Discrete Fourier Transform (DFT) & Fast Fourier Transform (FFT)

— Yilin Jiang, Minxi Lin,
Puyuan Zhang, Yixiao Zhang

Guidelines

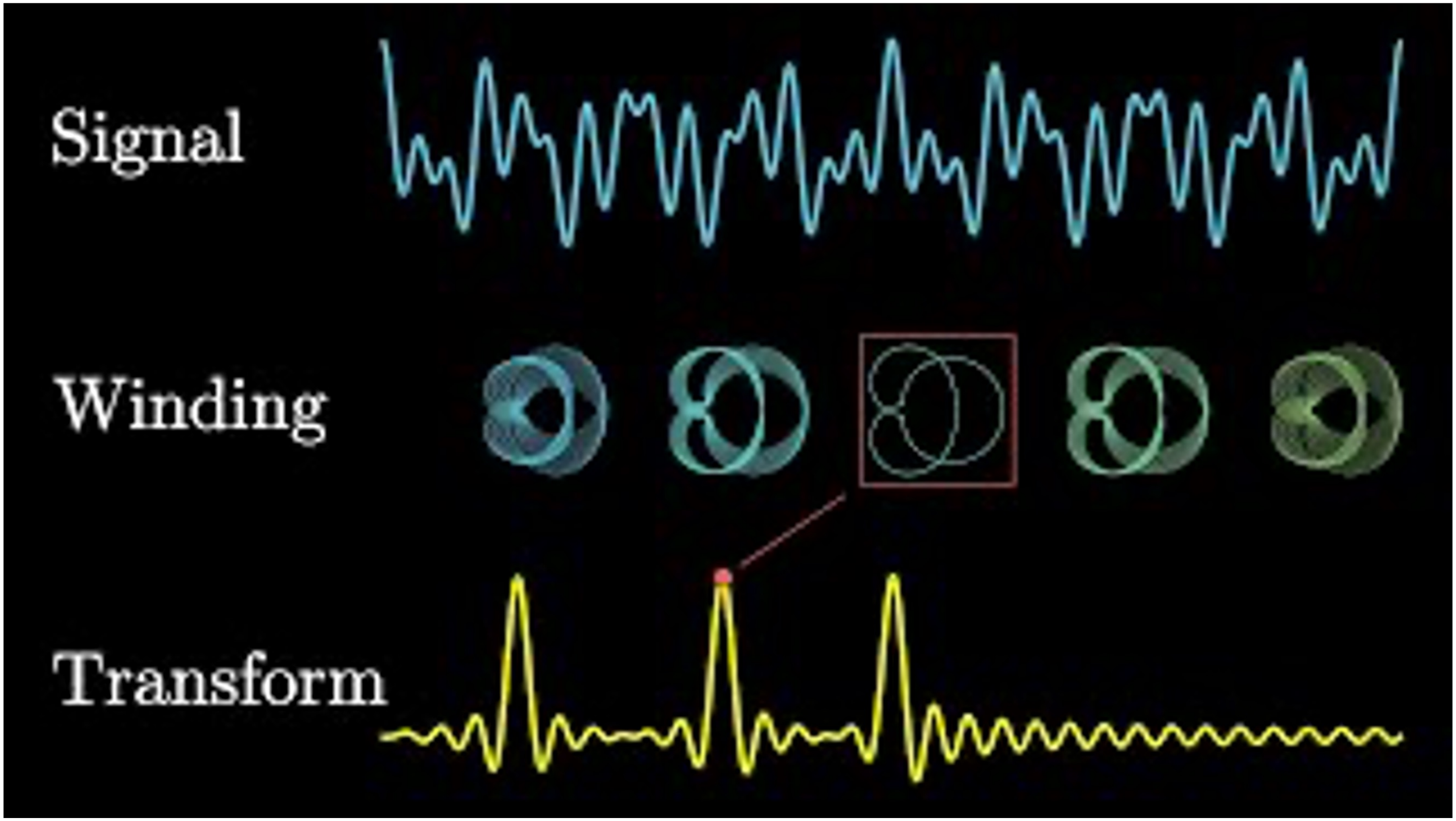
PART I: Introduction to Trigonometric Polynomial

PART II: Dive into DFT & FFT

Part III: Trigonometric Interpolation vs. Polynomial Interpolation

PART IV: Applications in Real Life

Consider Sound Filtering…



Introduction to Trigonometric Polynomial

Definition 3.3. *A function of the form*

$$s_n(x) = \sum_{k=-n}^{n} c_k e^{ikx}, \tag{3.147}$$

where $c_0, c_1, c_{-1}, \ldots, c_n, c_{-n}$ *are complex, or equivalently of the form*[5]

$$s_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx) \tag{3.148}$$

where the coefficients $a_0, a_1, b_1, \ldots, a_n, b_n$ *are real is called a* trigonometric polynomial of degree (at most) n.

Introduction to Trigonometric Polynomial

$$s_n(x) = \frac{1}{2}a_0 + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx)$$

The basis of this trigonometric polynomial is {1, cos x, sin x, cos 2x, sin 2x, …, cos nx, sin nx}

16:23 1月23日

$S_{\frac{N}{2}}(x) = \sum_{k=-N/2} C_k e^{\cdots}$

$$S_n(x) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cos kx + b_k \sin kx)$$
$$= \frac{a_0}{2} + (a_1\cos x + b_1 \sin x) + (a_2 \cos 2x + b_2 \sin 2x) + \cdots + (a_n \cos nx + b_n \sin nx)$$

$$\begin{bmatrix} \frac{a_0}{2} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & a_1 & b_1 & \cdots & & & \\ 0 & 0 & 0 & a_2 & b_2 & \cdots & \\ \vdots & & & \ddots & & & \vdots \\ & \cdots & & & & a_n & b_n \end{bmatrix} \underbrace{\begin{bmatrix} 1 \\ \cos x \\ \sin x \\ \cos 2x \\ \sin 2x \\ \vdots \\ \cos nx \\ \sin nx \end{bmatrix}}_{\text{basis}}$$

$S_n(x) \in \operatorname{span}\{1, \cos x, \sin x, \cdots, \cos nx, \sin nx\}$

离散傅里叶变换

连续信号：$x(t) \longrightarrow X(j\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$

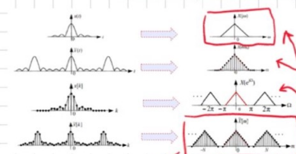
离散信号：$x[k] \longrightarrow X(e^{j\Omega}) = \sum_{k=-\infty}^{+\infty} x[k] e^{-j\Omega k}$ 无法用数字化的方法计算

因此，有必要分析有限长序列的傅里叶变换

由两个抽样定理，信号时域的离散化对应其频域的周期化

信号频域的离散化对应其时域的周期化

即：



只分析一个周期的情况

DFS：$\tilde{X}[m] = \sum_{k=0}^{N-1} \tilde{x}[k] \cdot e^{-j\frac{2\pi}{N}mk}$

$\Downarrow$

DFT：$X[m] = \sum_{k=0}^{N-1} x[k] \cdot e^{-j\frac{2\pi}{N}mk}$ $\quad 0 \le m \le N-1$

$\Downarrow$

IDFT：$x[k] = \frac{1}{N}\sum_{m=0}^{N-1} X[m] e^{j\frac{2\pi}{N}mk}$ $\quad 0 \le k \le N-1$

$X[m] = \mathrm{DFT}\{x[k]\}$; $x[k] = \mathrm{IDFT}\{x[m]\}$; $x[k] \xleftrightarrow{\mathrm{DFT}} X[m]$

