A Fast Hybrid Transform Algorithm for Beam Digitization

Sirani M. Perera, Arjuna Madanayake, and Levi Lingsch
Departments of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, and Florida International University, Miami.

INTRODUCTION

The Fast Fourier Transform is used to compute the Discrete Fourier Transform (DFT) and its inverse efficiently. It is sufficient to quote "The Fast Fourier Transform (FFT) is the most important numerical algorithm of our lifetime", by Gilbert Strang. Analogues of the DFT, such as the Discrete Cosine Transform (DCT) and the Discrete Sine Transform (DST) are the building blocks for the algorithms used in image and video compression such as JPEG. Without exaggeration, we can say that the FFT, DCT, and DST are the pillars of modern multimedia signal processing and play crucial roles in increasing throughput and decreasing integrated circuit complexity, manufacturing cost, and power consumption. The Beam Digitization Transform (BDT) is defined explicitly below;

$$A_n = [a_{ij}]_{i,j=0}^{n-1} = \left[\sum_{k=0}^{n-1} \epsilon_n(k)\, \omega_n^{ik} \cos\frac{k(2j+1)\pi}{2n}\right]_{i,j=0}^{n-1}$$

where $\epsilon_n(0) = \epsilon_n(n) = \frac{1}{\sqrt{2}}, \epsilon_n(k) = 1$ for $n \geq 2$ is an even integer and $\omega_n = e^{-2\pi i/n}$ is the primitive root of unity.

Two main techniques, namely polynomial arithmetic and matrix factorization techniques, can be used to derive fast discrete transform algorithms, which lead to the BDT algorithm. The polynomial arithmetic technique uses a divide-and-conquer strategy to reduce the degree of the polynomials. The matrix factorization technique directly factorizes the discrete transform matrices into the product of sparse matrices. If the factorizations do not preserve orthogonality, the resulting algorithms can lead to interior numerical instability.

METHODOLOGY

The reduction and minimization of the number of analog-to-digital converters (ADCs) is of paramount importance in array processing receivers with applications in wireless communications, radar, microwave imaging and radio astronomy. We propose a fast, exact, and stable BDT algorithm via a novel matrix factorization technique in connection to Chebyshev-like polynomials. The proposed BDT algorithm is used to explore spectral analysis and thereafter to minimize the number of ADCs in the design stage.

CONCLUSION

Results on the proposed project lead to:

- **Sparse and Orthogonal/Unitary Factorization**
- **Recursive BDT Algorithm**
- **Reduction in Complexity from *O(n^3)* to *O(n log n)***
- **Reduction in Number of ADCs**
- **Minimize Chip Area and Reduce Power Consumption**

RESULTS

The BDT is defined as $\boldsymbol{A_n} = \boldsymbol{F_n C_n^{II}}$, where $\boldsymbol{F_n}$ is the DFT matrix and $\boldsymbol{C_n^{II}}$ is the type II DCT matrix. The matrix $\boldsymbol{A_n}$ can be factored into the product of sparse and orthogonal matrices for any $n = 2^t \;\; (t \geq 1)$;

$$\boldsymbol{A_n} = \widetilde{D}_n \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & F_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} A_{\frac{n}{2}} & 0 \\ 0 & I_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & W_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & C_{\frac{n}{2}}^{III} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & B_{\frac{n}{2}}^{T} \end{bmatrix} H_n,$$

where

$$\widetilde{D}_n = \begin{bmatrix} I_{\frac{n}{2}} & D_{\frac{n}{2}} \\ I_{\frac{n}{2}} & -D_{\frac{n}{2}} \end{bmatrix}, H_n = \begin{bmatrix} I_{\frac{n}{2}} & \tilde{I}_{\frac{n}{2}} \\ I_{\frac{n}{2}} & -\tilde{I}_{\frac{n}{2}} \end{bmatrix},$$

