A MODIFIED RECURSIVE TRIANGULAR FACTORIZATION FOR CAUCHY-LIKE SYSTEMS*

Zhao Chen

Mathematics Department
New York City Technical College
300 Jay Street
Brooklyn, NY 11201
zchen@nyctc.cuny.edu

Abstract   We derive new algorithms for solving strongly nonsingular Cauchy-like systems of linear equations $C\tilde{x} = \tilde{v}$ in $O(n \log^2 n)$ running time, where $\mathbf{F}$ is a field and $\tilde{v} \in \mathbf{F}^{n \times 1}$ is a vector, $C \in \mathbf{F}^{n \times n}$ is a strongly nonsingular Cauchy-like matrix. Morf, Bitmead and Anderson presented the efficient algorithms to solve strongly nonsingular Toeplitz-like equations of linear systems by using the Recursive Triangular Factorization in 1980. Recently, Pan and Zheng extended the Recursive Triangular Factorization to solve Cauchy-like systems with the complexity of $O(n \log^3 n)$ operations. This is the best known complexity bound by using the direct approach of Recursive Triangular Factorization in Cauchy-like cases. However, these algorithms are still slower than the well known algorithms with the asymptotic bound of $O(n \log^2 n)$ operations, which have been proposed by the means of reducing Cauchy-like matrices into Toeplitz-like matrices. In our present paper, we will modify the Recursive Triangular Factorization so that the complexity bound of the direct recursive approach can be decreased to $O(n \log^2 n)$ operations. This matches the asymptotic bound without transforming to Toeplitz-like matrices. Our improvement of the direct recursive approach is by a factor off $\log n$ due to changing the original vectors which expressed in the given Cauchy-like matrix into the special vectors, where the entries are unit roots. The applications of structured matrices include Nevanlinna-Pick tangential interpolation problems.

1. Introduction

Computations with displacement structured matrices such as Cauchy (or-like), Vandermonde (or-like) and Toeplitz (or-like) have a variety of applications to science and engineering, for example, conformal mapping, tangential Nevanlinna-Pick interpolation, solution of integral equations, rational interpolation, polynomial interpolation and evaluation (cf. [25], [24], [23], [4], [19], [22], [18], [14], [12], [16] and [15]).

By exploiting the displacement structure, we can save running time and memory spaces dramatically in solving Cauchy-like systems of linear equations. Instead of working on the entries of a matrix itself, we use its generators in our computations. Many matrices can be re-constructed through their generators with the displacement operators [11], [1], [10] and [9]. The displacement representation provides many opportunities to accelerate our computations.

Let $\mathbf{F}$ be a field and $\tilde{s}$ denotes a vector. We also denote $A_{M\tilde{v}}(n)$ as the complexity of the product of a matrix $M \in \mathbf{F}^{n \times n}$ by a vector $\tilde{v} \in \mathbf{F}^{n \times 1}$. Moreover, Let us work over a field $\mathbf{F}$ which supports Fast Fourier Transforms of length n.

To solve $n \times n$ strongly nonsingular Toeplitz-like equations of linear systems $T\tilde{x} = \tilde{v}$, where $T \in \mathbf{F}^{n \times n}$ is a Toeplitz-like matrix, Morf, Bitmead and Anderson (see e.g. [13] and [2]) applied the Recursive Triangular Factorization to design fast algorithms with complexity of $O(n \log^2 n)$ arithmetic operations (hereafter, we refer to arithmetic operations as ao). These algorithms are faster than Gaussian Elimination algorithms, which run $O(n^3)$ times.

The efficient techniques of the Recursive Triangular Factorization have been extended recently to $n \times n$ Cauchy-like matrices by Pan and Zheng [20] with complexity of

$$A_{C_{like}^{-1}\tilde{v}}(n) = O(A_{C\tilde{v}}(n) \log n) = O(n \log^3 n), \qquad (1.1)$$

where C_{like}^{-1} denotes as the inverse of a Cauchy-like matrix and $A_{C\tilde{v}}(n) = O(n \log^2 n)$ denotes as the complexity of multiplication of an $n \times n$ Cauchy matrix by a vector. $O(n \log^3 n)$ is the best known bound by using direct recursive approach to solve Cauchy-like systems. However, these algorithms are still slower than the well known algorithms with the asymptotic bound of $O(n \log^2 n)$ operations, which have been proposed by the means of reducing Cauchy-like matrices into Toeplitz-like matrices. Due to the difference structure between the Cauchy-like matrix and Toeplitz-like matrix, transitions from Cauchy-like matrices into Toeplitz-like matrices will lose the Cauchy-like extension of the Recursive Triangular Factorization from the Toeplitz-like case. Moreover, the

transitions which used Vandermonde matrices may produce errors in numerical computations because of the ill-condition of the Vandermonde matrices [5] (except the Fourier matrices). In our present paper, we will modify the Recursive Triangular Factorization so that the complexity bound of the direct recursive approach can be decreased to $O(n \log^2 n)$ operations. This matches the asymptotic bound without transforming to Toeplitz-like matrices.

