

# Advanced Systems Lab

Spring 2022

*Lecture:* Discrete Fourier transform, fast Fourier transforms

**Instructor:** Markus Püschel, Ce Zhang

**TA:** Joao Rivera, several more

**ETH**

Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

## Linear Transforms

Overview: Transforms and algorithms

Discrete Fourier transform

Fast Fourier transforms

After that:

- *Optimized implementation and autotuning (FFTW)*
- *Automatic program synthesis (Spiral)*

## FFT References

Complexity: *Bürgisser, Clausen, Shokrollahi, Algebraic Complexity Theory, Springer, 1997*

History: *Heideman, Johnson, Burrus: Gauss and the History of the Fast Fourier Transform, Arch. Hist. Sc. 34(3) 1985*

FFTs:

- *Cooley and Tukey, "An algorithm for the machine calculation of complex Fourier series," Math. of Computation, vol. 19, pp. 297–301, 1965*
- *Nussbaumer, Fast Fourier Transform and Convolution Algorithms, 2nd ed., Springer, 1982*
- *van Loan, Computational Frameworks for the Fast Fourier Transform, SIAM, 1992*
- *Tolimieri, An, Lu, Algorithms for Discrete Fourier Transforms and Convolution, Springer, 2nd edition, 1997*
- *Franchetti, Püschel, Voronenko, Chellappa and Moura, Discrete Fourier Transform on Multicore, IEEE Signal Processing Magazine, special issue on "Signal Processing on Platforms with Multiple Cores", Vol. 26, No. 6, pp. 90-102, 2009*

3

## Linear Transforms

Very important class of functions: signal processing, communication, scientific computing, ...

*Mathematically:* Change of basis = Multiplication by a fixed matrix  $T$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = y = Tx \quad \leftarrow \quad \boxed{T} \quad \leftarrow \quad x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

$T = [t_{k,\ell}]_{0 \leq k, \ell < n}$

**Output**  **Input**

Equivalent definition: Summation form

$$y_k = \sum_{\ell=0}^{n-1} t_{k,\ell} x_{\ell}, \quad 0 \leq k < n$$

4

## Linear Transforms

Compute:  $y = Tx$        $x$ : input vector,  $y$ : output vector,  $T$ : fixed transform matrix

### Example: Discrete Fourier transform (DFT)

1. form (standard in signal processing):

given:  $x_0, \dots, x_{n-1}$

compute:  $y_k = \sum_{\ell=0}^{n-1} e^{-2k\ell\pi i/n} x_\ell, \quad k = 0, \dots, n-1$

$= \sum_{\ell=0}^{n-1} \omega_n^{k\ell} x_\ell, \quad k = 0, \dots, n-1, \quad \omega_n = e^{-2\pi i/n}$

primitive nth root of 1

2. form (we will use):

given:  $(x_0, \dots, x_{n-1})^T$

compute:  $y = \text{DFT}_n \cdot x, \quad \text{DFT}_n = [\omega_n^{k\ell}]_{0 \leq k, \ell < n}$

How does the  $\text{DFT}_2$  matrix look?  
Second row of  $\text{DFT}_4$  matrix?

5

## Smallest Relevant Example: DFT, Size 2

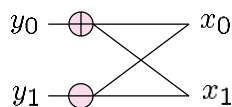
Transform (matrix):  $\text{DFT}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$

How many ops to compute the  $\text{DFT}_2$  of a vector?

Computation:  $y = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x$

or  $y_0 = x_0 + x_1$   
 $y_1 = x_0 - x_1$

As graph (direct acyclic graph or DAG):



called a butterfly



[http://charlottessmartypoints.blogspot.com/2011\\_02\\_01\\_archive.html](http://charlottessmartypoints.blogspot.com/2011_02_01_archive.html)

6

## DFT, Size 4

$$\text{DFT}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix}$$

How many (complex) operations to compute the  $\text{DFT}_4$  of a (complex) vector?