例：已知序列 $x[k] = \{1,2,2,1;\ k=0,1,2,3\}$，求 $x[k]$ 的DFT

已知 $X[m] = \sum_{k=0}^{N-1} x[k] e^{-j\frac{2\pi}{N}mk} = \sum_{k=0}^{N-1} x[k] W_N^{mk}$，其中 $W_N^{mk} = e^{-j\frac{2\pi}{N}mk}$

故 $X[0] = x[0]W_4^0 + x[1]W_4^0 + x[2]W_4^0 + x[3]W_4^0 = 6$

$X[1] = x[0]W_4^0 + x[1]W_4^1 + x[2]W_4^2 + x[3]W_4^3 = -1-j$

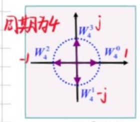
$X[2] = x[0]W_4^0 + x[1]W_4^2 + x[2]W_4^4 + x[3]W_4^6 = 0$

同理，$X[3] = -1+j$

故 $X[m] = \{6, -1-j, 0, -1+j;\ m=0,1,2,3\}$

用矩阵表示：
$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

周期为4



$\Downarrow$

$\begin{matrix} W_4^0 & W_4^2 \\ W_4^2 & W_4^1 \end{matrix}$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}$$

例：求有限长4点序列 $x[k] = \{1,1,1,1;\ k=0,1,2,3\}$ 的DFT

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

故 $X[m] = \{4,0,0,0;\ m=0,1,2,3\}$

如果在 $x[k]$ 后面补0，变成8点序列，其DFT会如何？

$x_0[k] = \{1,1,1,1,0,0,0,0\}$ $\quad X[m] = \sum_{k=0}^{N-1} x[k] W_N^{mk} = \sum_{k=0}^{7} x[k] W_8^{mk}$

$\Rightarrow X_{0m}[m] = \{4, 1-(1+\sqrt{2})j, 0, 1+(1-\sqrt{2})j, 0, 1+(-1+\sqrt{2})j, 0, 1+(1+\sqrt{2})j\}$

相当于在原来的 $\{4,0,0,0\}$ 中插入了4个值

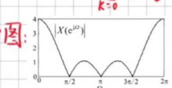
分别求序列 $x[k] = \{1,1,1,1\}$ 以及序列 $x_0[k] = \{1,1,1,1,0,0,0,0\}$ 的DTFT

$X(e^{j\Omega}) = \sum_{k=0}^{3} x[k] e^{-j\Omega k} = \sum_{k=0}^{3} e^{-j\Omega k}$

$X_0(e^{j\Omega}) = \sum_{k=0}^{7} x[k] e^{-j\Omega k} = \sum_{k=0}^{3} e^{-j\Omega k}$，故 $X(e^{j\Omega}) = X_0(e^{j\Omega})$

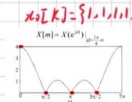
$= e^{-j\frac{3}{2}\Omega}\left[2\cos(\tfrac{1}{2}\Omega) + 2\cos(\tfrac{3}{2}\Omega)\right]$

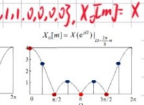
如图：



$x[k] = \{1,1,1,1\}$，$X[m] = X(e^{j\Omega})\big|_{\Omega = \frac{2\pi}{4}\cdot m}$ $\quad \Omega = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$

$x_0[k] = \{1,1,1,1,0,0,0,0\}$，$X_0[m] = X(e^{j\Omega})\big|_{\Omega = \frac{2\pi}{8}\cdot m}$ $\quad \Omega = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4},$





序列后补零，可获 $\frac{3\pi}{2}, \frac{7\pi}{4}$

得更多抽样值，观察 $X(e^{j\Omega})$ 更多细节

Introduction to Trigonometric Interpolation

Theorem 3.9.

$$s_{N/2}(x) = \sum_{k=-N/2}^{N/2}{}'' c_k e^{ikx} \tag{3.151}$$

interpolates $(j2\pi/N, f_j), j = 0, \ldots, N-1$ *if and only if*

$$c_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ik2\pi j/N}, \quad k = -\frac{N}{2}, \ldots, \frac{N}{2}. \tag{3.152}$$

Using the relations $c_0 = \frac{1}{2}a_0$, $c_k = \frac{1}{2}(a_k - ib_k)$, $c_{-k} = \bar{c}_k$, we find that

$$s_{N/2}(x) = \frac{1}{2}a_0 + \sum_{k=1}^{N/2-1}(a_k \cos kx + b_k \sin kx) + \frac{1}{2}a_{N/2}\cos\left(\frac{N}{2}x\right)$$

interpolates $(j2\pi/N, f_j), j = 0, \ldots, N-1$ if and only if

$$a_k = \frac{2}{N}\sum_{j=0}^{N-1} f_j \cos kx_j, \quad k = 0, 1, \ldots, N/2, \tag{3.161}$$

$$b_k = \frac{2}{N}\sum_{j=0}^{N-1} f_j \sin kx_j, \quad k = 1, \ldots, N/2-1. \tag{3.162}$$

The double prime in the summation sign means that the first and last terms (k = −N/2 and k = N/2) have a factor of 1/2.

Note that the interpolation nodes are equi-spaced points in [0, 2π]. One can accommodate any other period by doing a simple scaling.

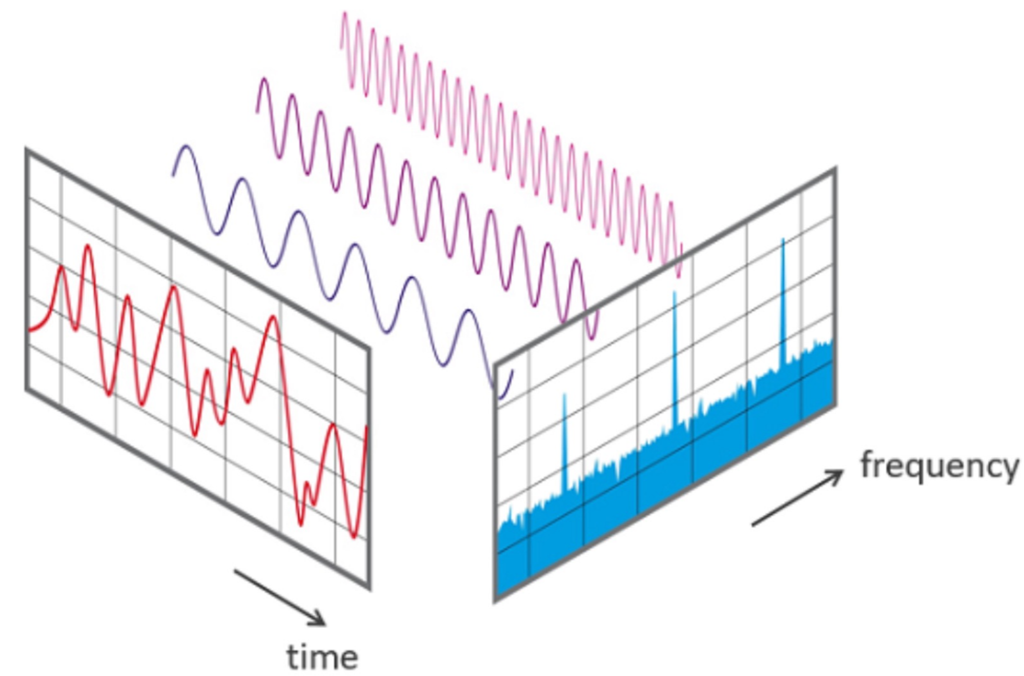
It is important to note that the DFT coefficients for k = N/2, . . . , N − 1 correspond to those for k = −N/2, . . . , −1 of the interpolating trigonometric polynomial $s_{N/2}$.

