CS711008Z Algorithm Design and Analysis

Lecture 5. FFT and Divide-and-Conquer [1]

Dongbo Bu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1] The slides are prepared based on Lecture 35 of The Design and Analysis of Algorithms (by D. C. Kozen), Mathematical methods for physics (by Qiao Gu), and Chapter 5 of Algorithm Design (by J. Kleinburg and E. Tardos).

Outline

- DFT: evaluate a polynomial at n special points;
- FFT: an efficient implementation of DFT;
- Applications of FFT: multiplying two polynomials (and multiplying two n-bits integers); time-frequency transform; solving partial differential equations;
- Appendix: relationship between continuous and discrete Fourier transforms.

DFT: Discrete Fourier Transform

- DFT evaluates a polynomial $A(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$ at n distinct points $1, \omega, \omega^2, ..., \omega^{n-1}$, where $\omega = e^{-\frac{2\pi}{n}i}$ is the n-th complex root of unity.
- Thus, it transforms the complex vector $a_0, a_1, ..., a_{n-1}$ into another complex vector $y_0, y_1, ..., y_{n-1}$, where $y_i = A(w^i)$, i.e.,

$$
\begin{array}{lllll}
y_0 = a_0 & + a_1 & + a_2 & \dots & + a_{n-1} \\
y_1 = a_0 & + a_1\omega^1 & + a_2\omega^2 & \dots & + a_{n-1}\omega^{n-1} \\
\dots & \dots & \dots & \dots & \dots \\
y_{n-1} = a_0 & + a_1\omega^{n-1} & + a_2\omega^{2(n-1)} & \dots & + a_{n-1}\omega^{(n-1)^2}
\end{array}
$$

- Matrix form:

$$
\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}
$$

FFT: a fast way to implement DFT [Cooley-Tukey 1965]

- Direct matrix-vector multiplication requires $O(n^2)$ operations when using the Horner's method, i.e., $A(x) = a_0 + x(a_1 + x(a_2 + \ldots + xa_{n-1}))$.
- FFT: reduce $O(n^2)$ to $O(n \log_2 n)$ using divide-and-conqueror technique.
- How does FFT achieve this? Or what calculations are redundant in the direct matrix-vector multiplication approach?
- Note: The idea of FFT was proposed by Cooley and Tukey in 1965 when analyzing earth-quake data, but the idea can be dated back to F. Gauss.

Let's evaluate $A(x)$ at two special points first

- Consider evaluating a 7-degree polynomial $A(x) = a_0 + a_1 x + a_2 x^2 + ... + a_7 x^7$ at two special points $1, -1$.
- **Divide:** Break the polynomial into even and odd terms, i.e.,
  - $A_{even}(x) = a_0 + a_2 x + a_4 x^2 + a_6 x^3$
  - $A_{odd}(x) = a_1 + a_3 x + a_5 x^2 + a_7 x^3$

  Then we have the following equations:
  - $A(x) = A_{even}(x^2) + x A_{odd}(x^2)$
  - $A(-x) = A_{even}(x^2) - x A_{odd}(x^2)$
- **Combine:** For two special points $1, -1$, we have
  - $A(1) = A_{even}(1) + A_{odd}(1)$
  - $A(-1) = A_{even}(1) - A_{odd}(1)$
- In other words, the values of $A(x)$ at **2 points** $1, -1$ can be calculated based on the values of $A_{even}(x), A_{odd}(x)$ at only **1 point** .

Let's evaluate $A(x)$ at four special points further

- Consider evaluating a 7-degree polynomial $A(x) = a_0 + a_1 x + a_2 x^2 + ... + a_7 x^7$ at four special points $1, -i, -1, i$.
- **Divide:** Break the polynomial into even and odd terms, i.e.,
  - $A_{even}(x) = a_0 + a_2 x + a_4 x^2 + a_6 x^3$
  - $A_{odd}(x) = a_1 + a_3 x + a_5 x^2 + a_7 x^3$

  Then we have the following equations:
  - $A(x) = A_{even}(x^2) + x A_{odd}(x^2)$
  - $A(-x) = A_{even}(x^2) - x A_{odd}(x^2)$
- **Combine:** For 4 special points $1, -i, i, -1$, we have
  - $A(1) = A_{even}(1) + A_{odd}(1)$
  - $A(-i) = A_{even}(-1) - i A_{odd}(-1)$
  - $A(-1) = A_{even}(1) - A_{odd}(1)$
  - $A(i) = A_{even}(-1) + i A_{odd}(-1)$
- In other words, the values of $A(x)$ at **4 points** $1, -i, -1, i$ can be calculated based on the values of $A_{even}(x), A_{odd}(x)$ at **2 points** $1, -1$.