$$y = \text{DFT}_4 \cdot x$$

7

## Transforms: Examples

A few dozen transforms are relevant

Some examples

$$\begin{aligned} \text{DFT}_n &= [e^{-2k\ell\pi i/n}]_{0 \leq k, \ell < n} \\ \text{RDFT}_n &= [r_{k\ell}]_{0 \leq k, \ell < n}, \quad r_{k\ell} = \begin{cases} \cos \frac{2\pi k\ell}{n}, & k \leq \lfloor \frac{n}{2} \rfloor \\ -\sin \frac{2\pi k\ell}{n}, & k > \lfloor \frac{n}{2} \rfloor \end{cases} \quad \text{universal tool} \\ \text{DHT} &= [\cos(2k\ell\pi/n) + \sin(2k\ell\pi/n)]_{0 \leq k, \ell < n} \\ \text{WHT}_n &= \begin{bmatrix} \text{WHT}_{n/2} & \text{WHT}_{n/2} \\ \text{WHT}_{n/2} & -\text{WHT}_{n/2} \end{bmatrix}, \quad \text{WHT}_2 = \text{DFT}_2 \\ \text{IMDCT}_n &= [\cos((2k+1)(2\ell+1+n)\pi/4n)]_{0 \leq k < 2n, 0 \leq \ell < n} \quad \text{MPEG} \\ \text{DCT-2}_n &= [\cos(k(2\ell+1)\pi/2n)]_{0 \leq k, \ell < n} \quad \text{JPEG} \\ \text{DCT-3}_n &= \text{DCT-2}_n^T \quad (\text{transpose}) \\ \text{DCT-4}_n &= [\cos((2k+1)(2\ell+1)\pi/4n)]_{0 \leq k, \ell < n} \end{aligned}$$

8

## Transform Algorithms

An algorithm for  $y = Tx$  is given by a factorization

$$T = T_1 T_2 \cdots T_m$$

Namely, instead of  $y = Tx$  we can compute in steps

$$\left. \begin{aligned} t_1 &= T_m x \\ t_2 &= T_{m-1} t_1 \\ &\dots \\ y &= T_1 t_{m-1} \end{aligned} \right\} m \text{ steps}$$

This reduces the op count only if:

- the  $T_i$  are sparse
- $m$  is not too large

### Example: Cooley-Tukey Fast Fourier Transform (FFT), size 4

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} x = \begin{bmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & -1 & \cdot \\ \cdot & 1 & \cdot & -1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & i \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ 1 & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} x$$

12 adds  
4 mults by i
4 adds
1 mult by i
4 adds
0 ops

9

## Cooley-Tukey FFT, $n = 4$

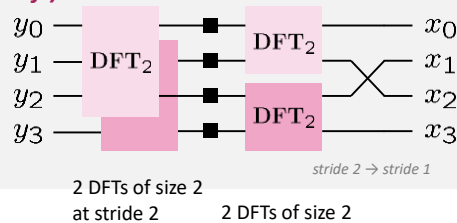
### Fast Fourier transform (FFT)

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & i & -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{bmatrix} = \begin{bmatrix} 1 & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & 1 \\ 1 & \cdot & -1 & \cdot \\ \cdot & 1 & \cdot & -1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & i \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdot & \cdot \\ 1 & -1 & \cdot & \cdot \\ \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & 1 & \cdot \\ \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

### Representation using matrix algebra

$$\text{DFT}_4 = (\text{DFT}_2 \otimes \text{I}_2) \text{diag}(1, 1, 1, i) (\text{I}_2 \otimes \text{DFT}_2) \text{L}_2^4$$

### Data flow graph (right to left)



10

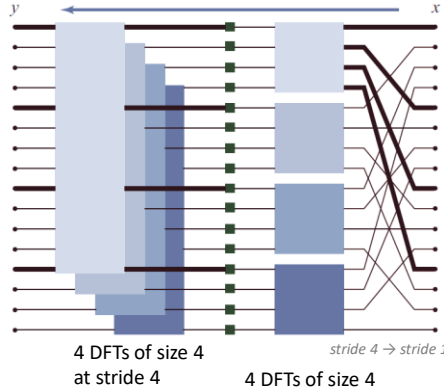
# Example Recursive FFT, n = 16, radix 4

$$\text{DFT}_{16} = \text{DFT}_4 \otimes I_4 \quad T_4^{16} \quad I_4 \otimes \text{DFT}_4 \quad L_4^{16}$$