This set of coefficient is DFT:

$$c_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}, \quad k = 0, \ldots, N-1,$$

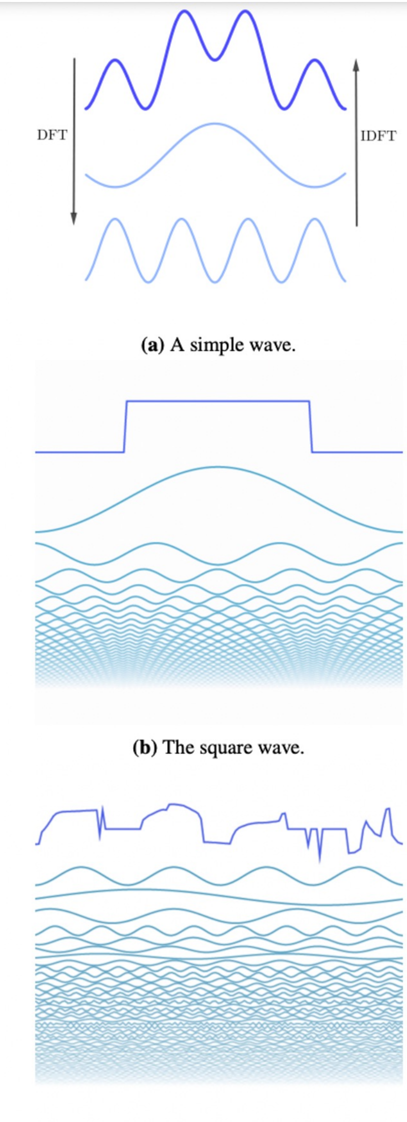
Dive into DFT

The discrete Fourier transform (DFT) algorithm transforms samples of signals from the time domain into the frequency domain. The DFT is widely used in the fields of spectral analysis, applied mechanics, acoustics, medical imaging, numerical analysis, instrumentation, and telecommunications.



Why is FFT called the "Fast" algorithm

DFT: $O(N^2)$
FFT: $O(N \log_2 N)$



(a) A simple wave.

(b) The square wave.

(c) A general wave function.

FFT (Fast Fourier Transform)

Formula for DFT: $c_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}, \quad k = 0, \ldots, N-1$ Operation: $O(n^2)$

Let us define $d_k = Nc_k$ for $k = 0, 1, \ldots, N-1$, and $\omega_N = e^{-i2\pi/N}$. Then we can rewrite the DFT (3.166) as

$$d_k = \sum_{j=0}^{N-1} f_j \omega_N^{kj}, \quad k = 0, 1, \ldots, N-1. \tag{3.168}$$

$$\begin{aligned} d_k &= \sum_{j=0}^{N-1} f_j w_N^{kj} \\ &= f_0 w_N^{k\cdot 0} + f_1 w_N^{k\cdot 1} + \cdots + f_{N-2} w_N^{k\cdot(N-2)} + f_{N-1} w_N^{k\cdot(N-1)} \\ &= (f_0 w_N^{k\cdot 0} + f_2 w_N^{k\cdot 2} + \cdots + f_{N-2} w_N^{k\cdot(N-2)}) + (f_1 w_N^{k\cdot 1} + f_3 w_N^{k\cdot 3} + f_{N-1} w_N^{k\cdot(N-1)}) \end{aligned}$$

FFT

$$\omega_N^{2jk} = e^{-i2jk\frac{2\pi}{N}} = e^{-ijk\frac{2\pi}{\frac{N}{2}}} = e^{-ijk\frac{2\pi}{n}} = \omega_n^{kj},$$
$$\omega_N^{(2j+1)k} = e^{-i(2j+1)k\frac{2\pi}{N}} = e^{-ik\frac{2\pi}{N}} e^{-i2jk\frac{2\pi}{N}} = \omega_N^k \omega_n^{kj}.$$

Let N = 2n, $d_k = \sum_{j=0}^{n-1} f_{2j} w_N^{k\cdot 2j} + \sum_{j=0}^{n-1} f_{(2j+1)} w_N^{k\cdot(2j+1)}$

$$= \sum_{j=0}^{n-1} f_{2j} w_{2n}^{k\cdot 2j} + \sum_{j=0}^{n-1} f_{(2j+1)} w_{2n}^{k\cdot(2j+1)}$$

$$= \sum_{j=0}^{n-1} f_{2j} w_n^{kj} + \sum_{j=0}^{n-1} f_{(2j+1)} w_N^k w_n^{kj}$$

$$= \underbrace{\sum_{j=0}^{n-1} f_j^e w_n^{kj}}_{\text{Size} = N/2} + w_N^k \underbrace{\sum_{j=0}^{n-1} f_j^o w_n^{kj}}_{\text{Size} = N/2}$$

denoting $f_j^e = f_{2j}$ and $f_j^o = f_{2j+1}$

Size = N/2 Size = N/2

FFT

Do the previous operations recursively, we can keep dividing the size of the DFT in half, until the size is equal to 1. At that point, we don't need to do multiplication.

Operation: $O(N \log_2 N)$

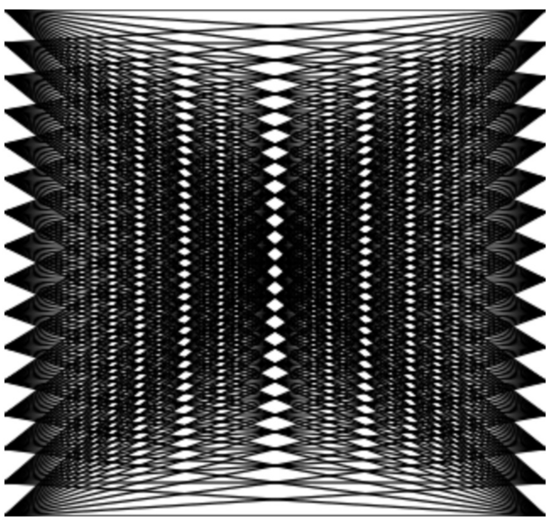
$$d_k = \sum_{j=0}^{n-1} f_j^e \omega_n^{jk} + \omega_N^k \sum_{j=0}^{n-1} f_j^o \omega_n^{jk}$$

Why is FFT faster than DFT

DFT: $O(N^2)$

$$c_k = \frac{1}{N}\sum_{j=0}^{N-1} f_j e^{-ikx_j}$$

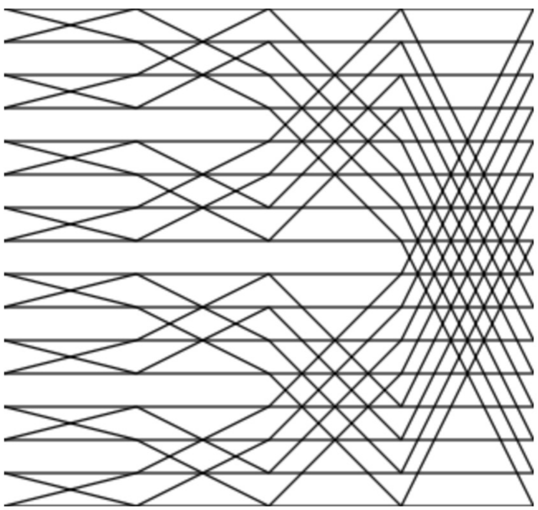
DFT, size 16



FFT: $O(N \log_2 N)$

$$d_k = \sum_{j=0}^{n-1} f_j^e \omega_n^{jk} + \omega_N^k \sum_{j=0}^{n-1} f_j^o \omega_n^{jk}$$

