsf{S}^{-1}(\mathsf{G}_2 - \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{G}_1)\end{bmatrix}.$$

It follows from (15a) and (17a) that $\mathsf{G}_1 = \mathsf{G}_{11}^{\mapsto\alpha}$ and that $\mathsf{G}_2 - \mathsf{A}_{21}\mathsf{A}_{11}^{-1}\mathsf{G}_1 = \mathsf{G}_\mathsf{S}^{\mapsto\alpha}$. The expression claimed for $\mathsf{Y} = -\mathsf{A}^{-1}\mathsf{G}$ then follows from applying the rule $\mathsf{A}(\mathsf{B}^{\mapsto\alpha}) = (\mathsf{A}\mathsf{B})^{\mapsto\alpha}$ twice, and from the definitions of Y_{11} and Y_S. The expression for Z can be obtained in a similar way, using (15b) and (17b). $\square$

A first consequence of this theorem is a "compressed" analogue of the classical recursive factorization formula (8):

$$\mathsf{Y}\,\mathsf{Z}^T = \mathsf{F}\begin{bmatrix}\mathsf{Y}_{11}^{\mapsto\alpha}\\ \mathsf{Y}_\mathsf{S}^{\mapsto\alpha}\end{bmatrix}\begin{bmatrix}\mathsf{Z}_{11}^{\mapsto\alpha}\\ \mathsf{Z}_\mathsf{S}^{\mapsto\alpha}\end{bmatrix}^T\mathsf{E}.$$

A second consequence of Theorem 2 is that, for A strongly regular, we immediately get a recursive algorithm *à la MBA* whose key steps are the computation of $(\mathsf{G}_{11}, \mathsf{H}_{11})$ and $(\mathsf{G}_\mathsf{S}, \mathsf{H}_\mathsf{S})$:

Given generators G, H of length α for A,

- Compute a generator $(\mathsf{G}_{11}, \mathsf{H}_{11})$ for A_{11} using (15);
- Recursively, compute $(\mathsf{Y}_{11}, \mathsf{Z}_{11}) = (-\mathsf{A}_{11}^{-1}\mathsf{G}_{11}, \mathsf{A}_{11}^{-T}\mathsf{H}_1)$;
- Compute a generator $(\mathsf{G}_\mathsf{S}, \mathsf{H}_\mathsf{S})$ for S using (17);

- Recursively, compute $(\mathsf{Y}_\mathsf{S}, \mathsf{Z}_\mathsf{S}) = (-\mathsf{S}^{-1}\mathsf{G}_\mathsf{S}, \mathsf{S}^{-T}\mathsf{H}_\mathsf{S})$;
- Compute $(-\mathsf{A}^{-1}\mathsf{G}, -\mathsf{A}^{-T}\mathsf{H})$ from the first α columns of Y_{11}, Y_S, Z_{11}, Z_S, using Theorem 2.

4. ALGORITHMS FOR LOWER TRIANGULAR OPERATOR MATRICES M AND N^T

In order to cover simultaneously the three displacements in (4) to which we have previously reduced, we assume in this section that both operator matrices M and N^T are lower triangular.

This assumption implies in particular that the blocks M_{12} and N_{21} in (6) are zero, so that their respective ranks μ and ν satisfy $\mu = \nu = 0$. From (15) it then follows that the submatrix A_{11} satisfies

$$\nabla[\mathsf{M}_{11}, \mathsf{N}_{11}](\mathsf{A}_{11}) = \mathsf{G}_1\mathsf{H}_1^T. \tag{27}$$

Thus, some generators of length at most α for A_{11} can be read off the first n_1 rows of some generators of length at most α for A.

Assuming that A_{11} is invertible, consider now the associated specified generators of A_{11}^{-1}, that is,

$$\mathsf{Y}_{11} = -\mathsf{A}_{11}^{-1}\mathsf{G}_1, \qquad \mathsf{Z}_{11} = \mathsf{A}_{11}^{-T}\mathsf{H}_1. \tag{28}$$

Combining the two identities in (28) with the explicit Schur complement generation formulas in (17) and (18) yields

$$\nabla[\mathsf{M}_{22}, \mathsf{N}_{22}](\mathsf{S}) = (\mathsf{G}_2 + \mathsf{A}_{21}\mathsf{Y}_{11})(\mathsf{H}_2 - \mathsf{A}_{12}^T\mathsf{Z}_{11})^T. \tag{29}$$

In other words, the precise specification of the above generators of the inverse of A_{11} can be exploited to simplify even further the generators of the Schur complement. In [4, Proposition 1], Cardinal had already noted this formula but only for the Cauchy-like structure (M and N diagonal).

Now, if we assume further that A is strongly regular (which, if randomization is allowed, makes sense in view of the probabilistic reductions to strong regularity shown in Section 3.1), we obtain the following general algorithm:

$\mathsf{GenInvLT}(\mathsf{M}, \mathsf{N}, \mathsf{G}, \mathsf{H})$

Input: $\mathsf{M}, \mathsf{N} \in \mathbb{K}^{n\times n}$ and $\mathsf{G}, \mathsf{H} \in \mathbb{K}^{n\times\alpha}$ such that M and N^T are lower triangular, and $\nabla[\mathsf{M}, \mathsf{N}](\mathsf{A}) = \mathsf{G}\mathsf{H}^T$.
Assumptions: A strongly regular, $m_{ii} \neq n_{jj}$ for all (i, j).
Output: $\mathsf{Y} = -\mathsf{A}^{-1}\mathsf{G}$ and $\mathsf{Z} = \mathsf{A}^{-T}\mathsf{H}$.