FFT Algorithm

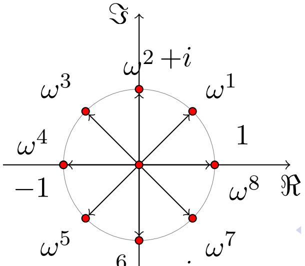
$\mathrm{FFT}(n, a_0, a_1, ..., a_{n-1})$

if $n == 1$ **then**
    return a_0 ;
end if
$(E_0, E_1, ..., E_{\frac{n}{2}-1}) = \mathrm{FFT}(\frac{n}{2}, a_0, a_2, ..., a_n);$
$(O_0, O_1, ..., O_{\frac{n}{2}-1}) = \mathrm{FFT}(\frac{n}{2}, a_1, a_3, ..., a_{n-1});$
for $k = 0$ to $\frac{n}{2} - 1$ **do**
    $\omega^k = e^{\frac{2\pi}{n}ki};$
    $y_k = E_k + \omega^k O_k;$
    $y_{\frac{n}{2}+k} = E_k - \omega^k O_k;$
end for
return $(y_0, y_1, ..., y_{n-1})$;

An example: $n = 8$

$$
\begin{array}{llllllll}
y_0 = a_0 & + a_1 & + a_2 & + a_3 & + a_4 & + a_5 & + a_6 & + a_7 \\
y_1 = a_0 & + a_1\omega^1 & + a_2\omega^2 & + a_3\omega^3 & + a_4\omega^4 & + a_5\omega^5 & + a_6\omega^6 & + a_7\omega^7 \\
y_2 = a_0 & + a_1\omega^2 & + a_2\omega^4 & + a_3\omega^6 & + a_4\omega^8 & + a_5\omega^{10} & + a_6\omega^{12} & + a_7\omega^{14} \\
y_3 = a_0 & + a_1\omega^3 & + a_2\omega^6 & + a_3\omega^9 & + a_4\omega^{12} & + a_5\omega^{15} & + a_6\omega^{18} & + a_7\omega^{21} \\
y_4 = a_0 & + a_1\omega^4 & + a_2 & + a_3\omega^{12} & + a_4\omega^{16} & + a_5\omega^{20} & + a_6\omega^{24} & + a_7\omega^{28} \\
y_5 = a_0 & + a_1\omega^5 & + a_2\omega^{10} & + a_3\omega^{15} & + a_4\omega^{20} & + a_5\omega^{25} & + a_6\omega^{30} & + a_7\omega^{35} \\
y_6 = a_0 & + a_1\omega^6 & + a_2\omega^{12} & + a_3\omega^{18} & + a_4\omega^{24} & + a_5\omega^{30} & + a_6\omega^{36} & + a_7\omega^{42} \\
y_7 = a_0 & + a_1\omega^7 & + a_2\omega^{14} & + a_3\omega^{21} & + a_4\omega^{28} & + a_5\omega^{35} & + a_6\omega^{42} & + a_7\omega^{49}
\end{array}
$$

- Objective: Evaluate $A(x)$ at 8 points: $1, \omega, \omega^2, ..., \omega^7$, where $\omega = e^{\frac{1}{8}2\pi}i$.



Step 1: Simplification

$$
\begin{array}{llllllll}
y_0 = a_0 & + a_1 & + a_2 & + a_3 & + a_4 & + a_5 & + a_6 & + a_7 \\
y_1 = a_0 & + a_1\omega^1 & + a_2\omega^2 & + a_3\omega^3 & + a_4\omega^4 & + a_5\omega^5 & + a_6\omega^6 & + a_7\omega^7 \\
y_2 = a_0 & + a_1\omega^2 & + a_2\omega^4 & + a_3\omega^6 & + a_4 & + a_5\omega^2 & + a_6\omega^4 & + a_7\omega^6 \\
y_3 = a_0 & + a_1\omega^3 & + a_2\omega^6 & + a_3\omega^1 & + a_4\omega^4 & + a_5\omega^7 & + a_6\omega^2 & + a_7\omega^5 \\
y_4 = a_0 & + a_1\omega^4 & + a_2\omega^8 & + a_3\omega^4 & + a_4 & + a_5\omega^4 & + a_6 & + a_7\omega^4 \\
y_5 = a_0 & + a_1\omega^5 & + a_2\omega^2 & + a_3\omega^7 & + a_4\omega^4 & + a_5\omega^1 & + a_6\omega^6 & + a_7\omega^3 \\
y_6 = a_0 & + a_1\omega^6 & + a_2\omega^4 & + a_3\omega^2 & + a_4 & + a_5\omega^6 & + a_6\omega^4 & + a_7\omega^2 \\
y_7 = a_0 & + a_1\omega^7 & + a_2\omega^6 & + a_3\omega^5 & + a_4\omega^4 & + a_5\omega^3 & + a_6\omega^2 & + a_7\omega^1
\end{array}
$$