FFT, size 16



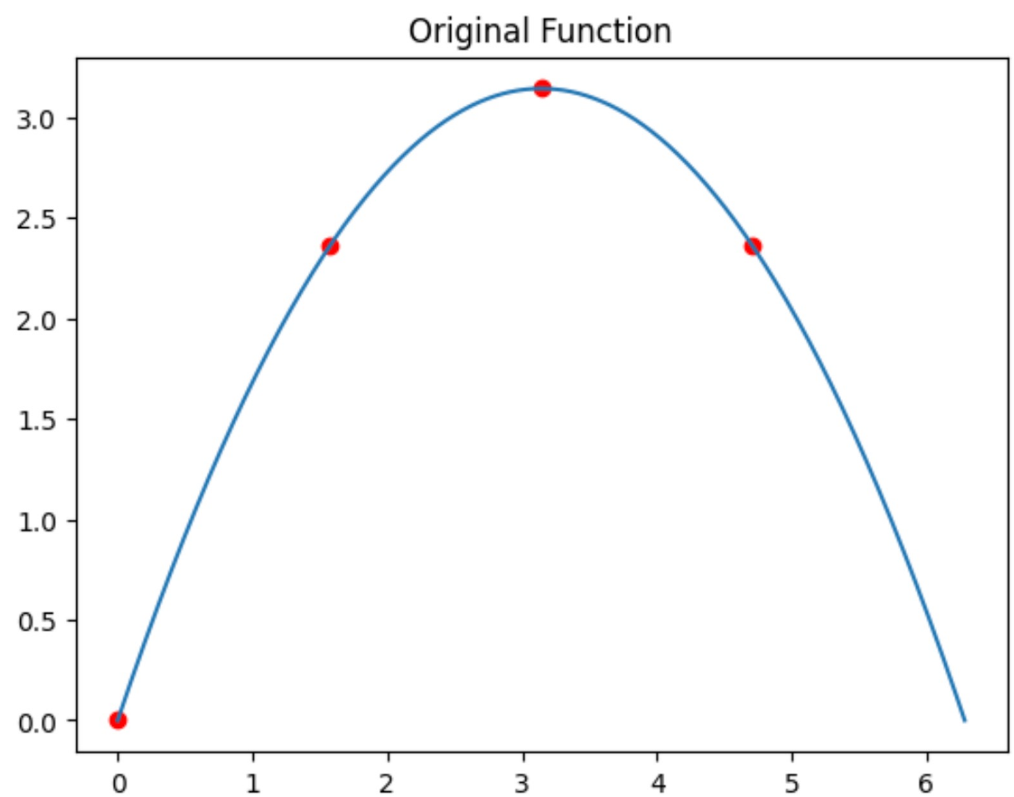
Back to Trigonometric Interpolation

（再强调下dft fft和interpolation的关系

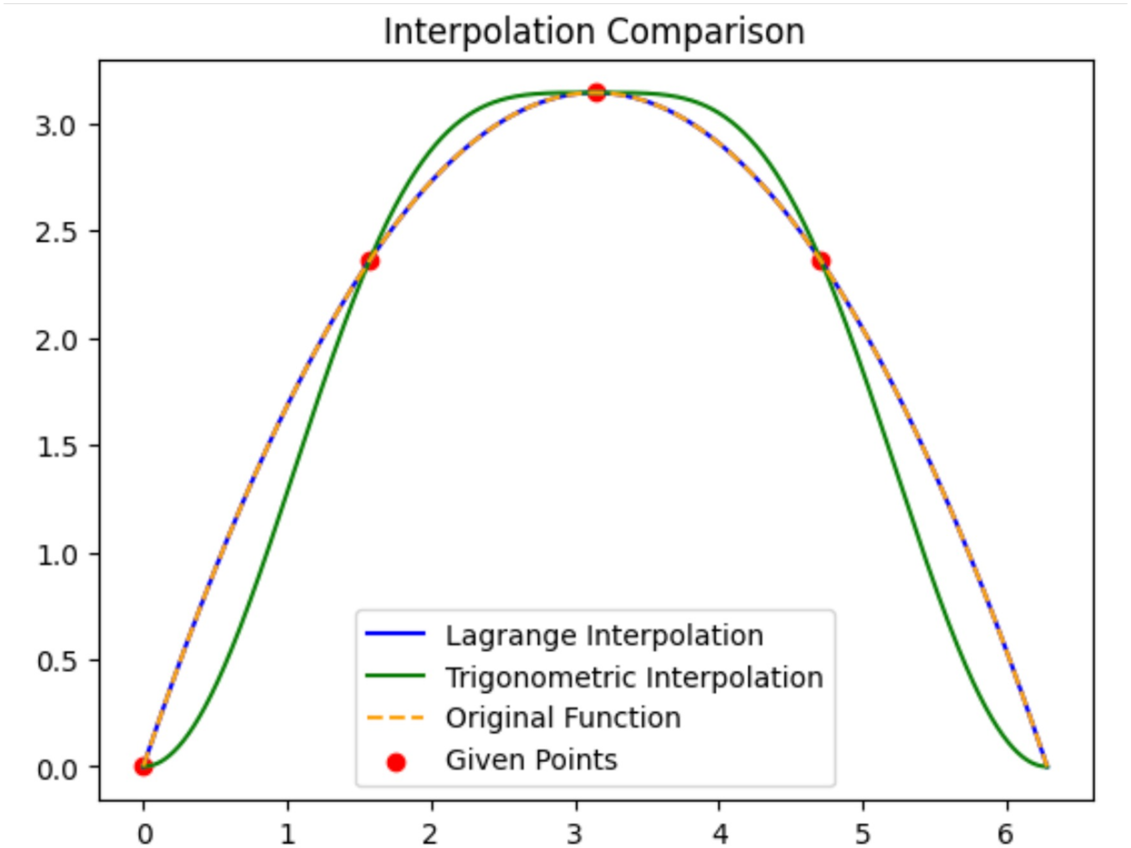
Visualization

Consider, for example, the function $f : [0, 2\pi] \to \mathbb{R}$ defined by $f(t) = 2t - \frac{t^2}{\pi}$.