if $n = 1$ then
  Evaluate the dot product $\mathsf{G}\mathsf{H}^T$;
  Deduce the scalar A;
  $\mathsf{Y} := -\mathsf{A}^{-1}\mathsf{G}$; $\mathsf{Z} := \mathsf{A}^{-T}\mathsf{H}$;
else
  $n_1 := \lceil n/2 \rceil$; $n_2 := \lfloor n/2 \rfloor$;
  $\mathsf{G}_{11} := \mathsf{G}_1$; $\mathsf{H}_{11} := \mathsf{H}_1$;
  $(\mathsf{Y}_{11}, \mathsf{Z}_{11}) := \mathsf{GenInvLT}(\mathsf{M}_{11}, \mathsf{N}_{11}, \mathsf{G}_{11}, \mathsf{H}_{11})$;
  $\mathsf{G}_\mathsf{S} := \mathsf{G}_2 + \mathsf{A}_{21}\mathsf{Y}_{11}$; $\mathsf{H}_\mathsf{S} := \mathsf{H}_2 - \mathsf{A}_{12}^T\mathsf{Z}_{11}$;
  $(\mathsf{Y}_\mathsf{S}, \mathsf{Z}_\mathsf{S}) := \mathsf{GenInvLT}(\mathsf{M}_{22}, \mathsf{N}_{22}, \mathsf{G}_\mathsf{S}, \mathsf{H}_\mathsf{S})$;
  $\mathsf{Y} := \begin{bmatrix} \mathsf{Y}_{11} - \mathsf{A}_{11}^{-1}\mathsf{A}_{12}\mathsf{Y}_\mathsf{S} \\ \mathsf{Y}_\mathsf{S} \end{bmatrix}$; $\mathsf{Z} := \begin{bmatrix} \mathsf{Z}_{11} - \mathsf{A}_{11}^{-T}\mathsf{A}_{21}^T\mathsf{Z}_\mathsf{S} \\ \mathsf{Z}_\mathsf{S} \end{bmatrix}$;
fi;
return (Y, Z).

THEOREM 3. *Algorithm* $\mathsf{GenInvLT}$ *is correct.*

PROOF. When $n = 1$, the assumption on M and N implies that A is the scalar $(\sum_{i=1}^{\alpha} g_{1i}h_{1i})/(m_{11} - n_{11})$. Correctness then follows immediately in this case. Assume now that $n > 1$ and, in order to proceed by induction, assume correctness for $n' < n$. The matrix A_{11} is strongly regular (since A is) and it satisfies (27), where, by assumption M_{11} and N_{11}^T are both lower triangular and with disjoint diagonals. Since $n_1 < n$, the induction assumption then implies that the pair $(\mathsf{Y}_{11}, \mathsf{Z}_{11})$ returned by the first recursive call is precisely $(-\mathsf{A}_{11}^{-1}\mathsf{G}_1, \mathsf{A}_{11}^{-T}\mathsf{H}_1)$. Therefore, the computed pair $(\mathsf{G}_\mathsf{S}, \mathsf{H}_\mathsf{S})$ satisfies (29), where, by assumption, S is strongly regular (since A is) and where M_{22} and N_{22}^T are both lower triangular and have disjoint diagonals. Since $n_2 < n$, the induction assumption implies that the pair $(\mathsf{Y}_\mathsf{S}, \mathsf{Z}_\mathsf{S})$ returned by the second recursive call is exactly $(-\mathsf{S}^{-1}\mathsf{G}_\mathsf{S}, \mathsf{S}^{-T}\mathsf{H}_\mathsf{S})$. The conclusion then follows from Theorem 2. $\square$

To implement Algorithm $\mathsf{GenInvLT}$ and bound its cost, all we need is to be able to evaluate the four matrix products

$$\mathsf{A}_{21}\mathsf{Y}_{11}, \quad \mathsf{A}_{12}^T\mathsf{Z}_{11}, \quad \mathsf{A}_{11}^{-1}\mathsf{A}_{12}\mathsf{Y}_\mathsf{S}, \quad \mathsf{A}_{11}^{-T}\mathsf{A}_{21}^T\mathsf{Z}_\mathsf{S}. \tag{30}$$

In the next subsections, we study the evaluation of those expressions for each of three basic structures of the Cauchy, Vandermonde, and Hankel types. That requires in each case a detailed analysis of the structure of the matrices A_{11}^{-1}, A_{12}, A_{21}, and their transposes. Since in (30) there are two ways of parenthesizing the products of three matrices, we will also study the structure of $\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$ and $(\mathsf{A}_{21}\mathsf{A}_{11}^{-1})^T$. The parenthesizations $(\mathsf{A}_{11}^{-1}\mathsf{A}_{12})\mathsf{Y}_\mathsf{S}$ and $(\mathsf{A}_{21}\mathsf{A}_{11}^{-1})^T\mathsf{Z}_\mathsf{S}$ will be referred to as "Cardinal's trick" later on, as they have been initially used in [4] for the Cauchy-like case.