Step 2. Divide into odd- and even-terms

$$
\begin{aligned}
y_0 &= a_0 &+ a_4 &+ a_2 &+ a_6 &+ a_1 &+ a_5 &+ a_3 &+ a_7 \\
y_1 &= a_0 &+ a_4\omega^4 &+ a_2\omega^2 &+ a_6\omega^6 &+ a_1\omega^1 &+ a_5\omega^5 &+ a_3\omega^3 &+ a_7\omega^7 \\
y_2 &= a_0 &+ a_4 &+ a_2\omega^4 &+ a_6\omega^4 &+ a_1\omega^2 &+ a_5\omega^2 &+ a_3\omega^6 &+ a_7\omega^6 \\
y_3 &= a_0 &+ a_4\omega^4 &+ a_2\omega^6 &+ a_6\omega^2 &+ a_1\omega^3 &+ a_5\omega^7 &+ a_3\omega^1 &+ a_7\omega^5 \\
y_4 &= a_0 &+ a_4 &+ a_2 &+ a_6 &+ a_1\omega^4 &+ a_5\omega^4 &+ a_3\omega^4 &+ a_7\omega^4 \\
y_5 &= a_0 &+ a_4\omega^4 &+ a_2\omega^2 &+ a_6\omega^6 &+ a_1\omega^5 &+ a_5\omega^1 &+ a_3\omega^7 &+ a_7\omega^3 \\
y_6 &= a_0 &+ a_4 &+ a_2\omega^4 &+ a_6\omega^4 &+ a_1\omega^6 &+ a_5\omega^6 &+ a_3\omega^2 &+ a_7\omega^2 \\
y_7 &= a_0 &+ a_4\omega^4 &+ a_2\omega^6 &+ a_6\omega^2 &+ a_1\omega^7 &+ a_5\omega^3 &+ a_3\omega^5 &+ a_7\omega^1
\end{aligned}
$$

The specific order of these terms will be explained later.

Key observation: redundant calculations

$$
\begin{array}{lllll|llll}
y_0 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1 & + a_5 & + a_3 & + a_7 \\
y_1 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^1 & + a_5\omega^5 & + a_3\omega^3 & + a_7\omega^7 \\
y_2 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^2 & + a_5\omega^2 & + a_3\omega^6 & + a_7\omega^6 \\
y_3 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^3 & + a_5\omega^7 & + a_3\omega^1 & + a_7\omega^5 \\
y_4 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1\omega^4 & + a_5\omega^4 & + a_3\omega^4 & + a_7\omega^4 \\
y_5 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^5 & + a_5\omega^1 & + a_3\omega^7 & + a_7\omega^3 \\
y_6 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^6 & + a_5\omega^6 & + a_3\omega^2 & + a_7\omega^2 \\
y_7 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^7 & + a_5\omega^3 & + a_3\omega^5 & + a_7\omega^1
\end{array}
$$

Note: the calculations in the two red frames are the same; and the two calculations in the blue frames are also identical after multiplying by ω^4.

Step 3: Divide-and-conquer

$$
\begin{array}{lllllllll}
y_0 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1 & + a_5 & + a_3 & + a_7 \\
y_1 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^1 & + a_5\omega^5 & + a_3\omega^3 & + a_7\omega^7 \\
y_2 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^2 & + a_5\omega^2 & + a_3\omega^6 & + a_7\omega^6 \\
y_3 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^3 & + a_5\omega^7 & + a_3\omega^1 & + a_7\omega^5 \\
y_4 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1\omega^4 & + a_5\omega^4 & + a_3\omega^4 & + a_7\omega^4 \\
y_5 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^5 & + a_5\omega^1 & + a_3\omega^7 & + a_7\omega^3 \\
y_6 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^6 & + a_5\omega^6 & + a_3\omega^2 & + a_7\omega^2 \\
y_7 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^7 & + a_5\omega^3 & + a_3\omega^5 & + a_7\omega^1
\end{array}
$$

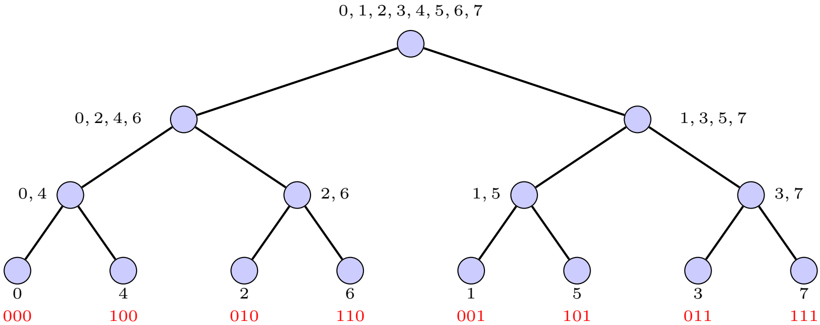
Note: the calculations in the top-left and bottom-right frames are redundant.