We propose the new Cauchy-like extension of the Recursive Triangular Factorization, which is inferior to the best known direct recursive approach.

By observing from (1.1), the complexity of the Recursive Triangular Factorization depends on the complexity of the products of a Cauchy matrix by a vector. Let $\left(\frac{1}{e_i - f_j}\right)_{i,j=0}^{n-1}$ be a Cauchy matrix. It is well known that a Cauchy matrix times a vector can be computed in $O(n \log^2 n)$ *ao* (see [8]). If the entries of the vectors $\tilde{e} = e_0, \cdots, e_{n-1}$ and $\tilde{f} = f_0, \cdots, f_{n-1}$ are n-th roots of 1 and -1, it costs only $O(n \log n)$ *ao* to compute their product. This property motivates us to change the original vectors which expressed in generalized Cauchy-like matrices into the special vectors, where the entries the vectors are unit roots.

Due to the new matrix preserves the same Cauchy-like structure, the computations of finding the inverse of the new matrix is faster than the computations of finding the inverse of the original given matrix by using the Recursive Triangular Factorization. This new process of modifying the Recursive Triangular Factorization can be used to solve strongly nonsingular Cauchy-like systems of linear equations in the complexity of

$$A_{C_{like}^{-1}\tilde{v}}(n) = O(A_{C_s\tilde{v}}(n) \log n) = O(n \log^2 n), \qquad (1.2)$$

where $A_{C_s\tilde{v}}(n) = O(n \log n)$ denotes the complexity of multiplication of an $n \times n$ special Cauchy matrix by a vector. (we refer a Cauchy matrix $\left(\frac{1}{e_i - f_j}\right)_{i,j=0}^{n-1}$ as a special Cauchy matrix if the entries of the vectors $\tilde{e}, \tilde{f}$ are roots of unity). Our improvement for the direct recursive approach is by a factor off $\log n$, versus running time of the best known algorithms for the direct recursive approach. In fact, Heinig [10] has proposed to change the vectors into unit roots by applying Vandermonde linear solver. Computations with Vandermonde matrices will create the numerical instabilities due to the ill-conditions of Vandermonde matrices (cf. [5]). Our algorithms avoid the computations of the solutions of Vandermonde linear systems. The complexity bound of our algorithms reaches to the asymptotic bound.

We organize our paper as follow. In section 2, we present our new representation of the inverse of Cauchy-like matrices. In section 3, we recall the basic algorithms of the Recursive Triangular Factorization. In our final section, we modify the Recursive Triangular Factorization to solve Cauchy-like linear systems of equations. Let us work over a field F which supports FFT and follow some lines of [20] and [21].

2. New Expression of the Inverse of a Cauchy-like matrix

It is fast for us to work with the Cauchy-like matrices with special vectors because of lowering our computation complexity. We will show that the vectors in Cauchy-like matrices can be reduced to special vectors. The new representation of Cauchy-like matrices will be given in this section. Let us review some well known definitions.

Definition 1 *Let* $\alpha = e^{-\frac{2\Pi\sqrt{-1}}{n}}$, $\beta = e^{\frac{\Pi\sqrt{-1}}{n}}$. *The elements* $\alpha_i = \alpha^i$, $\beta_j = \beta^j$, *for* $i, j = 0, \ldots, n-1$ *are* n-*roots of* 1 *and* -1 *respectively.*

Definition 2 *(cf.[17]) We denote* W^T *as the transpose of a matrix or a vector* W. *The inverse of a matrix* M *is denoted as* M^{-1}. *Let* $D(\tilde{x}) \in \mathbf{F}^{n \times n}$ *be a diagonal matrix with the diagonal entries* $\tilde{x} = x_0, \ldots, x_{n-1}$. *Let* $c(\tilde{t}, \tilde{v}) = \left(\frac{1}{t_i - v_j}\right)_{i,j=0}^{n-1} \in \mathbf{F}^{n \times n}$ *denoted as a Cauchy matrix.*