Kronecker product:

$$A = [a_{ij}]_{i,j}$$

$$A \otimes B = [a_{ij} \cdot B]_{i,j}$$



## General Radix, Recursive Cooley-Tukey FFT

Assume  $n = km$ :  $\text{DFT}_n = (\text{DFT}_k \otimes I_m) \underbrace{T_m^n(I_k \otimes \text{DFT}_m)}_{\text{permutation matrix}} L_k^n$   
diagonal matrix with roots of unity

3 key structures:  $I_k \otimes A_m, A_k \otimes I_m, L_k^n$

$$y = (I_k \otimes A_m)x = \begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix} * x$$

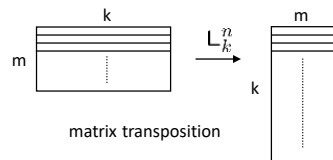
for  $i = 0:k-1$   
 $y[im:im+m-1] = A*x[im:im+m-1]$

$$y = (A_k \otimes I_m)x = \begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix} * x$$

m A's at stride m

for  $i = 0:m-1$   
 $y[i:m:i+(k-1)m] = A*x[i:m:i+(k-1)m]$

$y = L_k^n x$  view  $x$  as  $m \times k$  matrix:

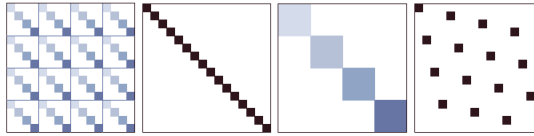


for  $i = 0:k-1$   
 for  $j = 0:m-1$   
 $y[im+j] = x[i+kj]$

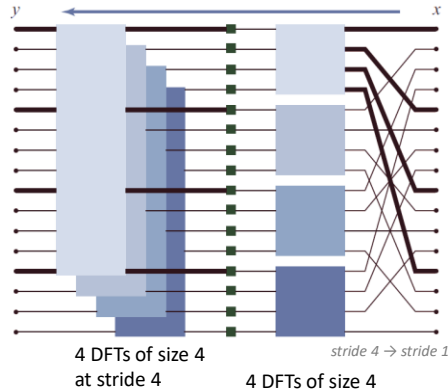
read at stride k, write at stride 1

equivalent: read at stride 1, write at stride m

## Example FFT, $n = 16$ (Recursive, Radix 4)

$$\text{DFT}_{16} = \text{DFT}_4 \otimes I_4 \quad T_4^{16} \quad I_4 \otimes \text{DFT}_4 \quad L_4^{16}$$


DFT<sub>4</sub> is expanded recursively



13

## Recursive Cooley-Tukey FFT

$$\text{DFT}_{km} = (\text{DFT}_k \otimes I_m) T_m^{km} (I_k \otimes \text{DFT}_m) L_k^{km} \quad \text{decimation-in-time}$$

$$\text{DFT}_{km} = L_m^{km} (I_k \otimes \text{DFT}_m) T_m^{km} (\text{DFT}_k \otimes I_m) \quad \text{decimation-in-frequency}$$

For powers of two  $n = 2^t$  sufficient together with base case

$$\text{DFT}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Cost:

- (complex adds, complex mults) =  $(n \log_2(n), n \log_2(n)/2)$   
independent of recursion
- (real adds, real mults)  $\leq (3n \log_2(n), 2n \log_2(n)) = 5n \log_2(n)$  flops  
depends on recursion: best is at least radix-8

14

# Recursive vs. Iterative FFT

Recursive, radix-k Cooley-Tukey FFT

$$\text{DFT}_{km} = (\text{DFT}_k \otimes \text{I}_m) T_m^{km} (\text{I}_k \otimes \text{DFT}_m) L_k^{km}$$

$$\text{DFT}_{km} = L_m^{km} (\text{I}_k \otimes \text{DFT}_m) T_m^{km} (\text{DFT}_k \otimes \text{I}_m)$$

