Computation *

Z. Chen
Mathematics Department
New York City College of Technology
300 Jay Street
Brooklyn, NY 11201
email: zchen@citytech.cuny.edu

Abstract

We will formulate and present a new class of C-Kronecker-like matrices with the displacement rank \( r \otimes r \), \( r < n \). It is an extension of the displacement structure to the a new class of matrices. The computational complexity of multiplication with vectors for C-Kronecker-like matrices has been accelerated. Applying the displacement, which was originally discovered by Kailath, Kung and Morf [14], a new superfast algorithm for the multiplication of a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field with a vector will be designed. The memory space cost of the number of the elements stored for a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field is \(O(r \log n)\). The cost of the number of the arithmetic operations for the product of a C-Kronecker-like matrix with the displacement rank \( r \otimes r \) and a Kronecker-vector is reduced dramatically to \(O(r \log n)\).

Key words: Structure matrices, Matrix-vector product, C-Kronecker-like matrices, Superfast algorithm.


68W35.

1 Introduction

Many active researchers are looking into fast algorithms for reducing the computational complexity. In fact, some of them have been developed many novel methods which have been implemented in our practical computations. For example, Kailath, Kung and Morf ([14]) have successfully exploited the displacement structure in fast computations. The displacement structure has applied in the efficient computations of many structured matrices such as Toeplitz-like [19]. A variety of applications includes conformal mapping, tangential Nevanlinna-Pick interpolation, convolution, solution of integral equations, rational interpolation, Fast Fourier Transform, Fast Cosine/Sine transform, algebraic decoding, linear processing, oil exploration, polynomial interpolation, polynomial evaluation, signal and image processing (cf. [27], [8], [26], [25], [6], [23], [15], [21], [20], [2], [7], [18], [25]).

To make our computer perform at the optimum speed, we always apply the mathematics theory in the software development. Reducing the computational complexity is one of the main goal for many active researchers. Many developments have been proposed to solve the problem of a
matrix-vector product more efficiently. In general, it is well known that a matrix-vector product costs at most $O(n^2)$ operations for a general matrix of the size $n \times n$. One simply way to speed up the computing process is to observe the special properties of matrices such as Cauchy, Hankel, Toeplitz, diagonal, tridiagonal and large sparse matrices over a field. For more complicated way, the displacement can be used ([9], [10]). A substantial improvement has been developed by using the displacement approach for matrices such as Cauchy-like, Hankel-like, Toeplitz-like and Confluent Cauchy-like (see, [13], [9], [10], [22]). One of the challenge problem is that whether the displacement approach can be extended to a new class of structured matrices.

The memory space to store the structured matrices can be saved by applying the displacement structure. The running time in computing the structured matrices can be reduced. Instead of working on the entries of a matrix itself, we may use the generators of a matrix in our computations. Many structured matrices can be re-constructed through their generators with the displacement operators (cf. [5], [9], [13], [11], [15], [16], [17], [24]).

Let recall the well known Sylvester operator, [14], [9], [10], $\Delta^{[V,U]}(\cdot) : \mathbb{F}^{n \times n} \rightarrow \mathbb{F}^{n \times n}$, acting on the linear space $\mathbb{F}^{n \times n}$ which is defined by

$$\Delta^{[V,U]}(C) = VC - CU,$$

where $\mathbb{F}$ is a field, $\vec{s}$ is denoted a vector, $C, V, U \in \mathbb{F}^{n \times n}$ are given matrices. Let the rank of image $\Delta^{[V,U]}(C)$ smaller than $n$, then $\Delta^{[V,U]}(C)$ can be re-written non-uniquely as

$$\Delta^{[V,U]}(C) = GH^T,$$

where $G, H \in \mathbb{F}^{n \times r}$ are called generator matrices. This displacement structure has been applied to computations of a Cauchy-like matrix, a Toeplitz-like matrix and a Hankel-like matrix. We will plan to extend this displacement approach to a new class of structured matrix, referred as a C-Kronecker-like matrix.

A new class of structured matrix has been formulated based on the displacement structure. As a result, a most efficient algorithm is presented to compute the product of a C-Kronecker-like matrix over a field using the generators. The memory space cost of the number of the elements stored for a C-Kronecker-like matrix of the size $(n \times n) \otimes (n \times n)$ over a field is $O(r \log n)$ with the displacement rank $r \otimes r$. The cost of the number of the arithmetic operations for a matrix-vector product is reduced to $O(r \log n)$.

The organization of this paper is as follows: In the section 2, the definitions and properties of C-Kronecker matrices are given. In section 3, a C-Kronecker-like matrix representation of the size $(n \times n) \otimes (n \times n)$ will be presented. In section 4, an efficient algorithm of the multiplication of C-Kronecker-like matrix will be shown. A summery is included in section 5.

Acknowledgment. I would like to express my deepest appreciation Dr. Pan.

2 Definitions and Properties

In this section, we will going to introduce some definitions and properties. The survey of the displacement structure is summarized as follow.

Definition 2.1 Let $\mathbb{F}$ be denoted as a field. $\mathbb{F}^{n \times n}$ is a linear space. $W \otimes z$ is a tensor space.