Let us recall some well known results as follows: (cf. [8])

Lemma 1 *Let* $\tilde{w}$, $\tilde{u}$ *and* $\tilde{v}$ *be triple vectors, where the elements* $u_i \in \tilde{u}$ *and* $w_j \in \tilde{w}$, *are not equal in a field* $\mathbf{F}$, *for* $i, j = 0, 1, \ldots n-1$. *Let* $C(\tilde{u}, \tilde{w})$ *be an* $n \times n$ *Cauchy matrix. Then the computations of the product* $\tilde{x} = C(\tilde{u}, \tilde{w})\tilde{v}$ *cost*

$$A_{C\tilde{v}}(n) = O(n \log^2 n) \tag{2.1}$$

ao. *If the elements of the vectors* $\tilde{u}$ *and* $\tilde{w}$ *are roots of unity, then the complexity can be reduced to*

$$A_{C_s\tilde{v}}(n) = O(n \log n) \tag{2.2}$$

ao.

By following [6], and [9], we have the following definitions.

Definition 3 *A linear operator* $\Delta^{[E,F]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$ *is defined to map each matrix* $C \in \mathbf{F}^{n \times n}$ *to its displacement* $EC - CF$, *where* $E \in \mathbf{F}^{n \times n}, F \in \mathbf{F}^{n \times n}$ *are given matrices. The operator* $\Delta^{[E,F]}(\cdot)$ *is*

called the symmetric Sylvester operator. The rank of the image, $r =$ rank$(EC - CF)$ *is called $[E, F]$-displacement rank of the matrix C for $r \ll n$. Let $E = D(\tilde{a}) \in \mathbf{F}^{n \times n}$, $F = D(\tilde{b}) \in \mathbf{F}^{n \times n}$ and $G_r, H_r \in \mathbf{F}^{n \times r}$ be matrices. A matrix $C \in \mathbf{F}^{n \times n}$ is called a Cauchy-like matrix if it satisfies*

$$\Delta^{[D(\tilde{a}), D(\tilde{b})]}(C) = D(\tilde{a})C - CD(\tilde{b}) = G_r H_r^T, \qquad (2.3)$$

where the pair of the matrices G_r, $H_r \in \mathbf{F}^{n \times r}$ are called the $\Delta^{[D(\tilde{a}), D(\tilde{b})]}$-generators of C with length r. We denote this matrix as $C_{G_r H_r}(\tilde{a}, \tilde{b})$ or C_{like}.

It is easy to realize the following well known relationships of the Cauchy and Cauchy-like matrices (cf. [9]). We observe that $\Delta^{[D(\tilde{s}), D(\tilde{u})]}$ $C(\tilde{s}, \tilde{u}) = D(\tilde{s})C(\tilde{s}, \tilde{u}) - C(\tilde{s}, \tilde{u})D(\tilde{u}) = \tilde{1}\tilde{1}^T$, where $\tilde{1}^T = (1, \dots, 1)$. Thus it follows.

Lemma 2 *let $c(\tilde{s}, \tilde{u})$ be an $n \times n$ Cauchy matrix with the vectors $\tilde{s}$ and $\tilde{u}$ having $2n$ distinct values in $\mathbf{F}$, then it has the $[D(\tilde{s}), D(\tilde{u})]$-displacement rank equal to 1.*

Let the Sylvester operator, $\Delta^{[E, F]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$, act on the linear space $\mathbf{F}^{n \times n}$ with the empty kernel. The Cauchy-like matrices can be completely reconstructed as follows:

Proposition 1 *([7]) Let $C_{G_k H_k}(\tilde{v}, \tilde{w}) \in \mathbf{F}^{n \times n}$ be a Cauchy-like matrix associated with the Sylvester operator $\Delta^{[D(\tilde{v}), D(\tilde{w})]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$ and generators $G_k = [\tilde{g}_1, \dots, \tilde{g}_k]$, $H_k = [\tilde{h}_1, \dots, \tilde{h}_k]$ as defined in Definition 3. Then the Cauchy-like matrix $C_{G_k H_k}(\tilde{v}, \tilde{w})$ can be expressed as*

$$C_{G_k H_k}(\tilde{v}, \tilde{w}) = \sum_{i=1}^{k} D(\tilde{g}_i)C(\tilde{v}, \tilde{w})D(\tilde{h}_i) = \left(\frac{\tilde{g}_i^T \tilde{h}_j}{v_i - w_j} \right)_{i,j=0}^{n-1}, \qquad (2.4)$$

where $C(\tilde{v}, \tilde{w})$ is a Cauchy matrix and $k \ll n$.