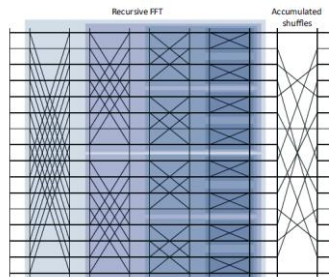
Iterative, radix 2, decimation-in-time/decimation-in-frequency

$$\text{DFT}_{2^t} = \left( \prod_{j=1}^t (\text{I}_{2^{j-1}} \otimes \text{DFT}_2 \otimes \text{I}_{2^{t-j}}) \cdot (\text{I}_{2^{j-1}} \otimes T_{2^{t-j}}^{2^{t-j+1}}) \right) \cdot R_{2^t}$$

$$\text{DFT}_{2^t} = R_{2^t} \cdot \left( \prod_{j=1}^t (\text{I}_{2^{t-j}} \otimes T_{2^{j-1}}^{2^j}) \cdot (\text{I}_{2^{t-j}} \otimes \text{DFT}_2 \otimes \text{I}_{2^{j-1}}) \right)$$

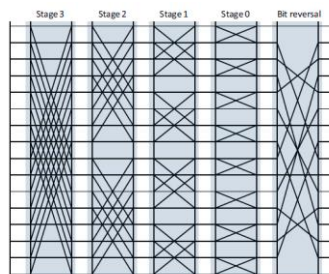
15

## Radix 2, recursive



$$(\text{DFT}_2 \otimes \text{I}_8) T_8^{16} \left( \text{I}_2 \otimes \left( (\text{DFT}_2 \otimes \text{I}_4) T_4^8 \left( \text{I}_2 \otimes \left( (\text{DFT}_2 \otimes \text{I}_2) T_2^4 \left( \text{I}_2 \otimes \text{DFT}_2 \right) L_2^2 \right) L_2^2 \right) \right) \right) L_2^{16}$$

## Radix 2, iterative



$$\left( (\text{I}_1 \otimes \text{DFT}_2 \otimes \text{I}_8) D_8^{16} \right) \left( (\text{I}_2 \otimes \text{DFT}_2 \otimes \text{I}_4) D_4^{16} \right) \left( (\text{I}_4 \otimes \text{DFT}_2 \otimes \text{I}_2) D_2^{16} \right) \left( (\text{I}_8 \otimes \text{DFT}_2 \otimes \text{I}_1) D_1^{16} \right) R_2^{16}$$



## Recursive vs. Iterative

Iterative FFT computes in stages of butterflies =  
 *$\log_2(n)$  passes through the data*

Recursive FFT reduces passes through data =  
*better locality*

Same computation graph but different topological sorting

Rough analogy:

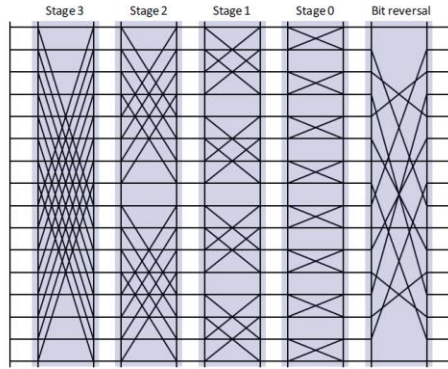
MMM	DFT
Triple loop	Iterative FFT
Blocked	Recursive FFT