$$C_n^{III} = H_n^T \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & W_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} C_{\frac{n}{2}}^{III} & 0 \\ 0 & C_{\frac{n}{2}}^{III} \end{bmatrix} \begin{bmatrix} I_{\frac{n}{2}} & 0 \\ 0 & B_{\frac{n}{2}}^{T} \end{bmatrix},$$

$$F_n = \frac{1}{\sqrt{n}} [\omega_n^{jk}]_{j,k=0}^{n-1},\; B_{\frac{n}{2}}^T = \begin{bmatrix} \sqrt{2} & & & & \\ 1 & 1 & & & \\ & 1 & \ddots & & \\ & & \ddots & 1 & \\ & & & 1 & 1 \end{bmatrix},$$

$$D_{\frac{n}{2}} = diag[\omega_n^k]_{k=1}^{\frac{n}{2}}, W_{\frac{n}{2}} = \frac{1}{2}\, diag\left[\sec(\frac{(2k-1)\pi}{2n})\right]_{k=1}^{\frac{n}{2}}.$$

Recursive BDT Algorithm

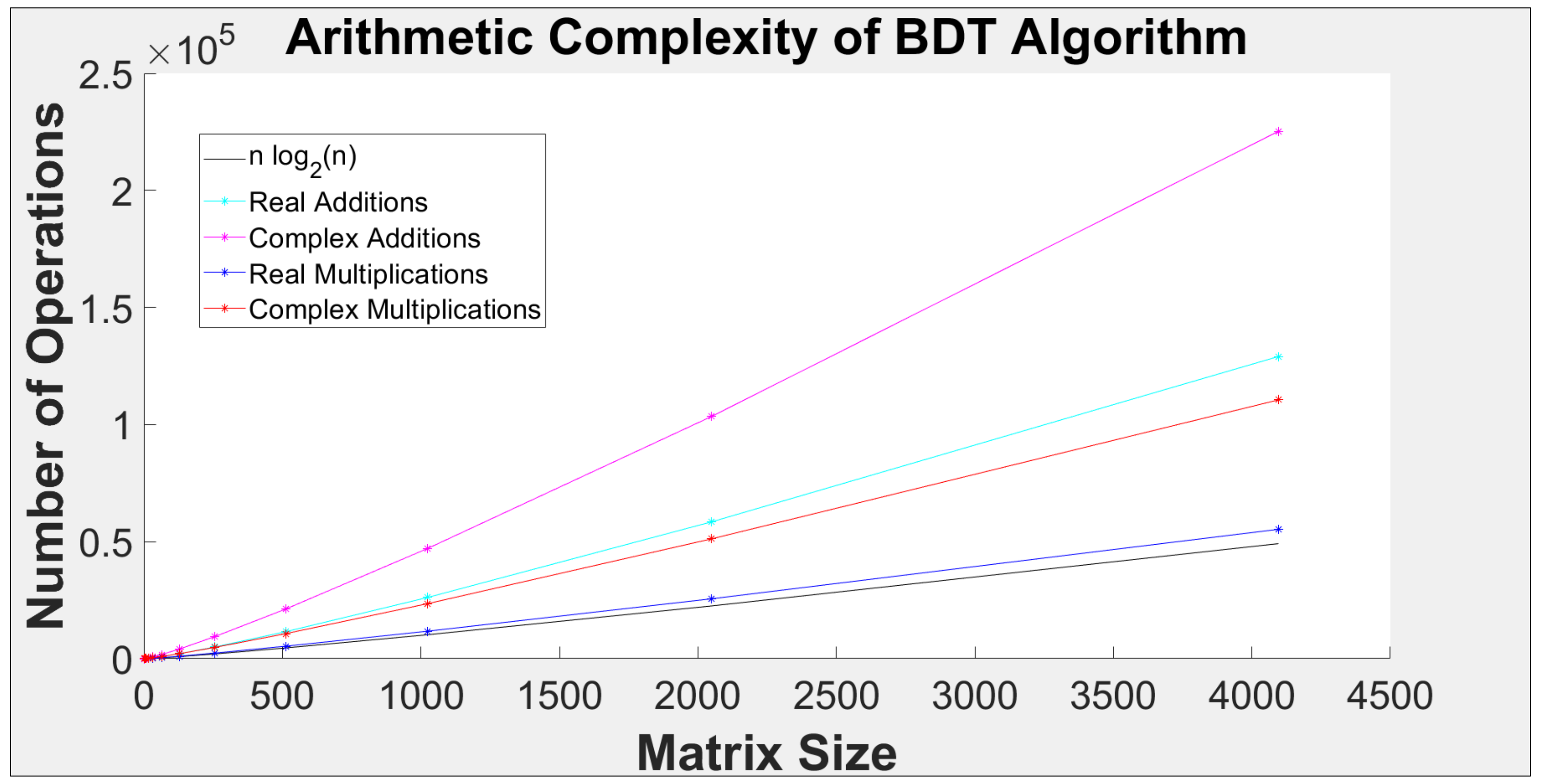
```
Input: $n = 2^t (t \geq 1), n_1 = \frac{n}{2}, x \in \mathbb{R}^n$ or $\mathbb{C}^n$.

Steps:
If $n = 2$, then
    $y := \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ x.
If $n \geq 4$, then
    $u := H_n x,$
    $v := \text{blkdiag}\left(I_{n_1}, B_n^T\right)u$
    $p := \text{blkdiag}\left(I_{n_1}, C_n^{III}\right)v$
    $[q_j]_{j=0}^{n-1} := \text{blkdiag}\left(I_{n_1}, W_n\right)p$
    $a := A\left([q_j]_{j=0}^{n_1-1}, n_1\right),$
    $b := I_{n_1}[q_j]_{j=n_1}^{n}$
    $r := \text{blkdiag}\left(I_{n_1}, F_{n_1}\right)(a^T, b^T)^T$
    $y := \widetilde{D}_n r.$

Output: $y = \boldsymbol{A_n} x$
```