By Combining lemma 1 and proposition 1, we obtain the following results:

Corollary 1 *Let $\tilde{v} \in \mathbf{F}^{n \times 1}$ be a vector and $C_{G_r R_r}(\tilde{d}, \tilde{p}) \in \mathbf{F}^{n \times n}$ be a Cauchy-like matrix. Then the product $\tilde{y} = C_{G_r H_r}(\tilde{d}, \tilde{p})\tilde{v}$ can be computed in*

$$A_{C_{like}\tilde{v}}(n) = O(rA_{C\tilde{v}}(n)) = O(rn \log^2 n) \qquad (2.5)$$

ao. Furthermore, if the elements of the vectors $\tilde{d}$ and $\tilde{p}$ are roots of unity, then the complexity can be accelerated to

$$A_{C_{like}\tilde{v}}(n) = O(rA_{C_s\tilde{v}}(n)) = O(rn \log n) \qquad (2.6)$$

ao.

Theorem 1 *Let $\Delta^{[E,F]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$ be the Sylvester operator. Let $C_{G_a H_a}(\tilde{z}, \tilde{x})$, $C_{G_b H_b}(\tilde{x}, \tilde{y})$ and $C_{G_d H_d}(\tilde{y}, \tilde{w})$ denote the Cauchy-like matrices according to Definition 3 with $\Delta^{[D(\tilde{z}), D(\tilde{x})]}(C_{G_a H_a}(\tilde{z}, \tilde{x})) = G_a H_a^T$, $\Delta^{[D(\tilde{x}), D(\tilde{y})]}(C_{G_b H_b}(\tilde{x}, \tilde{y})) = G_b H_b^T$ and $\Delta^{[D(\tilde{y}), D(\tilde{w})]}(C_{G_d H_d}(\tilde{y}, \tilde{w})) = G_d H_d^T$ respectively, where the matrices $G_a, H_a \in \mathbf{F}^{n \times a}$, G_b, $H_b \in \mathbf{F}^{n \times b}$, G_d, $H_d \in \mathbf{F}^{n \times d}$ are generators repectively, for $a, b, d << n$, and all values of x_i, y_j, z_k, w_m are all distinct in a field $\mathbf{F}$. Then the product matrix*

$$C_{G_s R_s}(\tilde{z}, \tilde{w}) = C_{G_a H_a}(\tilde{z}, \tilde{x}) C_{G_b H_b}(\tilde{x}, \tilde{y}) C_{G_d H_d}(\tilde{y}, \tilde{w}), \qquad (2.7)$$

is a Cauchy-like matrix with the $[D(\tilde{z}), D(\tilde{w})]$-displacement rank s ($s = a + b + d$) such that

$$\Delta^{[D(\tilde{z}), D(\tilde{w})]}(C_{G_s H_s}(\tilde{z}, \tilde{w})) = G_s H_s^T, \qquad (2.8)$$

where the pair of matrices G_s, H_s are $[D(\tilde{z}), D(\tilde{w})]$-generators of the size $n \times s$,

$$G_s = [G_a, C_{G_a H_a}(\tilde{z}, \tilde{x}) G_b, C_{G_a H_a}(\tilde{z}, \tilde{x}) C_{G_b H_b}(\tilde{x}, \tilde{y}) G_d],$$

$$H_s = [C_{G_a H_a}(\tilde{y}, \tilde{w})^T C_{G_b H_b}(\tilde{x}, \tilde{y})^T H_a, C_{G_d H_d}(\tilde{y}, \tilde{w})^T H_b, H_d].$$