Step 3: divide-and-conquer

$$
\begin{array}{llllllllll}
y_0 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1 & + a_5 & + a_3 & + a_7 \\
y_1 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^1 & + a_5\omega^5 & + a_3\omega^3 & + a_7\omega^7 \\
y_2 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^2 & + a_5\omega^2 & + a_3\omega^6 & + a_7\omega^6 \\
y_3 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^3 & + a_5\omega^7 & + a_3\omega^1 & + a_7\omega^5 \\
y_4 = & a_0 & + a_4 & + a_2 & + a_6 & + a_1\omega^4 & + a_5\omega^4 & + a_3\omega^4 & + a_7\omega^4 \\
y_5 = & a_0 & + a_4\omega^4 & + a_2\omega^2 & + a_6\omega^6 & + a_1\omega^5 & + a_5\omega^1 & + a_3\omega^7 & + a_7\omega^3 \\
y_6 = & a_0 & + a_4 & + a_2\omega^4 & + a_6\omega^4 & + a_1\omega^6 & + a_5\omega^6 & + a_3\omega^2 & + a_7\omega^2 \\
y_7 = & a_0 & + a_4\omega^4 & + a_2\omega^6 & + a_6\omega^2 & + a_1\omega^7 & + a_5\omega^3 & + a_3\omega^5 & + a_7\omega^1
\end{array}
$$

Finally, we need only $2+4+2+8+2+4+2 = 8 \times \log 8$ calculations.

The final order



Inverse Discrete Fourier Transform

- Inverse Discrete Fourier Transform: to determine coefficients of a polynomial $a_0, a_1, ..., a_{n-1}$ based on n point-value pairs $(1, y_0), (\omega, y_1), ..., (\omega^{n-1}, y_{n-1})$, where $y_i = A(\omega^i)$, and $A(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1} x^{n-1}$.
- Matrix form

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)^2} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix}$$

- It takes $O(n^3)$ to calculate the inverse matrix when using the Gaussian elimination technique.

Inverse Discrete Fourier Transform cont'd

- Matrix form

$$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \end{bmatrix} = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \bar{\omega}^1 & \bar{\omega}^2 & \dots & \bar{\omega}^{n-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \dots & \bar{\omega}^{2(n-1)} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \bar{\omega}^{n-1} & \bar{\omega}^{2(n-1)} & \dots & \bar{\omega}^{(n-1)^2} \end{bmatrix} \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{bmatrix}$$

- Reason: it turns out that it is nearly its own inverse. More precisely, the conjugate transpose of this matrix is its own inverse.

IFFT Algorithm

$\mathrm{IFFT}(n, y_0, y_1, ..., y_{n-1})$

if $n == 1$ **then**

  return y_0 ;

end if

$(E_0, E_1, ..., E_{\frac{n}{2}-1}) = \mathrm{IFFT}(\frac{n}{2}, y_0, y_2, ..., y_n);$

$(O_0, O_1, ..., O_{\frac{n}{2}-1}) = \mathrm{IFFT}(\frac{n}{2}, y_1, y_3, ..., y_{n-1});$

for $k = 0$ to $\frac{n}{2} - 1$ **do**

  $\omega^k = e^{-\frac{2\pi}{n}ki};$

  $a_k = E_k + \omega^k O_k;$

  $a_{\frac{n}{2}+k} = E_k - \omega^k O_k;$

end for

return $\frac{1}{n}(a_0, a_1, ..., a_{n-1})$;

Note: here we assume n is the power of 2 for simplicity. The normalization factors multiplying FFT and IFFT (here 1 and $\frac{1}{n}$) and the signs of exponents are merely conventions, and differ in some treatments.

Application: fast multiplication of two polynomials (or two integers)

Multiply two polynomials: convolution

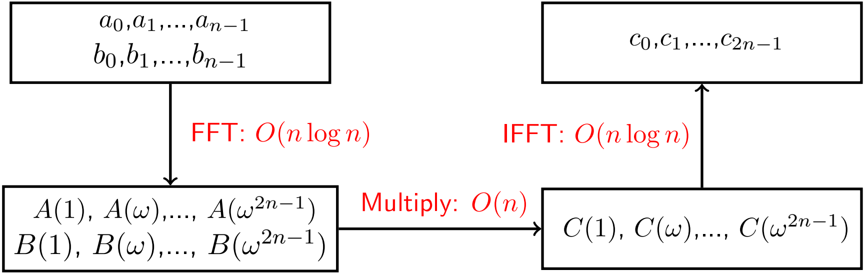
- Given two polynomials
  $A(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{n-1} x^{n-1}$, and
  $B(x) = b_0 + b_1 x + b_2 x^2 + ... + b_{n-1} x^{n-1}$
- Let's calculate its product
  $C(x) = A(x)B(x) = c_0 + c_1 x + c_2 x^2 + ... + c_{2n-2} x^{2n-2}$
- Brute-force (convolution): $c_k = \sum_{i=0}^{k} a_i b_{k-i}$.
- It costs $O(n^2)$ time if using the convolution technique.