17

## The FFT Is Very Malleable

18

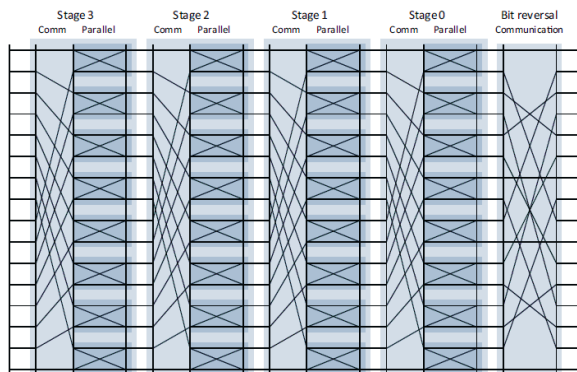
## Iterative FFT, Radix 2



$$\left( (I_1 \otimes \text{DFT}_2 \otimes I_8) D_0^{16} \right) \left( (I_2 \otimes \text{DFT}_2 \otimes I_4) D_1^{16} \right) \left( (I_4 \otimes \text{DFT}_2 \otimes I_2) D_2^{16} \right) \left( (I_8 \otimes \text{DFT}_2 \otimes I_1) D_3^{16} \right) R_2^{16}$$

19

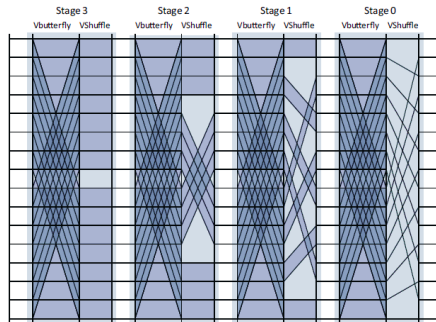
## Pease FFT, Radix 2



$$\left( L_2^{16} (I_8 \otimes \text{DFT}_2) D_0^{16} \right) \left( L_2^{16} (I_8 \otimes \text{DFT}_2) D_1^{16} \right) \left( L_2^{16} (I_8 \otimes \text{DFT}_2) D_2^{16} \right) \left( L_2^{16} (I_8 \otimes \text{DFT}_2) D_3^{16} \right) R_2^{16}$$

20

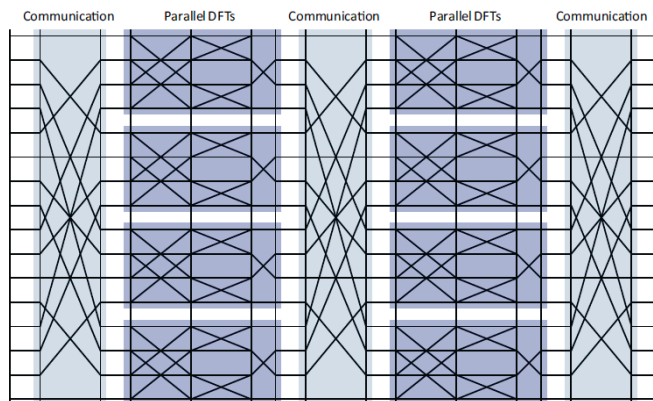
## Stockham FFT, Radix 2



$$\left( (DFT_2 \otimes I_8) D_0^{16} (L_2^2 \otimes I_8) \right) \left( (DFT_2 \otimes I_8) D_1^{16} (L_2^4 \otimes I_4) \right) \left( (DFT_2 \otimes I_8) D_2^{16} (L_2^8 \otimes I_2) \right) \left( (DFT_2 \otimes I_8) D_3^{16} (L_2^{16} \otimes I_1) \right)$$

21

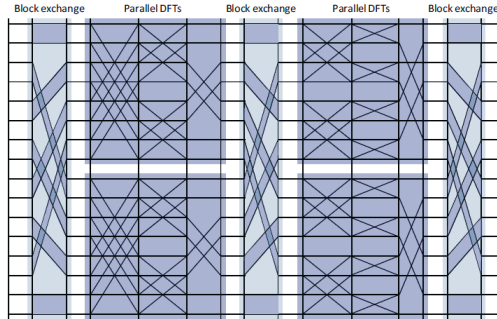
## Six-Step FFT



$$L_4^{16} \left( I_4 \otimes ((DFT_2 \otimes I_2) T_2^4 (I_2 \otimes DFT_2) L_2^4) \right) L_4^{16} T_4^{16} \left( I_4 \otimes ((DFT_2 \otimes I_2) T_2^4 (I_2 \otimes DFT_2) L_2^4) \right) L_4^{16}$$