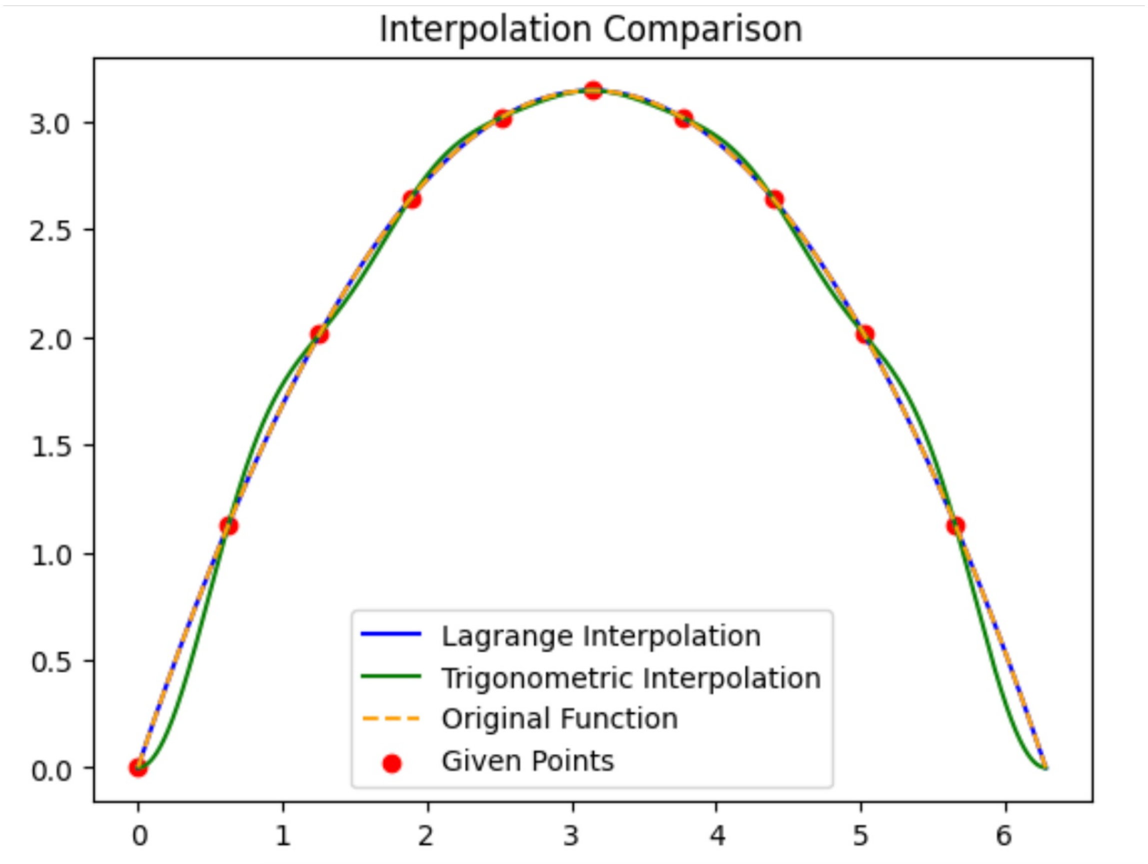
For N = 4, we have the following values $C = (0,0), (\pi/2, 3\pi/4), (\pi, \pi), (3\pi/2, 3\pi/4)$



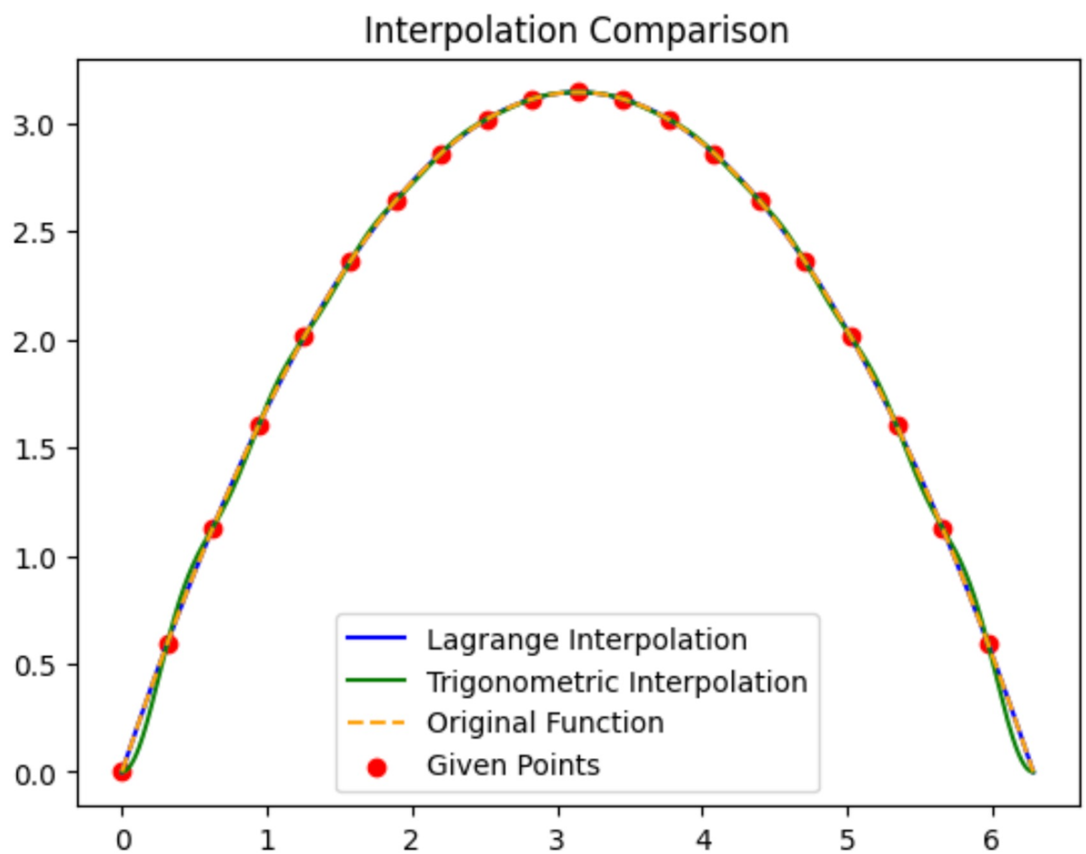
Visualization - Non-periodic Function - n=4



Visualization - Non-periodic Function - n=10



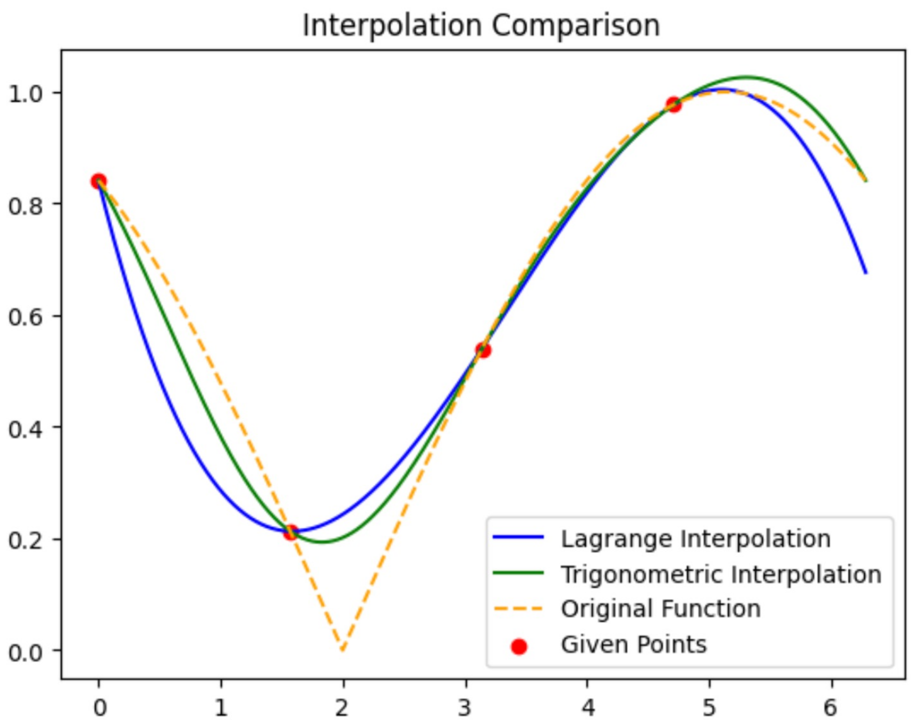
Visualization - Non-periodic Function - n=20