Proof. We have $\Delta^{[D(\tilde{z}), D(\tilde{w})]}(C_{G_s H_s}(\tilde{z}, \tilde{w})) = G_a H_a^T C_{G_b H_b}(\tilde{x}, \tilde{y}) C_{G_a H_a}(\tilde{y}, \tilde{w}) + C_{G_a H_a}(\tilde{z}, \tilde{x}) G_b H_b^T C_{G_d H_d}(\tilde{y}, \tilde{w}) + C_{G_a H_a}(\tilde{z}, \tilde{x}) C_{G_b H_b}(\tilde{x}, \tilde{y}) G_d H_d^T = G_s H_s^T$, where $s = a + b + d$ and $G_s = [G_a, C_{G_a H_a}(\tilde{z}, \tilde{x}) G_b, C_{G_a H_a}(\tilde{z}, \tilde{x}) C_{G_b H_b}(\tilde{x}, \tilde{y}) G_d]$, $H_s = [C_{G_a H_a}(\tilde{y}, \tilde{w})^T C_{G_b H_b}(\tilde{x}, \tilde{y})^T H_a, C_{G_d H_d}(\tilde{y}, \tilde{w})^T H_b, H_d]$. Q.E.D.

Theorem 2 *Let $C_{G_r H_r}(\tilde{e}, \tilde{f})$ be an $n \times n$ generalized Cauchy-like matrix associated with the symmetric Sylvester operator $\Delta^{[D(\tilde{e}), D(\tilde{f})]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$ of (2.3) such that $\Delta^{[D(\tilde{e}), D(\tilde{f})]}(C_{G_r H_r}(\tilde{e}, \tilde{f})) = G_r H_r^T$. Let $C(\tilde{p}, \tilde{e}), C(\tilde{f}, \tilde{q})$ be a pair of Cauchy matrices, where e_i, f_j, p_k, q_l are all distinct. Then we have the following matrix equations*

$$C_{G_r H_r}(\tilde{e}, \tilde{f})^{-1} = C(\tilde{f}, \tilde{q}) C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})^{-1} C(\tilde{p}, \tilde{e}), \qquad (2.9)$$

where $C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})$ is a Cauchy-like matrix associated with $[D(\tilde{p}), D(\tilde{q})]$-displacement generators,

$$G_{r+2} = [\tilde{1}, C(\tilde{p}, \tilde{e}) G_r, C(\tilde{p}, \tilde{e}) C_{G_r H_r}(\tilde{e}, \tilde{f}) \tilde{1}],$$

$$H_{r+2} = [C(\tilde{f}, \tilde{q})^T C_{G_r H_r}(\tilde{e}, \tilde{f})^T \tilde{1}, C(\tilde{f}, \tilde{q})^T H_r, \tilde{1}],$$

for $\tilde{1}^T = (1, \ldots, 1)$. Furthermore, the complexity of our computations is $O(nr \log^2 n)$ arithmetic operations.

Proof: Choose the $[D(\tilde{p}), D(\tilde{e})]$, $[D(\tilde{f}), D(\tilde{q})]$-displacement rank a, d of the Cauchy-like matrices $C_{G_a H_a}(\tilde{p}, \tilde{e})$, $C_{G_d H_d}(\tilde{f}, \tilde{q})$ respectively equal to 1; i.e., $a = d = 1$. We now have $C(\tilde{p}, \tilde{e}) = C_{G_a H_a}(\tilde{p}, \tilde{e})$, $C(\tilde{f}, \tilde{q}) = C_{G_d H_d}(\tilde{f}, \tilde{q})$ from the lemma 2. From the theorem 1, it follows that $C(\tilde{p}, \tilde{e}) C_{G_r H_r}(\tilde{e}, \tilde{f}) C(\tilde{f}, \tilde{q}) = C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})$. Substitute this matrix equation into the trivial matrix identity $C_{G_r H_r}(\tilde{e}, \tilde{f}) = C(\tilde{p}, \tilde{e})^{-1} C(\tilde{p}, \tilde{e})$ $C_{G_r H_r}(\tilde{e}, \tilde{f})$ $C(\tilde{f}, \tilde{q})$ $C(\tilde{f}, \tilde{q})^{-1}$ and obtain equations $C_{G_r H_r}(\tilde{e}, \tilde{f}) = C(\tilde{p}, \tilde{e})^{-1} C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q}) C(\tilde{f}, \tilde{q})^{-1}$. By inverting both sides of this equation, we immediately obtain the equation (2.9). The complexity is immediately followed by the corollary 1. *Q.E.D.*

Theorem 3 *(cf. [1]) Given the $[D(\tilde{m}), D(\tilde{n})]$-displacement generators G_u, H_u with displacement rank u of a Cauchy-like matrix $C_{G_u H_u}(\tilde{m}, \tilde{n})$, $x < u < n$, it requires $O(u^2 n)$ operations to compute the $D[(\tilde{m}), (\tilde{n})]$-displacement generators G_x, H_x of the Cauchy matrix $C_{G_x H_x}(\tilde{m}, \tilde{n})$.*