22

# Multi-Core FFT



$$(L_4^8 \otimes I_2) \left( I_2 \otimes \left( (\text{DFT}_2 \otimes I_2) T_2^4 (I_2 \otimes \text{DFT}_2) L_2^4 \otimes I_2 \right) \right) (L_2^8 \otimes I_2) T_4^{16} \left( I_2 \otimes \left( I_2 \otimes (\text{DFT}_2 \otimes I_2) T_2^4 (I_2 \otimes \text{DFT}_2) \right) R_2^8 \right) (L_2^8 \otimes I_2)$$

# Transform Algorithms

$$\begin{aligned} \text{DFT}_n &\rightarrow P_{k/2,2m}^\top \left( \text{DFT}_{2m} \oplus (I_{k/2-1} \otimes C_{2m} \text{rDFT}_{2m}(i/k)) \right) (\text{RDFT}'_k \otimes I_m), \quad k \text{ even,} \\ \begin{bmatrix} \text{RDFT}_n \\ \text{RDFT}'_n \\ \text{DHT}_n \\ \text{DHT}'_n \end{bmatrix} &\rightarrow (P_{k/2,2m}^\top \otimes I_2) \begin{bmatrix} \text{RDFT}_{2m} \\ \text{RDFT}'_{2m} \\ \text{DHT}_{2m} \\ \text{DHT}'_{2m} \end{bmatrix} \oplus \left( I_{k/2-1} \otimes D_{2m} \begin{bmatrix} \text{rDFT}_{2m}(i/k) \\ \text{rDFT}'_{2m}(i/k) \\ \text{rDHT}_{2m}(i/k) \\ \text{rDHT}'_{2m}(i/k) \end{bmatrix} \right) \begin{bmatrix} \text{RDFT}'_k \\ \text{RDFT}_k \\ \text{DHT}'_k \\ \text{DHT}_k \end{bmatrix} \otimes I_m, \quad k \text{ even,} \\ \begin{bmatrix} \text{rDFT}_{2n}(u) \\ \text{rDHT}_{2n}(u) \end{bmatrix} &\rightarrow L_m^{2n} \left( I_k \otimes \begin{bmatrix} \text{rDFT}_{2m}((i+u)/k) \\ \text{rDHT}_{2m}((i+u)/k) \end{bmatrix} \right) \left( \begin{bmatrix} \text{rDFT}_{2k}(u) \\ \text{rDHT}_{2k}(u) \end{bmatrix} \otimes I_m \right), \\ \text{RDFT-3}_n &\rightarrow (Q_{k/2,2m}^\top \otimes I_2) (I_k \otimes \text{rDFT}_{2m}(i+1/2/k)) (\text{RDFT-3}_k \otimes I_m), \quad k \text{ even,} \\ \text{DCT-2}_n &\rightarrow P_{k/2,2m}^\top (\text{DCT-2}_{2m} K_{2m}^2 \oplus (I_{k/2-1} \otimes N_{2m} \text{RDFT-3}_{2m}^\top)) B_n(L_{k/2}^{n/2} \otimes I_2) (I_m \otimes \text{RDFT}'_k) Q_{m/2,k}, \\ \text{DCT-3}_n &\rightarrow \text{DCT-2}_n^\top, \\ \text{DCT-4}_n &\rightarrow Q_{k/2,2m}^\top (I_{k/2} \otimes N_{2m} \text{RDFT-3}_{2m}^\top) B'_n(L_{k/2}^{n/2} \otimes I_2) (I_m \otimes \text{RDFT-3}_k) Q_{m/2,k}, \\ \text{DFT}_n &\rightarrow (\text{DFT}_k \otimes I_m) T_m^n (I_k \otimes \text{DFT}_m) L_m^n, \quad n = km \quad \text{Cooley-Tukey FFT} \\ \text{DFT}_n &\rightarrow P_p(\text{DFT}_k \otimes \text{DFT}_m) Q_n, \quad n = km, \text{gcd}(k, m) = 1 \quad \text{Prime-factor FFT} \\ \text{DFT}_p &\rightarrow R_p^\top (I_1 \oplus \text{DFT}_{p-1}) D_p (I_1 \oplus \text{DFT}_{p-1}) R_p, \quad p \text{ prime} \quad \text{Rader FFT} \\ \text{DCT-3}_n &\rightarrow (I_m \oplus J_m) L_m^n (\text{DCT-3}_m(1/4) \oplus \text{DCT-3}_m(3/4)) \\ &\quad \cdot (F_2 \otimes I_m) \begin{bmatrix} I_m & 0 & 0 & -J_{m-1} \\ & \frac{1}{\sqrt{2}} & & \\ & & \frac{1}{\sqrt{2}} & \\ & & & I_m \end{bmatrix}, \quad n = 2m \\ \text{DCT-4}_n &\rightarrow S_n \text{DCT-2}_n \text{diag}_{0 \leq k < n} (1/(2 \cos((2k+1)\pi/4n))) \\ \text{IMDCT}_{2m} &\rightarrow (J_m \oplus I_m \oplus I_m \oplus J_m) \left( \begin{bmatrix} 1 \\ & 1 \\ & & -1 \\ & & & -1 \end{bmatrix} \otimes I_m \right) \oplus \left( \begin{bmatrix} -1 \\ & -1 \\ & & 1 \\ & & & 1 \end{bmatrix} \otimes I_m \right) J_{2m} \text{DCT-4}_{2m} \\ \text{WHT}_{2k} &\rightarrow \prod_{i=1}^k (I_2^{k_1+\dots+k_{i-1}} \otimes \text{WHT}_{2k_i} \otimes I_2^{k_{i+1}+\dots+k_i}), \quad k = k_1 + \dots + k_t \\ \text{DFT}_2 &\rightarrow F_2 \\ \text{DCT-2}_2 &\rightarrow \text{diag}(1, 1/\sqrt{2}) F_2 \\ \text{DCT-4}_2 &\rightarrow J_2 R_{13\pi/8} \end{aligned}$$