Conversion between two representations of polynomials

- An efficient conversion between these two representations is extremely useful when multiplying two polynomials.

$$\boxed{a_0, a_1, \ldots, a_{n-1}} \rightleftarrows \boxed{(x_0, y_0), \ldots, (x_{n-1}, y_{n-1})}$$

- Given two polynomials
  $A(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{n-1} x^{n-1}$, and
  $B(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_{n-1} x^{n-1}$
- Let's calculate its product
  $C(x) = A(x)B(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{2n-2} x^{2n-2}$
- Brute-force: $c_k = \sum_{i=0}^{k} a_i b_{k-i}$. Cost $O(n^2)$ time
- Using FFT and IFFT: $O(n \log n)$



An example

- $A(x) = 1 + 2x$
- $B(x) = 3 + 4x$
- $C(x) = A(x)B(x) = c_0 + c_1x + c_2x^2 + c_3x^3$

x	1	$-i$	-1	i
$A(x)$	3	$1-2i$	-1	$1+2i$
$B(x)$	7	$3-4i$	-1	$3+4i$
$C(x)$	21	$-5-10i$	1	$-5+10i$

- By running $\text{IFFT}(4, (21, -5-10i, 1, -5+10i))$, we obtained the coefficients as $c_0 = 3, c_1 = 10, c_2 = 8$, and $c_3 = 0$.
- Extension: given two n-bit integers $a = a_{n-1}...a_1a_0$, and $b = b_{n-1}...b_1b_0$, it takes $O(n \log n)$ complex arithmetic steps to calculate $c = a \times b$.

Application: time-frequency transform

- DFT, denoted as $\mathbf{X} = \mathcal{F}\{\mathbf{x}\}$, transforms a sequence of N complex numbers $x_0, x_1, ..., x_{N-1}$ (time-domain) into a N-periodic sequence of complex numbers $X_0, X_1, ..., X_{N-1}$ (frequency-domain):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{i} Nkn}, \quad k = 0, 1, ..., N-1$$

- Here, X_k encodes both amplitude and phase of a sinusoidal component $e^{-\frac{2\pi}{N} kni}$ of the function x_n (the sinusoid's frequency is k **cycles per** N **samples**).
- **Inverse transform** of DFT:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi}{i} Nkn}$$

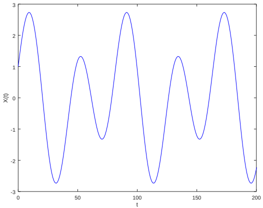
An interpretation of DFT is that its inverse transform is the **discrete analogy** of the formula for **a Fourier series**:

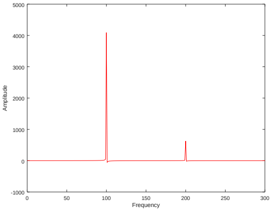
$$f(x) = \sum_{n=-\infty}^{+\infty} F_n e^{inx}, \;\; F_n = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-inx} \mathrm{d}x$$

DFT: an example

- An example:

```
Fs = 8192;
t=0:1/Fs:1;
x = 1*cos(2*pi*100*t) + 2*sin(2*pi*200*t);
N = length(x);
Freq = (0:N-1)*Fs/N;
plot( Freq, abs(fft(x)) );
```





Appendix: Relationship between continuous and discrete Fourier transforms

Fourier series, Fourier transform, DTFT, and DFT

- Fourier series decomposes a periodic function into a set of sine/cosine waves, and one of the motivations of Fourier transform comes from the extension of Fourier series to non-periodic functions.
- DTFT uses discrete-time samples of a continuous function as input, and generates a continuous function of frequency.
- Using a finite sequence of equally-spaced samples of a function as input, DFT computes a sequence of identical length, representing equally-spaced samples of DTFT. The interval at which the DTFT is sampled is reciprocal of the duration of the input sequence.
- The inverse DFT is a Fourier series using the DTFT samples as coefficients of corresponding frequency, and it is essentially a periodic summation of the original input sequence.

Figure: Jean-Baptiste Joseph Fourier (1768-1830)

- In 1807, Joseph Fourier proposed the idea of Fourier series when solving heat equation, a partial differential equation.
- Prior to Fourier's work, no solution to heat equation was known in the general case. However, when the heat source was a simple sine or cosine wave, solutions were known (called eigensolutions).
- Thus, Fourier modelled complicated heat source as a superposition of simple sine/cosine waves, and rewrote the solution as superposition of corresponding eigensolutions.

Fourier series

- Fourier series is a way to represent a **periodic function** of time as the sum of a set of simple sines and cosines (or, equivalently, complex exponentials). For example, the Fourier series of a periodic function $f(x)$ (period 2π) is:

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx)$$

where

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(t) \mathrm{d}t$$

$$a_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \cos nt \mathrm{d}t \quad (n = 1, 2, ...)$$

$$b_n = \frac{1}{\pi} \int_0^{2\pi} f(t) \sin nt \mathrm{d}t \quad (n = 1, 2, ...)$$

Fourier series: orthogonality of basis functions

- Unlike Taylor's expansion, the basis functions of Fourier series are orthogonal over $[0, 2\pi]$, i.e.,

$$\int_0^{2\pi} 1 \cdot \sin x \mathrm{d}x = 0, \quad \int_0^{2\pi} 1 \cdot \cos x \mathrm{d}x = 0$$

$$\int_0^{2\pi} \sin mx \cdot \sin nx \mathrm{d}x = 0, \quad \int_0^{2\pi} \cos mx \cdot \cos nx \mathrm{d}x = 0 \quad (m \neq n)$$

$$\int_0^{2\pi} \cos mx \cdot \sin nx \mathrm{d}x = 0$$

- The orthogonality plays an important role in solving coefficients a_0, a_n, b_n.