Corollary 2 *Let the matrix $C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})$ be the Cauchy-like matrix as in Theorem 3. Then we use $O(r^2 n)$ to compute the new generator $G_{r'}$ and $H_{r'}$ for the matrix $C_{G_{r'} H_{r'}}(\tilde{p}, \tilde{q})$, where $r' \leq r < r + 2 < n$.*

Proof. The running time of computing the displacement generators $G_{r'}$, $H_{r'}$ is immediately followed by Theorem 3. *Q.E.D.*

3. Recursive Triangular Factorization of a matrix.

In this section, we will recall the known method of the Recursive Triangular Factorization to a strongly nonsingular matrix [20], [21].

Definition 4 *A matrix M is strongly nonsingular if all its leading principal submatrices are nonsingular.*

Let us denote I identity matrix, O denotes a null matrix. Given an $n \times n$ strongly nonsingular matrix M, we partition the matrix M into four $n/2 \times n/2$ blocks, e.g.

$$M = \begin{pmatrix} A & B \\ D & F \end{pmatrix}. \tag{3.1}$$

The matrix S is called the *Schur complement* of A in M and can be computed as

$$S = F - DA^{-1}B. \tag{3.2}$$

we may write its inverse as

$$M^{-1} = \begin{pmatrix} A^{-1} + A^{-1}BS^{-1}DA^{-1} & -A^{-1}BS^{-1} \\ -S^{-1}DA^{-1} & S^{-1} \end{pmatrix}. \qquad (3.3)$$

Let matrices A and S be investable. We can continue this process recursively for the submatrices A and S. Based on the following propositions, we may do the factorization to all leading principal submatrices, $1 \times 1, 2 \times 2, 4 \times 4, \ldots, n \times n$. The total numbers of the steps for the Recursive Triangular Factorization is $\lceil \log_2 n \rceil$. Each step of the computations cost matrices multiplications and subtractions.

Proposition 2 *([20], [21]) If $M \in \mathbf{F}^{n \times n}$ is matrix with leading principal submatrices being nonsingular, so are A and S.*

Proposition 3 *([20], [21]) Suppose M is a matrix with leading principal submatrices being nonsingular and write S as (3.2). Let M_1 be a leading principal submatrix of S and let S_1 denote the Schur complement of M_1 in S. Then S^{-1} and S_1^{-1} are located in the southeastern blocks of M^{-1}.*

Here is the algorithms of the Recursive Triangular Factorization for inversion.
Algorithms 3.1. *Recursive Triangular Factorization for inversion*
Input: an matrix $M \in \mathbf{F}^{n \times n}$.
Output: the inversive matrix M^{-1}.
Computations:
  1. Use Algorithms 3.1 Compute A^{-1} as (3.1).
  2. Compute the Schur complement $S = F - DA^{-1}B$.
  3. Use Algorithms 3.1 to the inverse matrix S^{-1}
  4. Compute M^{-1} from (3.3).

4. Modified Recursive Triangular Factorization to a Cauchy-like matrix

In this section, we will describe the techniques of the modified Recursive Triangular Factorization to a Cauchy-like matrix. As we can see from the equation (1.2), the complexity of the Recursive Factorization depends on the complexity of a Cauchy matrix by a vector. It needs only $O(n \log n)$ *ao* to compute a special Cauchy matrix times a vector. It motivates us to modify the process of the Recursive Factorization so that the complexity bound of [20] can be improved by a factor of $\log n$.
Let us recall some well known results first.

Lemma 3 *([20], [21]) Let $\tilde{b}_j \in \mathbf{F}^{n \times 1}$, $j = 1, 2, 3$, be tree distinct vectors and $M_i \in \mathbf{F}^{n \times n}$ be cauchy-like matrices such that $\Delta^{[D(\tilde{b}_i), D(\tilde{b}_{i+1})]}(M_i) = G_i H_i^T$, where G_i, $H_i \in \mathbf{F}^{n \times r_i}$, $r_i < n$, for $i = 1, 2$. The matrix $M = M_1 M_2$ is a Cauchy-like matrix with $\Delta^{[D(\tilde{b}_1), D(\tilde{b}_3)]}(M) = G_r H_r^T$, $G_r = [G_1, M_1 G_2]$, $H_r = [M_2^T H_1, H_2]$, G_r, $H_r \in \mathbf{F}^{n \times r}$, $r = r_1 + r_2$. Furthermore, $O(r_1 r_2 A_{C\tilde{v}}(n))$ ao suffice to compute G_r and H_r.*