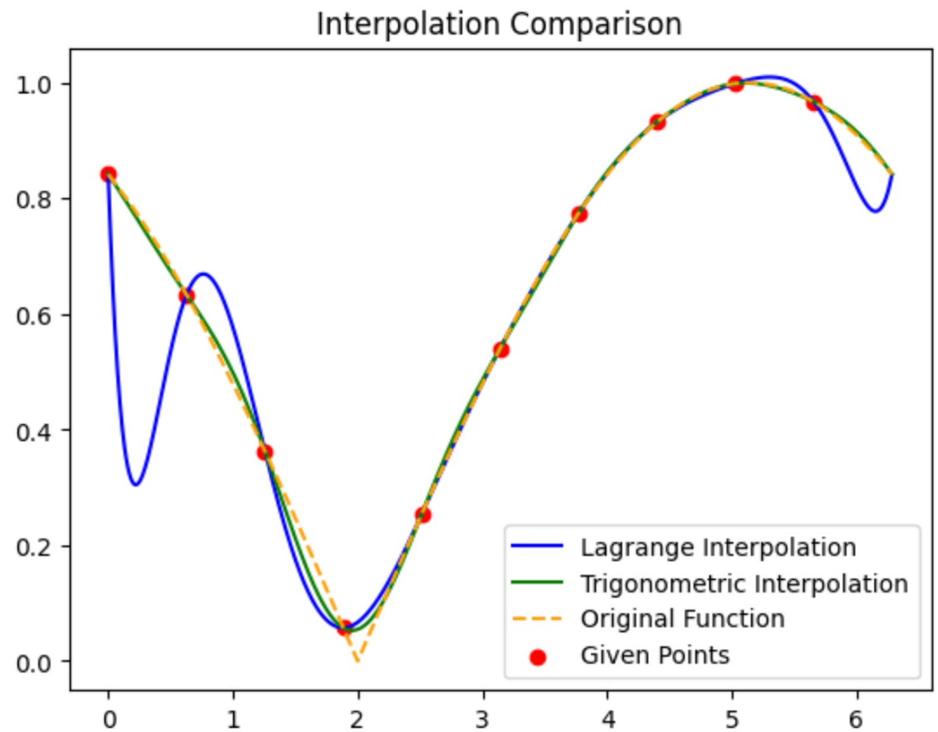
Visualization - Periodic Function

2π-periodic function $u(t) = \left|\sin\left(\frac{t}{2} - 1\right)\right|.$

n = 4

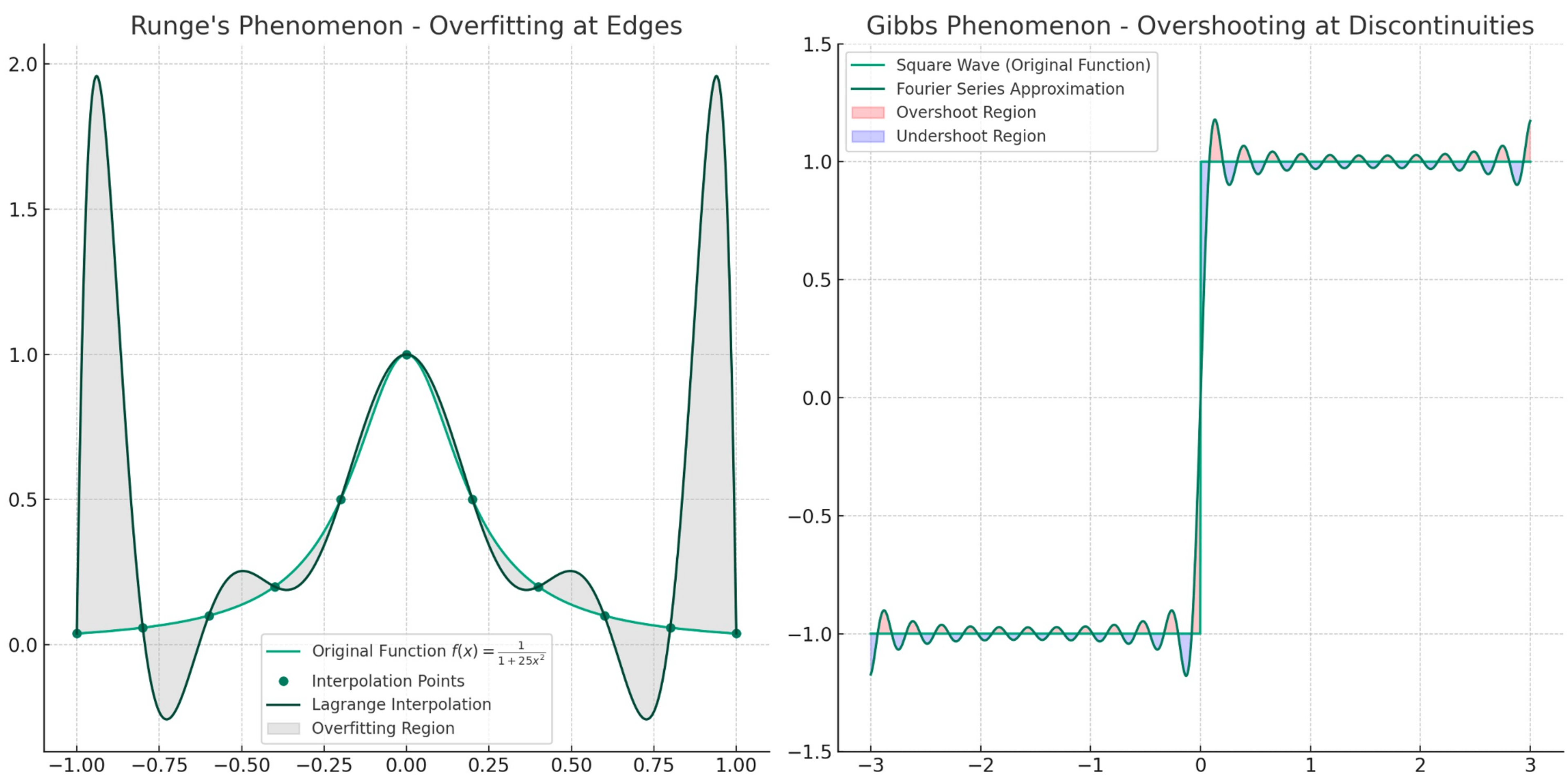


n = 10



Comparison with Polynomial Interpolation

Aspect	**Trigonometric Interpolation**	**Polynomial Interpolation**
Approach	Uses sine and cosine functions to approximate data.	Uses polynomial functions of varying degrees for approximation.
Suited For	Periodic or oscillatory data.	Non-periodic, scattered data or when a simple curve fit is needed.
Strengths	Produces smooth, wave-like curves that repeat over intervals.	Versatile for many general interpolation tasks. Can flexibly fit different data shapes.
LImitations	Not well-suited for non-periodic data. May suffer from Gibbs phenomenon near discontinuities.	High-degree polynomials can lead to overfitting and erratic behavior, particularly at data edges (Runge's phenomenon). Not inherently suited for periodic data.



Runge's Phenomenon - Overfitting at Edges
Original Function $f(x) = \frac{1}{1+25x^2}$
Interpolation Points
Lagrange Interpolation
Overfitting Region
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
0.0
0.5
1.0
1.5
2.0
Gibbs Phenomenon - Overshooting at Discontinuities
Square Wave (Original Function)
Fourier Series Approximation
Overshoot Region
Undershoot Region
−3
−2
−1
0
1
2
3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Take-away Message

Trigonometric Interpolation

- Ideal for periodic data
  - providing smooth and continuous representations.
- It visualizes cyclical trends effectively but may not be suitable for all types of data.