4.1 Application to Cauchy-like matrices

We consider here the specialization of Algorithm $\mathsf{GenInvLT}$ to the Cauchy-like structure defined in (19a). Partitioning the two vectors x and y conformally with A yields

$$\mathsf{x} = \begin{bmatrix} \mathsf{x}_1 \\ \mathsf{x}_2 \end{bmatrix}, \quad \mathsf{y} = \begin{bmatrix} \mathsf{y}_1 \\ \mathsf{y}_2 \end{bmatrix}, \quad \mathsf{x}_1, \mathsf{y}_1 \in \mathbb{K}^{n_1}, \quad \mathsf{x}_2, \mathsf{y}_2 \in \mathbb{K}^{n_2}.$$

LEMMA 3. *Let the matrices* $\mathsf{A}, \mathsf{G}, \mathsf{H}, \mathsf{Y}_{11}, \mathsf{Z}_{11}, \mathsf{G}_\mathsf{S}, \mathsf{H}_\mathsf{S}$ *be as in Algorithm* $\mathsf{GenInvLT}$. *Then*

- $\nabla[\mathbb{D}(\mathsf{x}_i), \mathbb{D}(\mathsf{y}_j)](\mathsf{A}_{ij}) = \mathsf{G}_i\mathsf{H}_j^T$ *for* $1 \le i, j \le 2$,
- $\nabla[\mathbb{D}(\mathsf{y}_1), \mathbb{D}(\mathsf{x}_1)](\mathsf{A}_{11}^{-1}) = \mathsf{Y}_{11}\mathsf{Z}_{11}^T$,
- $\nabla[\mathbb{D}(\mathsf{y}_1), \mathbb{D}(\mathsf{y}_2)](\mathsf{A}_{11}^{-1}\mathsf{A}_{12}) = -\mathsf{Y}_{11}\mathsf{H}_\mathsf{S}^T$,
- $\nabla[\mathbb{D}(\mathsf{x}_2), \mathbb{D}(\mathsf{x}_1)](\mathsf{A}_{21}\mathsf{A}_{11}^{-1}) = \mathsf{G}_\mathsf{S}\mathsf{Z}_{11}^T$.

PROOF. Since $\mathbb{D}(\mathsf{x})$ and $\mathbb{D}(\mathsf{y})$ are diagonal, all their off-diagonal blocks are zero, and the first identity follows from (2). To get the second identity, it suffices to pre- and postmultiply by A_{11}^{-1} both sides of the first identity for $(i, j) = (1, 1)$, and then to use the specification of Y_{11} and Z_{11}. Using the multiplication rule (10), we deduce further from the first identity for $(i, j) = (1, 2)$ and from the second one that

$$\begin{aligned} \nabla[\mathbb{D}(\mathsf{y}_1), \mathbb{D}(\mathsf{y}_2)](\mathsf{A}_{11}^{-1}\mathsf{A}_{12}) &= \mathsf{Y}_{11}\mathsf{Z}_{11}^T\mathsf{A}_{12} + \mathsf{A}_{11}^{-1}\mathsf{G}_1\mathsf{H}_2^T \\ &= \mathsf{Y}_{11}(\mathsf{Z}_{11}^T\mathsf{A}_{12} - \mathsf{H}_2^T), \end{aligned}$$

which by definition of H_S equals $-\mathsf{Y}_{11}\mathsf{H}_\mathsf{S}^T$. Similarly,

$$\begin{aligned} \nabla[\mathbb{D}(\mathsf{x}_2), \mathbb{D}(\mathsf{x}_1)](\mathsf{A}_{21}\mathsf{A}_{11}^{-1}) &= \mathsf{G}_2\,\mathsf{H}_1^T\mathsf{A}_{11}^{-1} + \mathsf{A}_{21}\mathsf{Y}_{11}\mathsf{Z}_{11}^T \\ &= (\mathsf{G}_2 + \mathsf{A}_{21}\mathsf{Y}_{11})\mathsf{Z}_{11}^T, \end{aligned}$$

which by definition of G_S equals $\mathsf{G}_\mathsf{S}\,\mathsf{Z}_{11}^T$. $\square$

THEOREM 4. *Let n be a power of two and $\mathsf{M}, \mathsf{N} \in \mathbb{K}^{n\times n}$ be as in (19a). Then Algorithm* GenInvLT *requires at most*

$$3\log(n)\,\mathsf{MM_C}(\alpha, n) + O(\alpha\, n \log(n))$$

field operations. If, in addition, the set $\{x_1,\dots,x_n,y_1,\dots,y_n\}$ has cardinality $2n$ then this bound drops to

$$2\log(n)\,\mathsf{MM_C}(\alpha, n) + O(\alpha\, n \log(n)).$$

PROOF. When $n = 1$, $\mathsf{A} = (\sum_{i=1}^{\alpha} g_{1i}h_{1i})/(m_{11} - n_{11})$. Hence A^{-1} can be computed using $2\alpha + 1$ operations in $\mathbb{K}$, and the cost for $n = 1$ is $C(\alpha, 1) := 4\alpha + 2$. Consider now the case $n \ge 2$. Using Lemma 3 together with (9), we see that the matrices A_{11}^{-1}, A_{12}, A_{21}, and their transposes are all of the Cauchy-like structure defined in (19a). Furthermore, for each of them a generator of length at most α can deduced in time $O(\alpha\, n)$ from the quantities computed by Algorithm GenInvLT. Consequently, one can compute $\mathsf{A}_{21}\mathsf{Y}_{11}$, $\mathsf{A}_{12}^T\mathsf{Z}_{11}$, $\mathsf{A}_{11}^{-1}(\mathsf{A}_{12}\mathsf{Y_S})$, and $\mathsf{A}_{11}^{-T}(\mathsf{A}_{21}^T\mathsf{Z_S})$ via six applications, in dimension $n/2$, of the reconstruction formula (19b) to α vectors in $\mathbb{K}^{n/2}$. Finally, Algorithm GenInvLT uses $2\,\alpha\, n$ additions to deduce $\mathsf{G_S}$, $\mathsf{H_S}$, and the upper parts of Y and Z. Overall, the cost for $n \ge 2$ thus satisfies

$$C(\alpha, n) \le 2\, C(\alpha, n/2) + 6\,\mathsf{MM_C}(\alpha, n/2) + k\,\alpha\, n$$

for some constant k. The superlinearity of $\mathsf{MM_C}(\cdot, n)$ then yields our first bound.