## Complexity of the DFT

Measure:  $L_c$ ,  $2 \leq c$

- Complex adds count 1
- Complex mults by a constant  $a$  with  $|a| < c$  counts 1
- $L_2$  is strictest,  $L_\infty$  the loosest (and most natural)

Upper bounds:

- $n = 2^k$ :  $L_2(\text{DFT}_n) \leq 3/2 n \log_2(n)$  (using Cooley-Tukey FFT)
- General  $n$ :  $L_2(\text{DFT}_n) \leq 8 n \log_2(n)$  (needs Bluestein FFT)

Lower bound:

- Theorem by Morgenstern: If  $c < \infty$ , then  $L_c(\text{DFT}_n) \geq \frac{1}{2} n \log_c(n)$
- Implies: in the measure  $L_c$  the DFT is  $\Theta(n \log(n))$

25

## Lowest Known FFT Cost (Powers of 2)

A modified split-radix FFT with fewer arithmetic operations, *Johnson and Frigo, IEEE Trans. Signal Processing 55(1), pp. 111-119, 2007*

Number of flops ( $n = 2^k$ ):

$$\frac{34}{9} n \log_2(n) - \frac{124}{27} n - 2 \log_2(n) - \frac{2}{9} (-1)^{\log_2(n)} \log_2(n) + \frac{16}{27} (-1)^{\log_2(n)} + 8$$

26

## History of FFTs

The advent of digital signal processing is often attributed to the FFT  
(Cooley-Tukey 1965)

History:

- Around 1805: FFT discovered by Gauss [1]  
(Fourier publishes the concept of Fourier analysis in 1807!)
- 1965: Rediscovered by Cooley-Tukey

[1]: Heideman, Johnson, Burrus: "Gauss and the History of the Fast Fourier Transform" Arch. Hist. Sc. 34(3) 1985 <sup>27</sup>

## Carl-Friedrich Gauss



1777 - 1855

Contender for the greatest mathematician of all times

Some contributions: Modular arithmetic, least square analysis, normal distribution, fundamental theorem of algebra, Gauss elimination, Gauss quadrature, Gauss-Seidel, non-Euclidean geometry, ...

28