Fourier series: complex exponential form

- According to the Euler's formula $e^{ix} = \cos x + i \sin x$, we have $\cos x = \frac{1}{2}(e^{ix} + e^{-ix})$, $\sin x = \frac{1}{2i}(e^{ix} - e^{-ix})$, and

$$
\begin{aligned}
f(x) &= a_0 + \sum_{n=1}^{\infty}(a_n \cos nx + b_n \sin nx) \\
&= a_0 + \sum_{n=1}^{\infty}(a_n \frac{1}{2}(e^{ix} + e^{-ix}) + b_n \frac{1}{2i}(e^{ix} - e^{-ix})) \\
&= a_0 + \sum_{n=1}^{\infty}(\frac{1}{2}(a_n - ib_n)e^{ix} + \frac{1}{2}(a_n + ib_n)e^{-ix})
\end{aligned}
$$

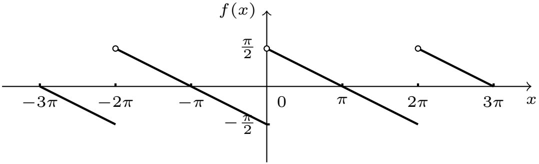
- Define $F_0 = a_0$, and $F_n = \frac{1}{2}(a_n - ib_n)$ $(n > 0)$. We have $F_{-n} = \frac{1}{2}(a_n + ib_n)$, and thus rewrite the Fourier series as:

$$
f(x) = \sum_{n=-\infty}^{+\infty} F_n e^{inx}, F_n = \frac{1}{2}(a_n - ib_n) = \frac{1}{2\pi}\int_0^{2\pi} f(t)e^{-int}\mathrm{d}t
$$

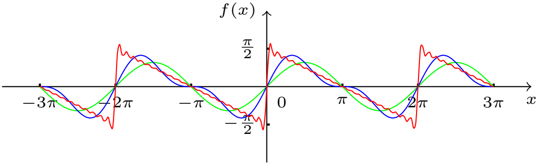
- Complex exponential form is necessary as the complex coefficients F_n (called *frequency spectrum* could encode both amplitude and phase of basic waves.

Fourier series: example 1

- Periodic function $f(x) = \begin{cases} \frac{1}{2}(\pi - x) & 0 < x \le 2\pi \\ f(x+2\pi) & \text{otherwise} \end{cases}$



- Fourier series: $f(x) = \sum_{n=1}^{\infty} \frac{1}{n} \sin nx$ (since $a_n = 0$, $b_n = \frac{1}{n}$)



Fourier series: extension to $f(x)$ with period of $2L$

- For a periodic function $f(x)$ with period of $2L$, the Fourier series is:

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos \frac{\pi}{L} nx + b_n \sin \frac{\pi}{L} nx)$$

- The coefficients are:

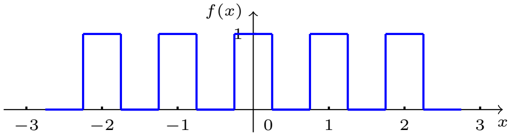
$$a_0 = \frac{1}{2L} \int_0^{2L} f(t) \mathrm{d}t$$

$$a_n = \frac{1}{L} \int_0^{2L} f(t) \cos \frac{\pi}{L} nt \mathrm{d}t$$

$$b_n = \frac{1}{L} \int_0^{2L} f(t) \sin \frac{\pi}{L} nt \mathrm{d}t$$

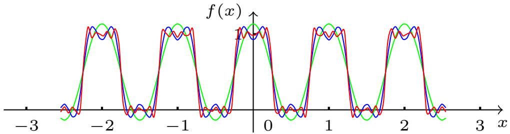
Fourier series: example 2

- Periodic function $f(x) = \begin{cases} 1 & |x| \le \frac{1}{4} \\ 0 & \frac{1}{4} < |x| \le \frac{T}{2} \end{cases}$, and $f(x)$ has a period $T = 1$.



- Fourier series:

$$f(x) = \frac{1}{2T} + \frac{2}{\pi} \sum_{n=1}^{\infty} \frac{1}{n} \sin(\frac{\pi}{2T} n) \cos(\frac{2\pi}{T} nx)$$



Convergence of Fourier series: Dirichlet's conditions

- Dirichlet's theorem states the sufficient conditions for the convergence of Fourier series, i.e., if $f(x)$ satisfies the following conditions:
  1. $f(x)$ is periodic, and absolutely integrable over a period;
  2. $f(x)$ must have a finite number of maxima and minima in any bounded interval;
  3. $f(x)$ must have a finite number of discontinuities in any bounded interval, and the discontinuity cannot be infinite.