Assume now that the x_i and y_i are $2n$ pairwise distinct values. From Lemma 3 the reconstruction formula (19b) can then be applied directly to $\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$ and to the transpose of $\mathsf{A}_{21}\mathsf{A}_{11}^{-1}$, in order to compute $(\mathsf{A}_{11}^{-1}\mathsf{A}_{12})\mathsf{Y_S}$ and $(\mathsf{A}_{21}\mathsf{A}_{11}^{-1})^T\mathsf{Z_S}$. This reduces the number of reconstructions from six to four, whence the second cost bound. $\square$

4.2 Application to Vandermonde-like matrices

Let us now focus on the cost of Algorithm GenInvLT when M and N correspond to the Vandermonde-like structure (20a). We assume x to be partitioned as in the previous section.

LEMMA 4. *Let the matrices $\mathsf{A}, \mathsf{G}, \mathsf{H}, \mathsf{Y}_{11}, \mathsf{Z}_{11}, \mathsf{G_S}, \mathsf{H_S}$ be as in Algorithm* GenInvLT. *Let also w_{11} be the last column of A_{11} and v_{12}^T be the first row of $\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$. Then*

- $\nabla[\mathbb{D}(\mathsf{x}_1), \mathbb{Z}_{n_2,0}^T](\mathsf{A}_{12}) = \mathsf{G}_1\mathsf{H}_2^T + \mathsf{w}_{11}\,\mathsf{e}_{n_2,1}^T,$
- $\nabla[\mathbb{D}(\mathsf{x}_2), \mathbb{Z}_{n_1,0}^T](\mathsf{A}_{21}) = \mathsf{G}_2\mathsf{H}_1^T,$
- $\nabla[\mathbb{Z}_{n_1,0}^T, \mathbb{D}(\mathsf{x}_1)](\mathsf{A}_{11}^{-1}) = \mathsf{Y}_{11}\mathsf{Z}_{11}^T,$
- $\nabla[\mathbb{D}(\mathsf{x}_2), \mathbb{D}(\mathsf{x}_1)](\mathsf{A}_{21}\mathsf{A}_{11}^{-1}) = \mathsf{G_S}\mathsf{Z}_{11}^T,$
- $\nabla[\mathbb{Z}_{n_1,1}^T, \mathbb{Z}_{n_2,0}^T](\mathsf{A}_{11}^{-1}\mathsf{A}_{12}) = -\mathsf{Y}_{11}\mathsf{H_S}^T + \mathsf{e}_{n_1,n_1}(\mathsf{e}_{n_2,1} + \mathsf{v}_{12})^T.$

PROOF. In this case, the upper-right block of N satisfies $\mathsf{N}_{12} = \mathsf{e}_{n_1,n_1}\mathsf{e}_{n_2,1}^T$. Hence we deduce from (2) that $\nabla[\mathbb{D}(\mathsf{x}_1), \mathbb{Z}_{n_2,0}^T](\mathsf{A}_{12}) = \mathsf{G}_1\mathsf{H}_2^T + \mathsf{A}_{11}\mathsf{e}_{n_1,n_1}\mathsf{e}_{n_2,1}^T$ and the first identity follows from the definition of vector w_{11}. The second to fourth identities are obtained in the same way as in the proof of Lemma 3. Let us now verify the last identity, which displays the structure of the product $\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$. First, applying the techniques of Lemma 3, we deduce that

$$\nabla[\mathbb{Z}_{n_1,0}^T, \mathbb{Z}_{n_2,0}^T](\mathsf{A}_{11}^{-1}\mathsf{A}_{12}) = -\mathsf{Y}_{11}\mathsf{H_S}^T + \mathsf{e}_{n_1,n_1}\mathsf{e}_{n_2,1}^T.$$

Then, using (26) with $(\varphi, n) = (1, n_1)$ together with the definition of v_{12} yields the announced expression. $\square$

THEOREM 5. *Let n be a power of two and $\mathsf{M}, \mathsf{N} \in \mathbb{K}^{n\times n}$ be as in (20a). Then Algorithm* GenInvLT *requires at most*

$$3\log(n)\,\mathsf{MM_V}(\alpha, n) + O(\alpha\,\mathsf{M}(n)\log^2(n))$$

field operations. If, in addition, the set $\{x_1,\dots,x_n\}$ has cardinality n then this bound drops to

$$2\log(n)\,\mathsf{MM_V}(\alpha, n) + O(\alpha\,\mathsf{M}(n)\log^2(n)).$$

PROOF. When $n = 1$, $\mathsf{A}^{-1} = x_1/(\sum_{i=1}^{\alpha} g_{1i}h_{1i})$, so that the cost is $C(\alpha, 1) := 4\alpha + 1$. Assume now that $n \ge 2$. Lemma 4 implies that A_{12}, A_{21}, and A_{11}^{-T} share the same Vandermonde-like structure (20a) as A and A_{11}. However, A_{12} has displacement rank bounded by $\alpha + 1$ and computing its generator can be done at cost $O(\alpha\,\mathsf{M}(n)\log(n))$ by applying (20b) to A_{11}. Hence, for $n \ge 2$,

$$\begin{aligned} C(\alpha, n) \;\le\; & 2\,C(\alpha, n/2) + 4\,\mathsf{MM_V}(\alpha, n/2) \\ & + 2\mathsf{MM_V}(\alpha + 1, n/2, \alpha) + k\,\alpha\,\mathsf{M}(n)\log(n), \end{aligned}$$

for some constant k. From (22a) and the superlinearity of $\mathsf{M}(n)$ and $\mathsf{MM_V}(., n)$, we then deduce the first cost bound.