RESULTS

Arithmetic Complexity

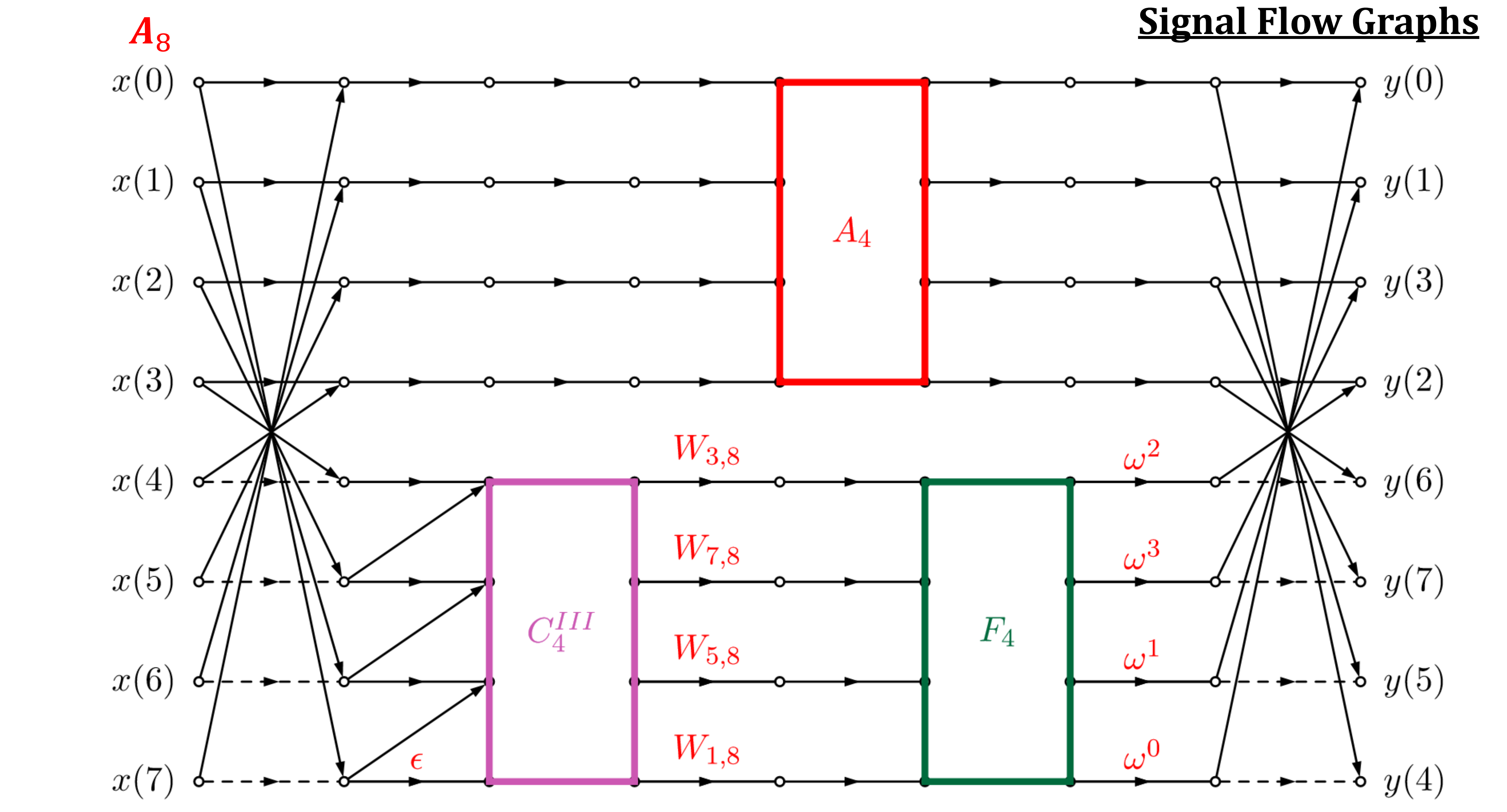
Real Input	**Complex Input**
$\mathbb{A}(\boldsymbol{A_n}, n) = 3nt - \frac{9}{2}n + 4t - 1$	$\mathbb{A}(\boldsymbol{A_n}, n) = \frac{9}{2}nt + n - 2t - 9$
$\mathbb{M}(\boldsymbol{A_n}, \text{n}) = nt + \frac{3}{2}n - 2t - 1$	$\mathbb{M}(\boldsymbol{A_n}, \text{n}) = 2nt + 3n - 4t - 2$

The BDT algorithm costs *O(n log n)* operations as opposed to the brute-force calculation with *O(n^3)* operations.