Polynomial Interpolation

- Offers a flexible approach for various data types but requires careful selection of the polynomial degree.
- It's not inherently suited for cyclic patterns but is excellent for general curve fitting.

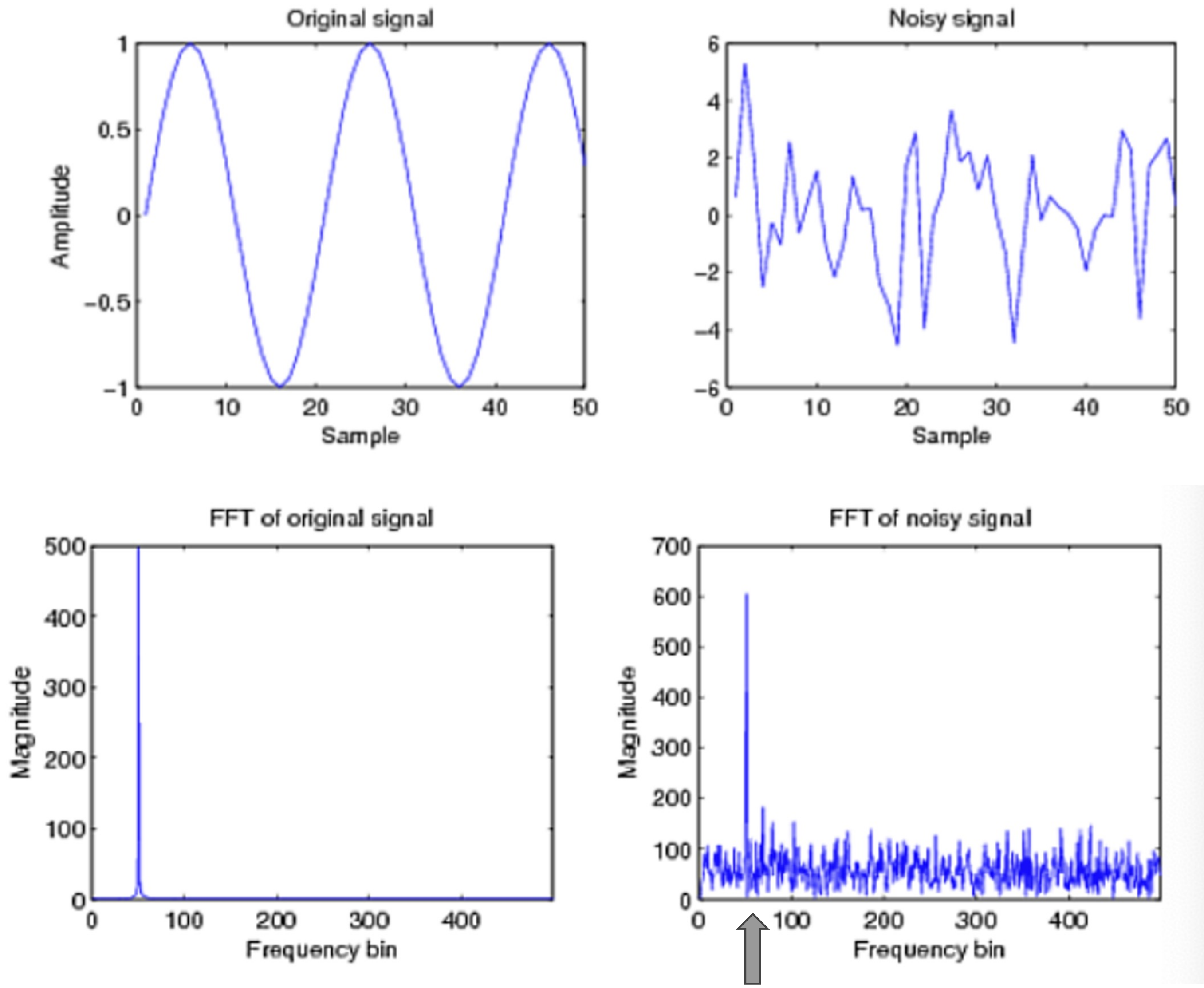
Application

- **Audio Filtering**
- **Digital Image Processing**
- **Modern Mobile Communications**

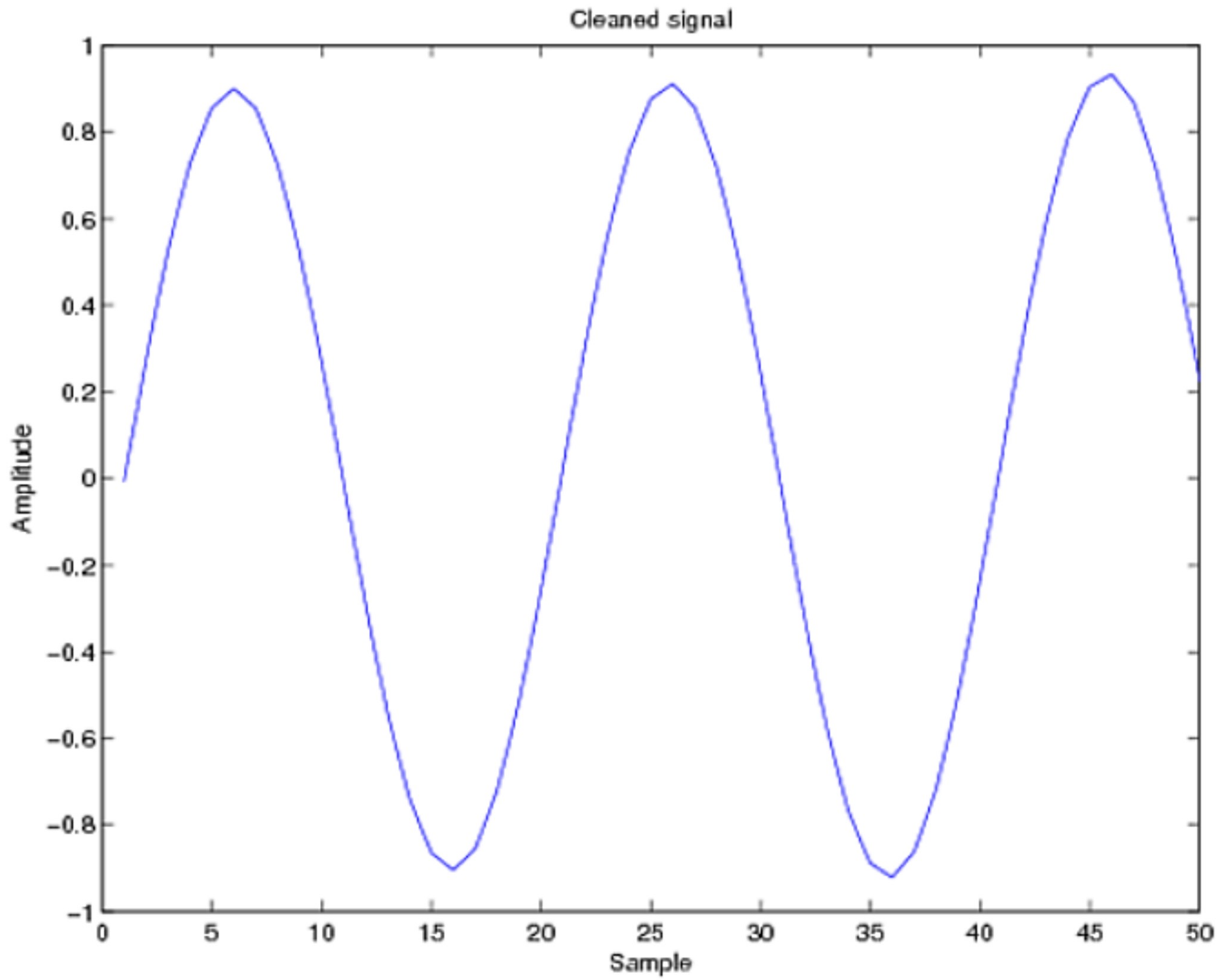
DFT and FFT play a pivotal role in the computation of interpolation coefficients, showcasing their utility in theoretical applications. Also in a myriad of practical real-life scenarios, FFT is helpful.