If all the x_i are distinct then, for $\mathsf{A}_{21}\mathsf{A}_{11}^{-1}$, we proceed as for the Cauchy-like case. For $\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$, note that $\mathbb{J}_{n_1}\mathsf{A}_{11}^{-1}\mathsf{A}_{12}$ is Hankel-like in the sense of (21a). Hence, one may first generate the latter matrix in time $O(\alpha\,\mathsf{M}(n)\log(n))$ by obtaining the vector v_{12} after two applications of (20b), then multiply by $\mathsf{Y_S}$ using (21b), and re-apply a reflexion. Thus,

$$\begin{aligned} C(\alpha, n) \;\le\; & 2\,C(\alpha, n/2) + \mathsf{MM_V}(\alpha, n/2) + \mathsf{MM_V}(\alpha + 1, n/2, \alpha) \\ & + \mathsf{MM_C}(\alpha, n/2) + \mathsf{MM_H}(\alpha + 1, n/2, \alpha) \\ & + k\,\alpha\,\mathsf{M}(n)\log(n), \end{aligned}$$

for some constant k, and the conclusion follows as before. $\square$

Note that unlike for the Cauchy-like case, if α is small enough then in the cost bounds of Theorem 5 both summands have the same order of magnitude.

4.3 Extension to Hankel-like matrices

Finally, let us consider the Hankel-like structure defined by $\mathsf{M} = \mathbb{Z}_{n,0}$ and $\mathsf{N} = \mathbb{Z}_{n,0}^T$. Although M and N^T are lower triangular, Algorithm GenInvLT cannot be used directly in this case as the operator $\nabla[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^T]$ is not invertible. Covering such a structure, however, is interesting in particular as it yields an immediate extension to some Toeplitz-like matrices (see [19, Remark 5.4.4] and our Section 3.1).

To cope with the singularity of the displacement operator, some additional data, called *irregularity set* in [19, p. 136], are needed, which typically consist in “a few” entries of A. An irregularity set for $\nabla[\mathbb{Z}_{n,0}, \mathbb{Z}_{m,0}^T]$ is given by the last row of A. Indeed, for $\mathsf{u}^T = \mathsf{e}_{n,n}^T\mathsf{A}$ we see that (2) and (26) imply

$$\nabla[\mathbb{Z}_{n,1}, \mathbb{Z}_{n,0}^T](\mathsf{A}) = [\mathsf{G} \,|\, \mathsf{e}_{n,1}]\,[\mathsf{H} \,|\, \mathsf{u}]^T, \tag{31}$$

so that the matrix A is Hankel-like in the sense of (21a), with displacement rank $\alpha + 1$. Consequently, the reconstruction formula (21b) can be used.

We need to exhibit an irregularity set for $\nabla[\mathbb{Z}_{n,0}^T, \mathbb{Z}_{n,0}]$ too, because we shall multiply with inverses of Hankel-like matrices. A suitable choice here is $\mathsf{v}^T = \mathsf{e}_{n,1}^T\mathsf{A}^{-1}$, the first row of the inverse of A: indeed, if $\nabla[\mathbb{Z}_{n,0}^T, \mathbb{Z}_{n,0}](\mathsf{A}^{-1}) = \mathsf{Y}\mathsf{Z}^T$ then, recalling (13c), we may check that $\mathbb{J}_n\mathsf{A}^{-1}\mathbb{J}_n$ satisfies an identity similar to (31); it is thus fully determined by, up to reflexions, Y, Z, and its last row $\mathsf{v}^T\mathbb{J}_n$.

The resulting adaptation of Algorithm GenInvLT to Hankel-like operator $\nabla[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^T]$ is as follows:

GenInvHL(G, H, u)

Input: $G, H \in \mathbb{K}^{n\times\alpha}$ such that $\nabla[\mathbb{Z}_{n,0}, \mathbb{Z}_{n,0}^T](A) = GH^T$, and $u = A^T e_{n,n}$ (the last row of A).
Assumption: A strongly regular.
Output: $Y = -A^{-1}G$, $Z = A^{-T}H$, and $v = A^{-T}e_{n,1}$ (the first row of A^{-1}).

```
if n = 1 then
    Y := −u⁻¹G; Z := u⁻¹H; v := u⁻¹;
else
    n1 := ⌈n/2⌉; n2 := ⌊n/2⌋;
    [u11; u12] := Aᵀ e_{n,n1}; [u21; u22] := u;
    (Y11, Z11, v11) := GenInvHL(G1, H1, u11);
    GS := G2 + A21 Y11; HS := H2 − A12ᵀ Z11;
    uS := u22 − A12ᵀ A11⁻ᵀ u21;
    (YS, ZS, vS) := GenInvHL(GS, HS, uS);
    Y := [Y11 − A11⁻¹ A12 YS; YS]; Z := [Z11 − A11⁻ᵀ A21ᵀ ZS; ZS];
    w := −S⁻ᵀ A12ᵀ v11; v := [v11 − A11⁻ᵀ A21ᵀ w; w];
fi;
return (Y, Z, v).
```