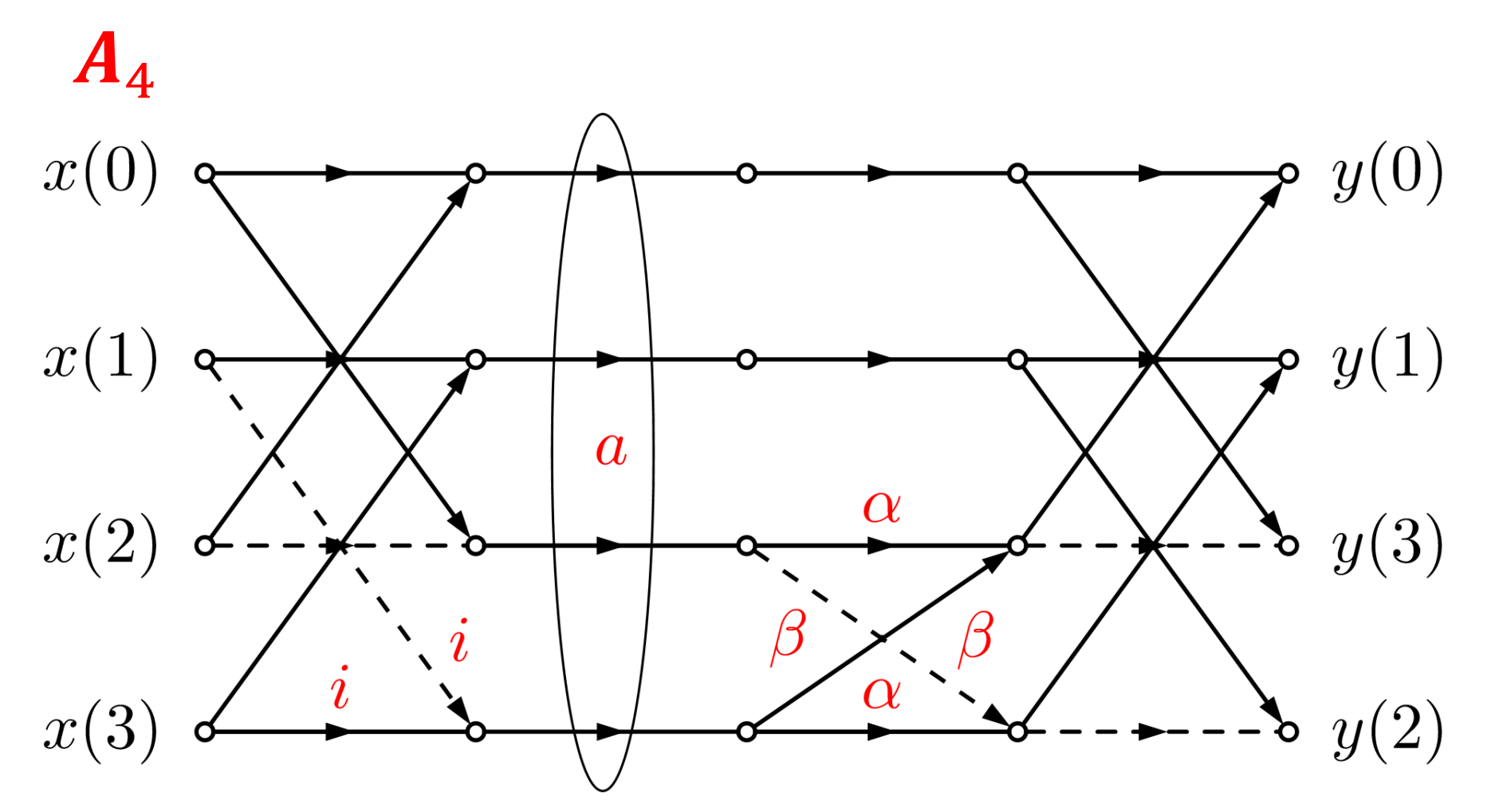


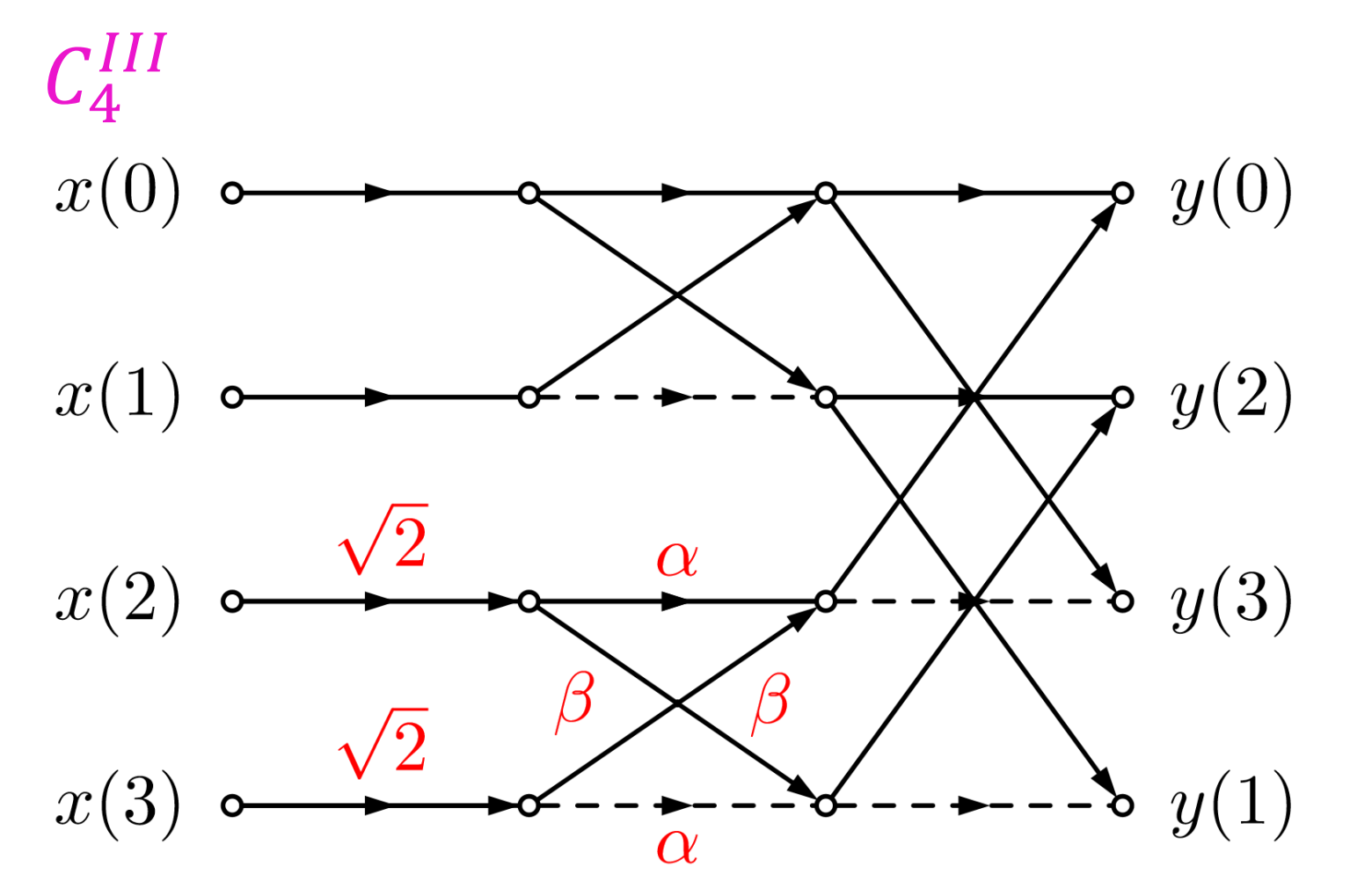
3.0508 E5 ≤ Speed Improvement Factor ≤ 1.2433 E6
With respect to complex addition (min) and real multiplication (max)