Application in Audio Filtering

- remove unwanted noise from a signal
- it is possible to identify and isolate the noise components
  - And then selectively filter them out before converting the signal back to the time domain.
- This is a common technique used in audio restoration, signal analysis, and other digital signal processing tasks.

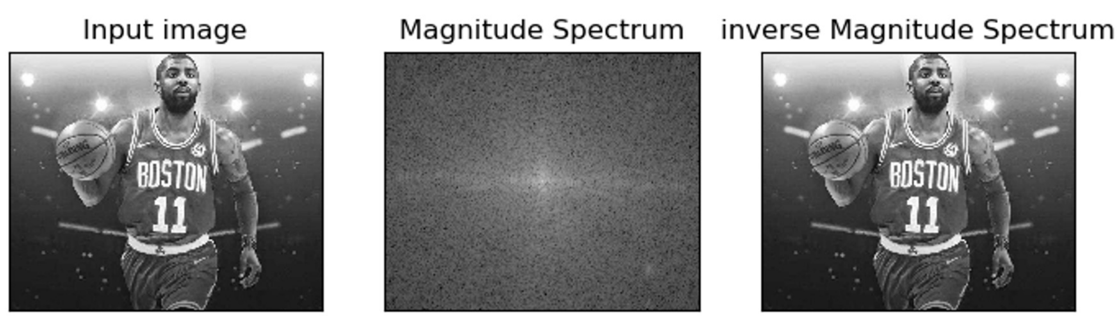


Filtering the noisy signal with a narrow band filter from 48 to 52 Hz, gives us a "cleaned" signal.


Cleaned signal
Amplitude
Sample
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
5
10
15
20
25
30
35
40
45
50

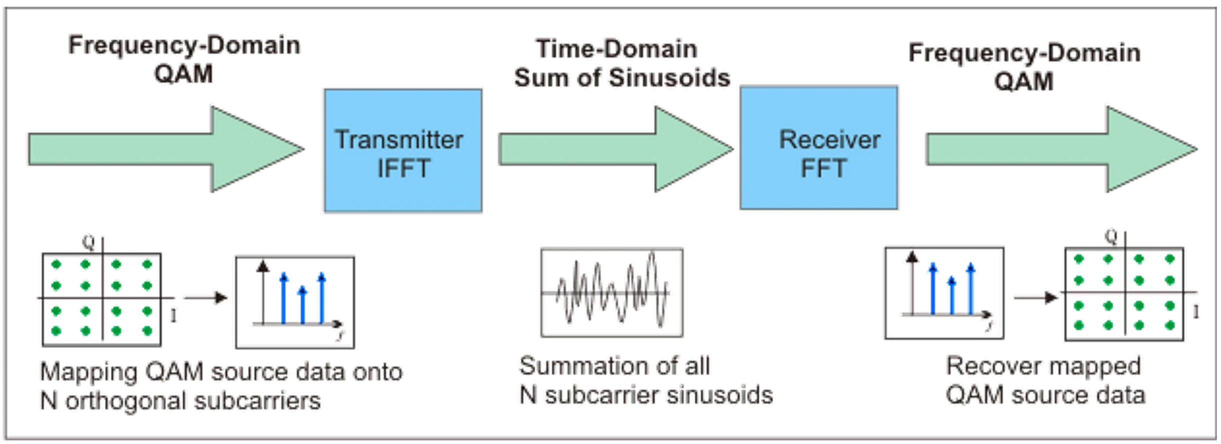
Application in Digital Image Processing

- Frequency Domain Conversion
  - 2nd image: the bright center represents low-frequency information, and the edges represent high-frequency information
- Data Compression and Noise Reduction
- Information Recovery
  - Through the inverse FFT (the third image), the image can be transformed back from the frequency domain to the spatial domain, thus recovering the original image.



Application in Modern Mobile Communications

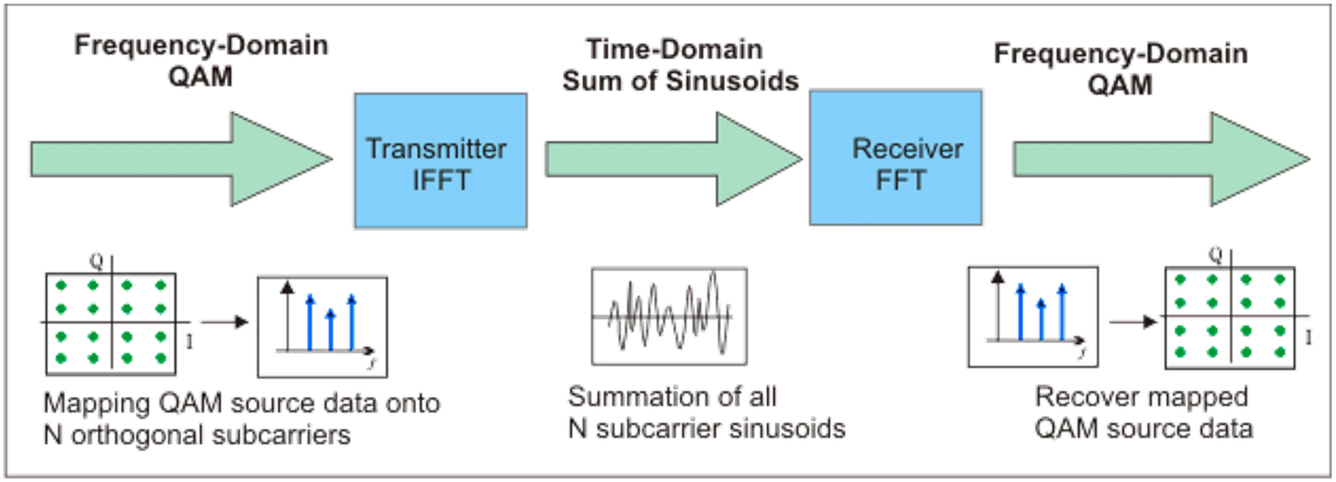
- Driving Efficiency in 4G and 5G Networks
- Role in 4G LTE
    - Essential for OFDM (Orthogonal Frequency Division Multiplexing).
    - Splits one high-speed data channel into multiple slower subchannels.
    - Reduces interference and improves signal quality.



Simplified OFDM System Block Diagram

Application in Modern Mobile Communications

- Enhancements in 5G
  - Supports OFDMA (Orthogonal Frequency Division Multiple Access).
  - Enables flexible and efficient use of wider frequency bands.
  - Facilitates high-speed data rates and low-latency communications.



Simplified OFDM System Block Diagram

Thank you for listening!

Please vote for us 🙏🙏🙏🙏

```
        🌸🌼🌷🌼🌸
          🌺🌷🌷🌺
 ∧ ∧    🌸🌺🌸
( ˘ ᴗ ˘   🌷🌸🌷
 ヽ  つ    \\ | //
  UU     / 🎀 \
```