Definition 2.2 We denote $\vec{z} = z_0, \cdots, z_{n-1}$ as a vector, where $z_i \in \mathbb{F}$, for $i = 0, \cdots, n - 1$.

Definition 2.3 The transpose of a matrix $M$ or a vector $\vec{v}$ over a field $\mathbb{F}$ is written as $M^T$ or $\vec{v}^T$.

Definition 2.4 Let all elements of the vectors $\vec{a}$ and $\vec{b}$ be chosen from a field $\mathbb{F}$. A matrix is referred as a C-Kronecker matrix, which is defined as

$$S(\vec{a}, \vec{b}) \otimes S(\vec{a}, \vec{b}) = \left( \frac{1}{a_i - b_j} \right)_{i,j=0}^{n-1}$$

(2.1)
Proposition 2.1 ([12]) The C-Kronecker product is a bilinear and associative and satisfies the following: where \( L, F, J \in \mathbb{F}^{n \times n} \) and \( \alpha \in \mathbb{F}^{*} \):

- C-Kronecker product is not commutative.
- \((L \otimes F) + (J \otimes Q) = (L + J) \otimes (F + Q)\)
- \(\alpha(L \otimes F) = L \otimes (\alpha F) = (\alpha L) \otimes F\)
- \((L \otimes F) \otimes J = L \otimes (F \otimes J)\)
- \((L + F) \otimes J = L \otimes J + F \otimes J\)
- \(L \otimes (F + J) = L \otimes F + L \otimes J\)
- \((L \otimes F)(J \otimes Q) = (LJ \otimes FQ)\)

Lemma 2.1 Let the Sylvester operator, \(\Delta^{[p,q]}(\cdot) : \mathbb{F}^{n \times n} \rightarrow \mathbb{F}^{n \times n}\), acting on the linear space \(\mathbb{F}^{n \times n}\) which is defined by

\[
\Delta^{[p,q]}(W) = PW - WQ,
\]

where \( P, Q, W \in \mathbb{F}^{n \times n} \) are given matrices, then the Sylvester operator \(\Delta(\cdot)\) is a linear operator.

Proof. Let \( W_1 \) and \( W_2 \) be any two matrices from \(\mathbb{F}^{n \times n}\) and \( \alpha \) be an element of \(\mathbb{F}^{*}\). It is easy to see that \( P(W_1 + W_2) - (W_1 + W_2)Q = (PW_1 - W_1Q) + (PW_2 - W_2Q) \) and \( P(\alpha W) - (\alpha W)Q = \alpha(PW - WQ) \). Hence, we have the following equations: \(\Delta^{[p,q]}(W_1 + W_2) = \Delta^{[p,q]}(W_1) + \Delta^{[p,q]}(W_2)\) and \(\Delta^{[p,q]}(\alpha W) = \alpha\Delta^{[p,q]}(W)\).

Q.E.D

Definition 2.5 Let \( x \) and \( y \) are different elements over a field \(\mathbb{F}\). The vectors \( x, y \) and \( 1 \) are denoted as \( x = x, \cdots, x \), \( y = y, \cdots, y \) and \( 1 = 1, \cdots, 1 \).

Definition 2.6 The inverse of a matrix \( M \) over a field \(\mathbb{F}\) is written as \( M^{-1} \).

Definition 2.7 A matrix of the size \((n \times n) \otimes (n \times n)\) over a field \(\mathbb{F}\) is called a C-Kronecker-like matrix if it associates with matrix \( S(\tilde{a}, \tilde{b}) = S(\tilde{a}, \tilde{b}) \) of (2.1) and satisfies the Sylvester equation of (1.2), where \( V = D(\tilde{a}) \otimes D(\tilde{a}) \), \( U = D(\tilde{b}) \otimes D(\tilde{b}) \), for the matrices \( G \otimes G, H \otimes H \in \mathbb{F}^{n \times r} \).

Definition 2.8 The matrices \( G, H \in \mathbb{F}^{n \times r} \) and \( S(\tilde{a}, \tilde{b}) \otimes S(\tilde{a}, \tilde{b}) \) as in the definition 2.7 are called the generators for a C-Kronecker-like matrix.

Theorem 2.1 ([9]) For any given arbitrary vector \( \hat{v} \in \mathbb{F}^{n \times 1} \) and a Cauchy \((S(\tilde{a}, \tilde{b}))\) matrix of the size \((n \times n)\) over a field then the product \( \hat{p} = (S(\tilde{a}, \tilde{b}))(\hat{v}) \) can be computed in \( O(n \log n) \) operations.

3 A C-Kronecker matrix generators

In this section, a new representation of a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field has developed. It is easy to see that a C-Kronecker matrix of the size \((n \times n) \otimes (n \times n)\) over a field, defined as the definition 2.4 is embedded in a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field. One of the most important property of a C-Kronecker matrix is having a low displacement rank.

Lemma 3.1 For any given arbitrary Kronecker-vector \( \hat{\hat{v}} \in \mathbb{F}^{n \times 1} \) and a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field as defined as the definition 2.4, then the product \( \hat{\hat{\hat{p}}} = (S(\tilde{a}, \tilde{b}) \otimes S(\tilde{a}, \tilde{b}))(\hat{\hat{v}} \otimes \hat{v}) \) can be computed in \( O(n \log n) \) operations.