Theorem 6. *Algorithm* GenInvHL *is correct.*

Proof. When $n = 1$, both A and v are reduced to the scalar $a_{1,1}$ and correctness is then straightforward. Assume now that $n > 1$ and, in order to proceed by induction, assume correctness for $n' < n$. The vector u is split into $u_{21} \in \mathbb{K}^{n_1}$ and $u_{22} \in \mathbb{K}^{n_2}$. Similarly, the vector of coefficients of row n_1 of A is split into $u_{11} \in \mathbb{K}^{n_1}$ and $u_{12} \in \mathbb{K}^{n_2}$. Hence u_{11} equals $A_{11}^T e_{n_1,n_1}$ (that is, the vector of coefficients of the last row of A_{11}), $u_{22} = A_{22}^T e_{n_2,n_2}$, and $u_{21} = A_{21}^T e_{n_2,n_2}$. Recalling that $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$, we deduce that the vector u_S computed by Algorithm GenInvHL satisfies $u_S = S^T e_{n_2,n_2}$ and thus is the vector of coefficients of the last row of S. Since the computation of Y and Z is unchanged in comparison to Algorithm GenInvLT, we still have $Y = -A^{-1}G$ and $Z = A^{-T}H$. All that remains is to prove that v is actually the vector of coefficients of the first row of A^{-1}. By induction, v_{11} and v_S correspond to the first rows of A_{11}^{-1} and S^{-1}, respectively. Using the factorization of A^{-1} seen in (8) and letting $w^T = -v_{11}^T A_{12} S^{-1}$, we get:

$$\begin{aligned} e_{n,1}^T A^{-1} &= \begin{bmatrix} e_{n_1,1}^T & -e_{n_2,1}^T A_{11}^{-1} A_{12} \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & \\ & S^{-1} \end{bmatrix} E \\ &= \begin{bmatrix} v_{11}^T & w^T \end{bmatrix} E \\ &= \begin{bmatrix} v_{11}^T - w^T A_{21} A_{11}^{-1} & w^T \end{bmatrix}, \end{aligned}$$

which is exactly the way vector v is computed. □

Lemma 5. *Let* $A, G, H, Y_{11}, Z_{11}, G_S, H_S, u_{11}$ *be as in Algorithm* GenInvHL. *Recall that* u_{11} *is the last row of the matrix* A_{11} *and let* w_{11} *be its last column. Then*

- $\nabla[\mathbb{Z}_{n_1,0}, \mathbb{Z}_{n_2,0}^T](A_{12}) = G_1 H_2^T + w_{11}\, e_{n_2,1}^T$,
- $\nabla[\mathbb{Z}_{n_2,0}, \mathbb{Z}_{n_1,0}^T](A_{21}) = G_2 H_1^T - e_{n_2,1}\, u_{11}^T$,
- $\nabla[\mathbb{Z}_{n_1,0}^T, \mathbb{Z}_{n_1,0}](A_{11}^{-1}) = Y_{11} Z_{11}^T$,
- $\nabla[\mathbb{Z}_{n_1,0}^T, \mathbb{Z}_{n_2,0}^T](A_{11}^{-1}A_{12}) = -Y_{11} H_S^T + e_{n_1,n_1} e_{n_2,1}^T$,
- $\nabla[\mathbb{Z}_{n_2,0}, \mathbb{Z}_{n_1,0}](A_{21}A_{11}^{-1}) = G_S Z_{11}^T - e_{n_2,1}\, e_{n_1,n_1}^T$.

Proof. Proceed as for Lemma 3 and Lemma 4. □

Theorem 7. *Let n be a power of two and* $M, N \in \mathbb{K}^{n\times n}$ *be as in (21a). Then Algorithm* GenInvHL *requires at most*

$$2\log(n)\, \mathsf{MM}_\mathsf{H}(\alpha, n) + O(\alpha\, \mathsf{M}(n)\log(n)).$$

field operations.

Proof. When $n = 1$, u is a scalar and the algorithm has cost $C(\alpha, 1) := 2\alpha + 2$. Assume now $n \geq 2$. Given G, H, and u, one has (31) and thus (21b) yields $[u_{11}^T, u_{12}^T]$ in time $O(\alpha\, \mathsf{M}(n))$. From Lemma 5, all the blocks involved have the same structure as A, up to transposition and row/column reflexion, and with sometimes a displacement rank $\alpha+1$ instead of α. Generating theses blocks requires the knowledge of the vectors u_{11} (already computed) and w_{11} (computable as u_{11}), which has cost $O(\alpha\, \mathsf{M}(n))$. Now, one may check that the irregularity sets of A_{12}, A_{21}, $\mathbb{J}_n A_{11}^{-1} \mathbb{J}_{n_1}$, $\mathbb{J}_{n_1} A_{11}^{-1} A_{12}$, $A_{21}A_{11}^{-1}\mathbb{J}_{n_1}$, $\mathbb{J}_{n_2} S^{-1} \mathbb{J}_{n_2}$ are, respectively, u_{12}, u_{21}, v_{11}, $v_{11}^T A_{12}$, $u_{21} A_{11}^{-1} \mathbb{J}_{n_1}$, v_S. The vector u_{12} has already been computed, u_{21} is part of the input, v_{11} and v_S are computed recursively, and the two remaining vectors can be recovered in time $O(\alpha\, \mathsf{M}(n))$ from u_{21} and the generators of A_{11}^{-1} and A_{12}. Consequently, all the products that appear in Algorithm GenInvHL can be produced by applications of (21b). Finally, Algorithm GenInvHL still uses $O(\alpha\, n)$ additions, so that the total cost bound is given by

$$C(\alpha, n) \;\leq\; 2\,C(\alpha, n/2) + 4\,\mathsf{MM}_\mathsf{H}(\alpha+1, n/2, \alpha) + k\,\alpha\,\mathsf{M}(n),$$

for some constant k. The conclusion follows from (22b) and the superlinearity assumptions. □

5. EXPERIMENTAL RESULTS

We have implemented the two variants of GenInvLT (with and without Cardinal's trick) as well as the MBA algorithm for Sylvester's displacement. Moreover, we have developed some code to handle Cauchy-like and Hankel-like structures.