  Then

  $$a_0 + \sum_{n=1}^{m} (a_n \cos nx + b_n \sin nx) \to \frac{1}{2}(f(x+0) + f(x-0))$$

  when $m \to \infty$.
- A succinct proof using Dirac's δ function can be found in *Mathematical Methods for Physics* (by Q. Gu).

Proof.

- Since $a_n \cos nx + b_n \sin nx = \frac{1}{\pi}\int_{-\pi}^{\pi} f(t)\cos n(x-t)\mathrm{d}t$, the partial sum of Fourier series is:

$$\begin{aligned} S_m(x) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} f(t)[1+2\sum_{n=1}^{m}\cos n(x-t)]\mathrm{d}t \\ &= \int_{-\pi}^{\pi} f(t)\frac{\sin((m+\frac{1}{2})(x-t))}{2\pi\sin\frac{1}{2}(x-t)}\mathrm{d}t \\ &= \int_{-\pi}^{\pi} f(t)D_m(x-t)\mathrm{d}t \end{aligned}$$

- Here $D_m(x) = \frac{1}{2\pi}(1+2\cos x+2\cos 2x+\ldots+2\cos mx)$.
- Note that $\lim_{m\to\infty} D_m(x) = \delta(x)$ since $\int_{-\pi}^{\pi} D_m(x)\mathrm{d}x = 1$ and $D_m(0) = \frac{1}{2\pi}(2m+1)\to\infty$.
- Thus, we have $\lim\limits_{m\to\infty} S_m(x) = \int_{-\pi}^{\pi} f(t)\delta(x-t) = f(x)$ (when $f(x)$ is continous at x). Please refer to *Mathematical Methods for Physics* (by Q. Gu) for complete proof.

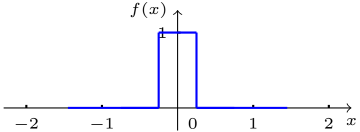
Fourier transform (in terms of angular frequency ω)

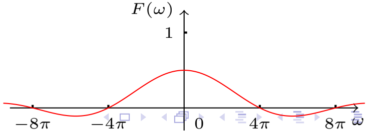
- Fourier transform of a function of time (a *signal*) is a complex-valued function of frequency (represented as *angular frequency* ω), whose absolute value represents the amount of that frequency present in the original function.

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-ix\omega} \mathrm{d}x, \;\; f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} \mathrm{d}\omega$$

- Fourier transform, denoted as $F(\omega) = \mathcal{F}\{f(x)\}$, is called *frequency representation of the original signal*, and $F(\omega)$ is called *spectral density*.
- For example, the Fourier transform of $f(x) = \begin{cases} 1 & |x| \le \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}$ is

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} \mathrm{d}x = \frac{2}{\omega} \sin(\frac{\omega}{4})$$





Fourier transform (in terms of ordinary frequency ν)

- For a sinusoidal wave with period T (measured in *seconds*), its frequency can be measured using angular frequency ω (measured in *radians per second*) or using ordinary frequency ν (measured in *cycles per second*, or hertz), where $\omega = 2\pi\nu$, and $\nu = \frac{1}{\nu}$.
- When using angular frequency ω, Fourier transform is defined as:

$$F(\omega) = \int_{-\infty}^{\infty} f(x) e^{-ix\omega} \mathrm{d}x$$

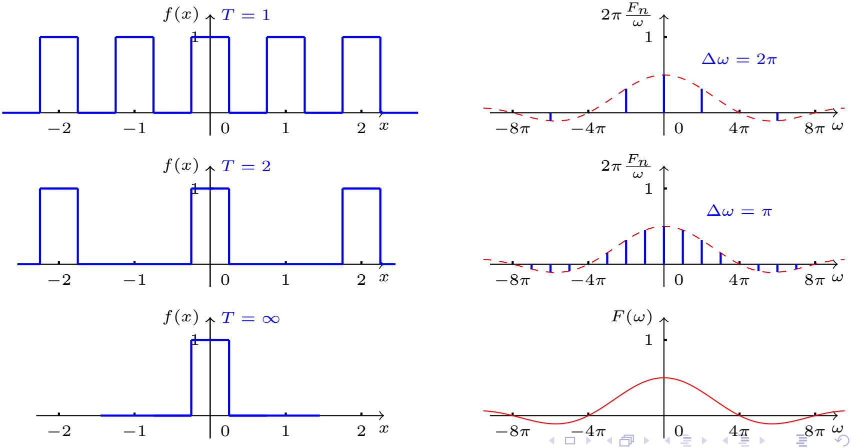
$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{i\omega x} \mathrm{d}\omega$$

- Replacing ω with $\omega = 2\pi\nu$, we obtain another representation of Fourier transform in terms of ordinary frequency ν:

$$F(\nu) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x\nu} \mathrm{d}x$$

$$f(x) = \int_{-\infty}^{\infty} F(\nu) e^{2\pi i x\nu} \mathrm{d}\nu$$

- For a function that are zero outside an interval, we can calculate Fourier series on any larger interval. As we lengthen the interval, the coefficients of Fourier series will approach Fourier transform.