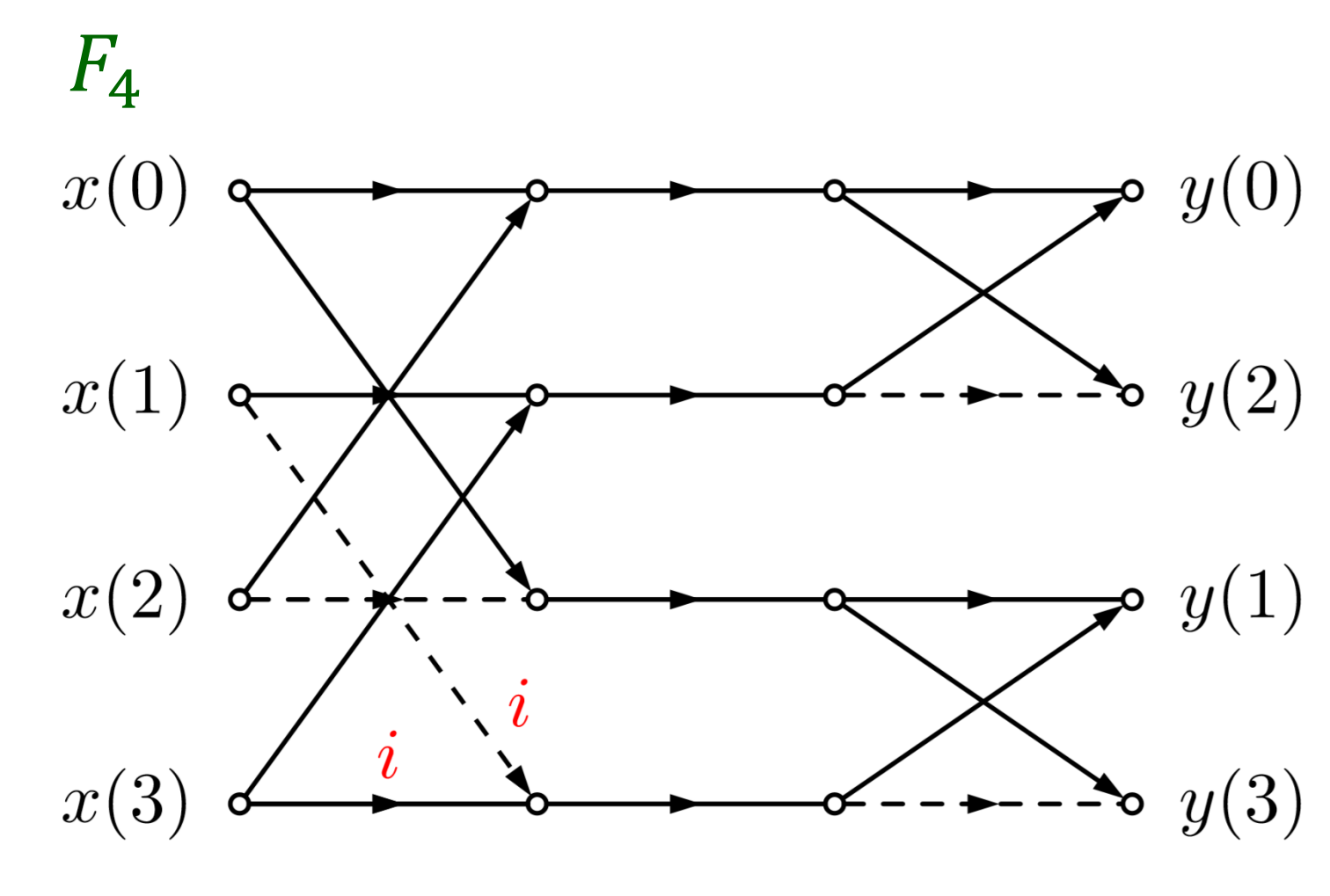
Signal Flow Graphs



a represents 2
α represents $\cos(\frac{\pi}{8})$
β represents $\sin(\frac{\pi}{8})$


A_4
$x(0)$
$x(1)$
$x(2)$
$x(3)$
a
i
i
α
β
β
α
$y(0)$
$y(1)$
$y(3)$
$y(2)$


C_4^{III}
$x(0)$
$x(1)$
$x(2)$
$x(3)$
$\sqrt{2}$
$\sqrt{2}$
α
β
β
α
$y(0)$
$y(2)$
$y(3)$
$y(1)$


F_4
$x(0)$
$x(1)$
$x(2)$
$x(3)$
i
i
$y(0)$
$y(2)$
$y(1)$
$y(3)$