By multiplying $C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1}$ on both sides of the symmetric Sylvester equation for $C_{M_t W_t}(\tilde{x}, \tilde{b})$, it follows:

Lemma 4 *Given an $n \times n$ Cauchy-like matrix $C_{M_t W_t}(\tilde{x}, \tilde{b})$ as defined in definition 3 such that (2.3) holds for $t << n$, the matrix $C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1}$ satisfies $\Delta^{[D(\tilde{b}), D(\tilde{x})]}(C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1}) = G_r H_r^T$, where $G_r = [C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1} M_t]$, $H_r = [(C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1})^T W_t]$.*

Fact 1 *By Lemma 4, the rank $\Delta^{[D(\tilde{b}), D(\tilde{x})]}(C_{M_t W_t}(\tilde{x}, \tilde{b})^{-1}) << t$.*

Lemma 5 *([20], [21]) Let M and N be $n \times n$ matrices which satisfy $\Delta^{[D(\tilde{p}), D(\tilde{q})]}(M) = G_{r_1} H_{r_1}^T$ and $\Delta^{[D(\tilde{p}), D(\tilde{q})]}(N) = G_{r_2} H_{r_2}^T$ respectively. Then the matrices $M + N$ and $M - N$ are Cauchy-like matrices associated with a $[D(\tilde{p}), D(\tilde{q})]$ -generator of length at most $r_2 + r_1$.*

Lemma 6 *([20], [21]) Let $C_{G_{r'} H_{r'}}(\tilde{p}, \tilde{q})$ be an $n \times n$ generalized Cauchy-like matrix associated with the symmetric Sylvester operator $\Delta^{[D(\tilde{p}), D(\tilde{q})]}(\cdot) : \mathbf{F}^{n \times n} \to \mathbf{F}^{n \times n}$ such that $\Delta^{[D(\tilde{p}), D(\tilde{q})]}(C_{G_{r'} H_{r'}}(\tilde{p}, \tilde{q})) = G_{r'} H_{r'}^T$. Let*

$$C_{G_{r'} H_{r'}}(\tilde{p}, \tilde{q}) = \begin{pmatrix} A & B \\ D & F \end{pmatrix}, \qquad (4.1)$$

where A, B, D, F and S as (3.1) and (3.2). Then the displacement ranks of A, B, D, F and S are all less than r'.

Fact 2 *([3]) Every Cauchy matrix is nonsingular and every square submatrix of Cauchy matrix is also nonsingular.*

Fact 3 *([20]) The matrix $C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})$ as in Theorem 2 is a strongly nonsingular matrix.*

Now let us describe the complexity of the modefied Recursive Triangular Factorization to Cauchy-like matrices.

Theorem 4 *The arithmetic complexity of the solution of a strongly nonsingular generalized Cauchy-like equations is bounded by $O(nr^2 \log^2 n)$ operations.*

Proof. Let $\tilde{v}$ be a given vector and $C_{G_r H_r}(\tilde{e}, \tilde{f}) \in \mathbf{F}^{n \times n}$ be a given generalized Cauchy-like matrix satifies $\Delta^{[D(\tilde{e}), D(\tilde{f})]}(C_{G_r H_x}(\tilde{e}, \tilde{f})) = G_r H_r^T$. By the theorem 2, the computations of $\tilde{x} = C_{G_r H_r}(\tilde{e}, \tilde{f})^{-1} \tilde{v}$ are equivalent to the computations of $\tilde{x} = C(\tilde{f}, \tilde{q}) C_{G_{r+2} H_{r+2}}(\tilde{p}, \tilde{q})^{-1} C(\tilde{p}, \tilde{e}) \tilde{v}$. We choose the new vectors $\tilde{p}$, $\tilde{q}$, where the elements of the vectors are n-th roots of 1 and -1, e.g. $\tilde{p} = \tilde{\alpha}$, $\tilde{q} = \tilde{\beta}$, where α_i and β_j of definition 1. Compute the generators of the matrix $G_{r'}$ and $H_{r'}$ by corollary 2. Then, we apply the Recursive Triangular Factorization Algorithms 3.1 to the matrix $C_{G_{r'} H_{r'}}(\tilde{\alpha}, \tilde{\beta})$. Partition the matrix into $\frac{n}{2} \times \frac{n}{2}$ blocks,