For our experiments, we take $\mathbb{K} = \mathbb{F}_p$ with $p = 999999937$, which lets us measure the algebraic costs. Basic operations in $\mathbb{K}$ are provided by NTL,[1] and we also use some code for fast polynomial arithmetic.[2] All the computations are carried out on a desktop machine with an Intel®Core™ 2 Duo processor at 2.66 GHz. Finally, generators (G, H) are picked randomly, while operator matrices $\mathbb{D}(x)$, $\mathbb{D}(y)$ are chosen in order to satisfy all the assumptions made on the algorithms.

Figure 1 shows computing times for inverting Cauchy-like matrices of displacement rank $\alpha = 10$ when n is increasing. It appears that the computing time is quasi-linear with respect to n for each method, and that the compression steps in MBA have negligible cost. Thus, the main difference explaining the various performances lies in the number of products "Cauchy-like matrix × vectors." We have already seen in Theorem 4 that the choice in the parenthesizations leads to one variant in $3\log(n)\,\mathsf{MM}_\mathsf{C}(\alpha, n)$ and, up to stronger conditions on the input, to another variant in $2\log(n)\,\mathsf{MM}_\mathsf{C}(\alpha, n)$. Let us now estimate this cost for our implementation of the MBA algorithm. Generators for the Schur complement and the inverse of A before the compression steps are computed using (10) according to the following parenthesization: $X_1 = A_{11}^{-1}A_{12}$, $S = A_{22} - A_{21}X_1$,

[1] http://www.shoup.net/ntl/
[2] http://www.math.uvsq.fr/~lecerf/software/tellegen/

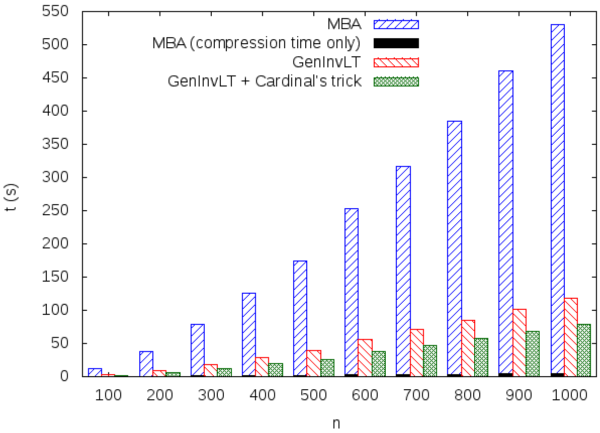

Figure 1: Cost (in seconds) of Cauchy-like matrix inversion for increasing values of n and $\alpha = 10$.

$X_2 = A_{21}A_{11}^{-1}$, and

$$A^{-1} = \begin{bmatrix} A_{11}^{-1}+(X_1S^{-1})X_2 & -X_1S^{-1} \\ -S^{-1}X_2 & S^{-1} \end{bmatrix}.$$

Counting the costs of all these products using (23) and the superlinearity of $\mathsf{MM_C}(\cdot, n)$ leads to a bound of $14\,\mathsf{MM_C}(\alpha, n)$ in the recurrence equation for the cost of MBA, which gives a total cost dominated by $14 \log(n)\,\mathsf{MM_C}(\alpha, n)$. In Figure 1, we observe a speed-up around $4.6 \approx 14/3$ between MBA and our first variant (GenInvLT), and around $6.7 \approx 14/2$ between MBA and the second variant (GenInvLT + Cardinal's trick), which is in agreement with our analysis above.

Moreover, we experimented with Hankel-like matrices in order to estimate the cost of row reconstruction and the additional time due to subblocks having displacement rank $\alpha + 1$ instead of α like in the Cauchy-like case. Timings are summarized in Table 1, where it appears that these costs become negligible when α is large enough. Indeed, they are linear in α as expected (see (22b) and Theorem 5) while the total cost seems quadratic in α.

α	10	30	50	70	90
Total cost	4.7	34.7	92.1	177.0	290.3
Irregularity related cost	0.8	2.5	3.9	5.3	7.1
Rank increase related cost	0.5	1.6	2.6	3.5	4.6

Table 1: Cost (in seconds) of Hankel-like matrix inversion for $n = 200$ and increasing values of α.

6. CONCLUSIONS

In this paper, we have extended Cardinal's compression-free algorithm to a broader class of structured matrices, including not only the Cauchy-like type but also the Vandermonde, Hankel, and Toeplitz-like types. Our main conclusion is that this approach yields variants of the MBA algorithm that are simpler to analyze and implement, and, according to our first experiments, significantly faster in practice. However, this study calls for a number of extensions:

On the practical side, we should first study the impact of stopping recursive calls (and reconstructing A^{-1} explicitly via fast dense linear algebra) when $n \approx \alpha$. It would also be interesting to measure the memory gains brought by Cardinal's extended approach over MBA.

On the algorithmic side, although we have focused only on $O\tilde{~}(\alpha^2 n)$ versions of MBA, it would be interesting to incorporate the matrix multiplication techniques of [3]. We should also study the impact of multiplicities in x and y on the cost bounds and adapt our work to structures like those of the Toeplitz+Hankel-like type.

7. ACKNOWLEDGMENTS

We thank Benoît Lacelle for providing a framework for structured matrices and the generator-compression subroutine used in our implementation of MBA, and Éric Schost for pointing out the code used for multipoint evaluation.

8. REFERENCES

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

[2] A. Bostan, C.-P. Jeannerod, and É. Schost. Solving Toeplitz- and Vandermonde-like linear systems with large displacement rank. In *ISSAC '07*, pages 33–40. ACM, 2007.

