# Introduction to Scientific Computing Languages

Prof. Paolo Bientinesi

pauldj@aices.rwth-aachen.de







# **Fourier Transform**

Let  $f : \mathbf{R} \to \mathbf{C}$  be an integrable function. Its *Fourier Transform* F is defined as the function

$$F(t) = \int_{-\infty}^{+\infty} f(x)e^{-2\pi ixt} dx.$$

Fourier transform: decomposition of a function into a series of waves.

# **Fourier Transform**

Let  $f : \mathbf{R} \to \mathbf{C}$  be an integrable function. Its *Fourier Transform* F is defined as the function

$$F(t) = \int_{-\infty}^{+\infty} f(x)e^{-2\pi ixt} dx.$$

Fourier transform: decomposition of a function into a series of waves.

#### Discrete Fourier Transform

Let  $\{x_k\}_0^{n-1}$  be a sequence of complex numbers. Its *Discrete Fourier Transform* (DFT) is defined as the sequence

$$X_k = \sum_{j=0}^{n-1} x_j e^{-i2\pi \frac{k}{n}j}.$$

# Fast Fourier Transform

One of the top-10 algorithms of the century

$$y := F_m x, \qquad x, y \in \mathbb{R}^m$$

# Fast Fourier Transform

One of the top-10 algorithms of the century

$$y := F_m x, \qquad x, y \in \mathbb{R}^m$$

- $F_m = e^{\frac{-2\pi i}{m}kj}, \quad k, j \in [0, 1, \dots, m-1]$
- $F_m = F_m^T$  (symmetry)
- Matlab: fft, ifft, fftshift, ...

# Fast Fourier Transform

One of the top-10 algorithms of the century

#### Definition

$$y := F_m x, \qquad x, y \in \mathbb{R}^m$$

- $F_m = e^{\frac{-2\pi i}{m}kj}, \quad k, j \in [0, 1, \dots, m-1]$
- $F_m = F_m^T$  (symmetry)
- Matlab: fft, ifft, fftshift, ...

### **Recursive Definition**

$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$

$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$

Identity matrix 
$$I_k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}_{[k \times k]}$$

$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$

#### Fourier matrix

$$F_m = e^{\frac{-2\pi i}{m}kj}, \quad k, j \in [0, 1, \dots, m-1]$$

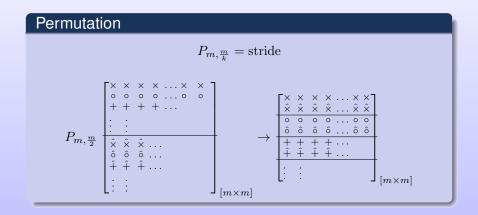
$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$

# Kronecker Product $A = \begin{bmatrix} a_{00} & a_{01} & \dots \\ a_{10} & a_{11} & \\ \vdots & & \ddots \end{bmatrix}, \quad A \otimes B = \begin{bmatrix} a_{00}B & a_{01}B & \dots \\ a_{10}B & a_{11}B & \\ \vdots & & \ddots \end{bmatrix}_{[?\times?]}$

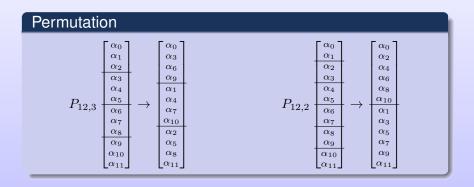
$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$

Twiddle Factors 
$$D_{k,\frac{m}{k}} = \begin{bmatrix} (D^0 = I) & 0 & \dots & 0 \\ 0 & D & & & \\ & & D^2 & & \\ \vdots & & & \ddots & \\ 0 & & & D^{k-1} \end{bmatrix}_{[m \times m]}$$
 
$$D = \operatorname{diag}(e^{\frac{-2\pi i}{m}j}), \quad j \in [0,\dots,\frac{m}{k}-1]$$

$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$



$$F_m = P_{m,\frac{m}{k}} \left( I_k \otimes F_{\frac{m}{k}} \right) D_{k,\frac{m}{k}} \left( F_k \otimes I_{\frac{m}{k}} \right)$$



# 2D FFT

# Definition

 $Y := F_m X F_n$ 

# 2D FFT

#### Definition

$$Y := F_m X F_n$$

• Sequence of 1d FFTs.

FFT of columns followed by FFT of rows, or viceversa

$$Y = (F_m X) F_n = F_m (X F_n)$$

# 2D FFT

#### **Definition**

$$Y := F_m X F_n$$

• Sequence of 1d FFTs.

FFT of columns followed by FFT of rows, or viceversa

$$Y = (F_m X) F_n = F_m (X F_n)$$

• 2-dimensional approach.

$$Y =$$

$$P_{m,\frac{m}{k}}(I_k \otimes F_{\frac{m}{k}})D_{k,\frac{m}{k}}(F_k \otimes I_{\frac{m}{k}})X(F_h \otimes I_{\frac{n}{h}})D_{h,\frac{n}{h}}(I_h \otimes F_{\frac{n}{h}})P_{n,\frac{n}{h}}^T$$

• Matlab: fft2, ifft2, ...