$$C_{G_{r'} H_{r'}}(\tilde{\alpha}, \tilde{\beta}) = \begin{pmatrix} A & B \\ D & F \end{pmatrix}, \tag{4.2}$$

where A, B, D, F and S as (3.1) and (3.2). We compute the generators of A, B, D, F, A^{-1}, S as (3.2), S^{-1}. This process continues until we have the generators of the inverse $C_{G_{r'} H_{r'}}(\tilde{\alpha}, \tilde{\beta})^{-1}$. Finally, we compute $\tilde{x} = C(\tilde{f}, \tilde{q}) C_{G_{r'} H_{r'}}(\tilde{\alpha}, \tilde{\beta})^{-1} C(\tilde{p}, \tilde{e}) \tilde{v}$. The computations of the generators of $C_{G_{r'} H_{r'}}(\tilde{\alpha}, \tilde{\beta})^{-1}$ in each step involve with the multiplications of a special Cauchy matrix times a vector. The complexity of $O(nr^2 \log^2 n)$ operations is followed by Corollary 1, 2, Lemma 1 and Theorem 2. *Q.E.D.*

Acknowledgments

Notes

1. Established in 1946, New York City Technical College is the senior college of technology at the City University of New York. It is a recognized national model for urban technological education and a pioneer in integrating technology into the teaching/learning experience. More than 11,000 students currently are enrolled in 50 career-specific baccalaureate, associate and specialized certificate programs in the technologies of art and design, business, computer systems, engineering, health care, hospitality, human services, the law-related professions, occupational and technology teacher education, and the liberal arts and sciences. Lo-

cated at 300 Jay Street in Downtown Brooklyn, City Tech is part of the MetroTech Center academic and commercial complex.

References

[1] D. Bini, V. Y. Pan, (1994), Polynomial and Matrix Computations, v. 1: Fundamental Algorithms, *Birkhaeuser* , Boston.

[2] R. R. Bitmead, B. D. O. Anderson, (1980), Asymptotically Fast Solution of Toeplitz and Related Systems of Linear Equations, *Linear Algebra Appl.*, **34**, 103-116.

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

[4] W.F. Donoghue, (1974), Monotone Matrix Functions and Analytic Continuation, *Springer, Berlin.*

[5] W. Gautschi, (1975), Norm Estimates for the Inverses of Vandermonde Matrices, *Numer. Math.*, **23**, 337-347.

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

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

[8] I. Gohberg, V. Olshevsky, (1994), Fast Algorithms with Preprocessing for Matrix-Vector Multiplication Problems, *J. of Complexity*, **10**, 4, 411-427.

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

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

[11] T. Kailath, A. Viera, and M. Morf, (1978), Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials, *SIAM Review*, 20, 1, pp. 106-119.

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

[13] M. Morf, (1980), Doubling Algorithms for Toeplitz and Related Equations, *Proc. IEEE Internat. Conf. on ASSP*, 954-959, IEEE Computer Society Press.

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

[15] V. Olshevsky, V. Y. Pan, (1999), Polynomial and Rational Evaluation and Interpolation (with Structured Matrices), *Proc. 26th Ann. Intern. Colloq. on Automata, Languages, Programming (ICALP'99), Lecture Notes in Computer Science*, Springer.

[16] V. Olshevsky, M. A. Shokrollahi, (1999), A Displacement Approach to Efficient Decoding of Algebraic-Geometric Codes, *Proc. 31st Ann. Symp. on Theory of Computing*, 235-244, ACM Press, New York.

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

[18] V. Y. Pan, M. AbuTabanjeh, Z. Chen, E. Landowne, A. Sadikou, (1998), New Transformations of Cauchy Matrices and Trummer's Problem, *Computers and Math. (with Applics.)*, **35, 12**, 1-5.

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

[20] V. Y. Pan, A. Zheng, (2000), Superfast algorithms for Cauchy-like matrix computations and extensions. *Linear Algebra Appl.* 310, no. 1-3, 83-108.

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

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

[23] L. Reichel, (1990), A Matrix Problem with Application to Rapid Solution of Integral Equations, *SIAM J. on Scientific and Stat. Computing*, **11**, 263-280.

[24] F. Rokhlin, (1985), Rapid Solution of Integral Equations of Classical Potential Theory, *J. Comput. Physics*, **60**, 187-207.

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