[3] A. Bostan, C.-P. Jeannerod, and É. Schost. Solving structured linear systems with large displacement rank. *Theoretical Computer Science*, 407(1:3):155–181, 2008.

[4] J.-P. Cardinal. On a property of Cauchy-like matrices. *C. R. Acad. Sci. Paris - Série I - Analyse numérique/Numerical Analysis*, 328:1089–1093, 1999.

[5] J.-P. Cardinal. A divide and conquer method to solve Cauchy-like systems. Technical report, The FRISCO Consortium, 2000.

[6] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, second edition, 2003.

[7] I. Gohberg and V. Olshevsky. Complexity of multiplication with vectors for structured matrices. *Linear Algebra Appl.*, 202:163–192, 1994.

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

[9] G. Heinig. Inversion of generalized Cauchy matrices and other classes of structured matrices. *Linear Algebra for Signal Processing*, 69:95–114, 1995.

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

[11] E. Kaltofen. Asymptotically fast solution of Toeplitz-like singular linear systems. In *ISSAC'94*, pages 297–304. ACM, 1994.

[12] E. Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, 1995.

[13] M. Morf. Doubling algorithms for Toeplitz and related equations. *IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 954–959, 1980.

[14] V. Olshevsky and V. Pan. A unified superfast algorithm for boundary rational tangential interpolation problems and for inversion and factorization of dense structured matrices. In *Proc. 39th IEEE FOCS*, pages 192–201, 1998.

[15] V. Pan. Parallel solution of Toeplitz-like linear systems. *Journal of Complexity*, 8(1):1–21, 1992.

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

[17] V. Y. Pan. Parametrization of Newton's iteration for computations with structured matrices and applications. *Computers Math. Applic.*, 24(3):61–75, 1992.

[18] V. Y. Pan. Nearly optimal computations with structured matrices. In *SODA*, pages 953–962, 2000.

[19] V. Y. Pan. *Structured Matrices and Polynomials*. Birkhäuser Boston Inc., 2001.

[20] V. Y. Pan and A. Zheng. Superfast algorithms for Cauchy-like matrix computations and extensions. *Linear Algebra Appl.*, 310:83–108, 2000.

APPENDIX

A. RECONSTRUCTION OF RECTANGULAR HANKEL-LIKE MATRICES

Theorem 8. *Equation (21b) is correct.*

Proof. The Hankel-like matrix $\mathsf{A} \in \mathbb{K}^{n\times m}$ satisfies

$$\mathbb{Z}_{n,1}\mathsf{A} - \mathsf{A}\mathbb{Z}_{m,0}^T = \mathsf{G}\mathsf{H}^T.$$

By left-multiplying with $\mathbb{Z}_{n,1}^T$, we deduce the following recurrence formula:

$$\mathsf{A} = \mathbb{Z}_{n,1}^T\mathsf{A}\mathbb{Z}_{m,0}^T + \mathbb{Z}_{n,1}^T\mathsf{G}\mathsf{H}^T.$$

We can now apply Theorem 4.3.6 from [19] to obtain that, for all integer k,

$$\mathsf{A} = (\mathbb{Z}_{n,1}^T)^k\mathsf{A}(\mathbb{Z}_{m,0}^T)^k + \sum_{l=0}^{k-1}(\mathbb{Z}_{n,1}^T)^{l+1}\mathsf{G}\mathsf{H}^T(\mathbb{Z}_{m,0}^T)^l.$$

As $(\mathbb{Z}_{m,0}^T)^l$ is the null matrix as soon as $l \geq m$, we can simplify the above equation:

$$\begin{aligned}
\mathsf{A} &= \sum_{l=0}^{m-1}(\mathbb{Z}_{n,1}^T)^{l+1}\mathsf{G}\mathsf{H}^T(\mathbb{Z}_{m,0}^T)^l \\
&= \sum_{l=0}^{m-1}(\mathbb{Z}_{n,1}^T)^{l+1}\left(\sum_{j=1}^{\alpha}\mathsf{g}_j\mathsf{h}_j^T\right)(\mathbb{Z}_{m,0}^T)^l \\
&= \sum_{j=1}^{\alpha}\sum_{l=0}^{m-1}(\mathbb{Z}_{n,1}^T)^{l+1}\mathsf{g}_j\mathsf{h}_j^T(\mathbb{Z}_{m,0}^T)^l.
\end{aligned}$$

The inner sum is a sum of m outer products. This can be rewritten as a product of two matrices $B \in \mathbb{K}^{n\times m}$ and $C \in \mathbb{K}^{m\times m}$ by taking $(\mathbb{Z}_{n,1}^T)^{l+1}\mathsf{g}_j$ for column $m-l$ of B and $\mathsf{h}_j^T(\mathbb{Z}_{m,0}^T)^l$ for row $m-l$ of C. Thus, the ith row of C is exactly $\begin{bmatrix} 0 & \cdots & 0 & \mathsf{h}_{j,1} & \cdots & \mathsf{h}_{j,i}\end{bmatrix}$ so that $C = \mathbb{L}(\mathsf{h}_j)\mathbb{J}_m$.

Finally, the last column of B is $\begin{bmatrix} \mathsf{g}_{j,2} & \cdots & \mathsf{g}_{j,n} & \mathsf{g}_{j,1}\end{bmatrix}^T$ and we can deduce its column l by applying $\mathbb{Z}_{n,1}^T$ to its column $l+1$, that is by moving the elements of column $l+1$ one location up (except for the first element which goes at bottom). This defines a Toeplitz matrix whose coefficient (k,l) is $\mathsf{g}_{j,1+(k-l+m) \bmod n}$, which is $\mathbb{T}^{n\times m}(\mathsf{g}_j)$ by definition. $\square$