Proof. Based on the proposition 2.1, we have \( \hat{\hat{\hat{p}}} = (S(\tilde{a}, \tilde{b}) \otimes S(\tilde{a}, \tilde{b}))(\hat{\hat{v}} \otimes \hat{v}) = (S(\tilde{a}, \tilde{b})\hat{v}) \otimes (S(\tilde{a}, \tilde{b})\hat{v}) \), the cost of computing the product \( \hat{\hat{\hat{p}}} \) is \( O(n \log n) \) operations.

Q.E.D

Lemma 3.2 Let \( S(\tilde{a}, \tilde{b}) \otimes S(\tilde{a}, \tilde{b}) \) be a matrix as defined as the definition 2.4. The displacement rank of a \( S(\tilde{a}, \tilde{b}) \otimes S(\tilde{a}, \tilde{b}) \) matrix is \( 1 \otimes 1 \).
Proof. From the Proposition 2.1, it follows from the equation below:
\[ \Delta[D(\hat{a}) \otimes D(\hat{b}), D(\hat{b}) \otimes D(\hat{b})] S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}) = D(\hat{a}) \otimes D(\hat{b})(S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}) - S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}))D(\hat{b}) \otimes D(\hat{b}) = D(\hat{a}) \otimes D(\hat{b})S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}) - S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}) D(\hat{b}) \otimes D(\hat{b}) = GH^T \] where \( G = 1 \otimes 1 \) and \( H = 1 \otimes 1 \). Q.E.D

Theorem 3.1 We let linear operator \( \Delta(\cdot) \) act on a tensor space with trivial kernel. Let \( W \) be a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field, which defined as definition 2.7 with the generators \( G, H \in F^{n \times r} \) and \( S(\hat{a}, \hat{b}) \otimes S(\hat{a}, \hat{b}) \) of (2.1). Then this matrix can be represented as
\[ W = \sum_{i=1}^{r} \sum_{j=1}^{r} D(\hat{g}_i)S(\hat{a}, \hat{b})D(\hat{h}_i) \otimes D(\hat{g}_j)S(\hat{a}, \hat{b})D(\hat{h}_j) \] (3.1)

Proof. Let the matrix \( W \) be given as (3.1). From the lemma 2.1, the operator \( \Delta(\cdot) \) of (1.1) is a linear operator. By the lemma 3.2, we have
\[ \Delta[D(\hat{a}) \otimes D(\hat{b}), D(\hat{b}) \otimes D(\hat{b})](W) = D(\hat{a}) \otimes D(\hat{b})(\sum_{i=1}^{r} D(\hat{g}_i)S(\hat{a}, \hat{b})D(\hat{h}_i) \otimes \sum_{j=1}^{r} D(\hat{g}_j)S(\hat{a}, \hat{b})D(\hat{h}_j) - (D(\hat{g}_i)S(\hat{a}, \hat{b})D(\hat{h}_i) \otimes \sum_{j=1}^{r} D(\hat{g}_j)S(\hat{a}, \hat{b})D(\hat{h}_j)))(D(\hat{b}) \otimes D(\hat{b})) = \sum_{i=1}^{r} D(\hat{g}_i)(D(\hat{a})S(\hat{a}, \hat{b}) - S(\hat{a}, \hat{b})D(\hat{b}))(\hat{h}_i) \otimes \sum_{i=1}^{r} D(\hat{g}_i)(D(\hat{a})S(\hat{a}, \hat{b}) - S(\hat{a}, \hat{b})D(\hat{b}))(\hat{h}_i) = GH^T. \] Q.E.D

Algorithm 4.1

Input: an arbitrary Kronecker-vector \( \hat{v} \otimes \hat{v} \) and a C-Kronecker-like matrix \( W \) of the size \((n \times n) \otimes (n \times n)\) over a field as the definition 2.7 with displacement rank \( r \otimes r \).

Output: the Kronecker-vector \( \hat{p} \) such that \( \hat{p} = W \hat{v} \otimes \hat{v} \)

Computations:

1. Apply the Corollary 4.1 to compute the vector \( \hat{i}_i = \sum_{i=1}^{r} D(\hat{g}_i)S(\hat{a}, \hat{b})D(\hat{h}_i) \hat{v} \)
2. Compute the vector \( \hat{p} \) such that \( \hat{p} = \hat{v} \otimes \hat{i}_i \).

5 Conclusion

We have use the linear map of the linear space to compute a new class of C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field. The computational complexity of multiplication with vectors for C-Kronecker-like matrices has been improved. The displacement approach is not only reduce the running time but also save the memory space in the computer calculation. The number of the elements needed to store in a computer has been saved to \( O(r \log n) \) for a C-Kronecker-like matrix of the size \((n \times n) \otimes (n \times n)\) over a field.
The running time to solve the problem of matrix-vector product has been accelerated to $O(r \log n)$ arithmetic operations for a class of C-Kronecker-like matrix with the displacement rank $r \otimes r$.

References