Fourier transform: deduction

- Consider a periodic function $f(x)$ with period $2L$. Its Fourier series $f(x) = a_0 + \sum_{n=1}^{\infty}(a_n \cos \frac{\pi}{L}nx + b_n \sin \frac{\pi}{L}nx)$ can be rewritten as $f(x) = a_0 + \sum_{n=1}^{\infty}(a_n \cos \omega_n x + b_n \sin \omega_n x)$, where $\omega_n = \frac{\pi}{L}n$ represents angular frequency.
- Intuitively, when $L \to \infty$, $f(x)$ becomes a non-periodic function over $(-\infty, \infty)$, and

$$\sum_{n=1}^{\infty} ...\Delta\omega \to \int_0^{\infty} ...\mathrm{d}\omega$$

.

- In particular, we have $a_0 = \frac{1}{2L}\int_{-L}^{L} f(t)\mathrm{d}t \xrightarrow{L\to\infty} 0$ since $f(x)$ is absolutely integrable, and

$$\begin{aligned}
\sum_{n=1}^{\infty} a_n \cos \omega_n x &= \sum_{n=1}^{\infty} \frac{1}{L}\Big[\int_{-L}^{L} f(t)\cos \omega_n t \mathrm{d}t\Big] \cos \omega_n x \\
&= \sum_{n=1}^{\infty} \frac{\Delta\omega}{\pi}\Big[\int_{-L}^{L} f(t)\cos \omega_n t \mathrm{d}t\Big] \cos \omega_n x \\
&\to \int_0^{\infty} \mathrm{d}\omega \Big[\frac{1}{\pi}\int_{-\infty}^{\infty} f(t)\cos \omega t \mathrm{d}t\Big] \cos \omega x
\end{aligned}$$

- Similarly, we have

$$\sum_{n=1}^{\infty} b_n \sin\omega_n x \to \int_0^{\infty} \mathrm{d}\omega \Big[\frac{1}{\pi}\int_{-\infty}^{\infty} f(t)\sin\omega t \mathrm{d}t\Big]\sin\omega x$$

and rewrite Fourier series as:

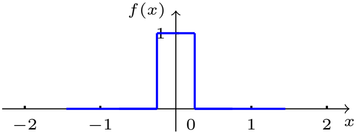
$$\begin{aligned}
f(x) &= \frac{1}{\pi}\int_{\omega=0}^{\infty}\int_{t=-\infty}^{\infty} f(t)(\cos\omega x\cos\omega t + \sin\omega x\sin\omega t)\mathrm{d}t\mathrm{d}\omega \\
&= \frac{1}{\pi}\int_{\omega=0}^{\infty}\int_{t=-\infty}^{\infty} f(t)\cos\omega(x-t)\mathrm{d}t\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int_{\omega=0}^{\infty}\int_{t=-\infty}^{\infty} f(t)(e^{i\omega(x-t)} + e^{-i\omega(x-t)})\mathrm{d}t\mathrm{d}\omega \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\Big[\int_0^{\infty} f(t)e^{i\omega(x-t)}\mathrm{d}\omega + \int_0^{\infty} f(t)e^{-i\omega(x-t)}\mathrm{d}\omega\Big]\mathrm{d}t \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t)e^{i\omega(x-t)}\mathrm{d}\omega\mathrm{d}t \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} F(\omega)\mathrm{d}t
\end{aligned}$$

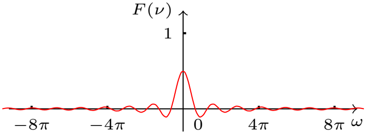
- Linear operations performed in one domain (time or frequency) have corresponding operations in the other domain.
- Differentiation in time domain corresponds to multiplication in the frequency domain, usually making it easier to analyze.
- Convolution in the time domain corresponds to the ordinary multiplication the frequency domain.
- Functions that are localized in one domain have Fourier transforms that are spread out across the other domain, known as the *uncertainty principle*.
- The Fourier transform of a Gaussian function is another Gaussian function.

- For an approximate function $f(x)$ with its Fourier transform (in terms of ordinary frequency) $F(\nu) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \nu} \mathrm{d}x$, the Poisson summation formula states $\sum_{k=-\infty}^{\infty} f(k) = \sum_{k=-\infty}^{\infty} F(k)$.
- For example, the Fourier transform of $f(x) = \begin{cases} 1 & |x| \leq \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}$ is

$$F(\nu) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \nu x} \mathrm{d}x = \frac{1}{\pi\nu} \sin(\frac{\pi\nu}{2})$$

- Poisson summation formula states that $\sum_{k=-\infty}^{\infty} f(k) = 1 = \sum_{k=-\infty}^{\infty} F(k)$.



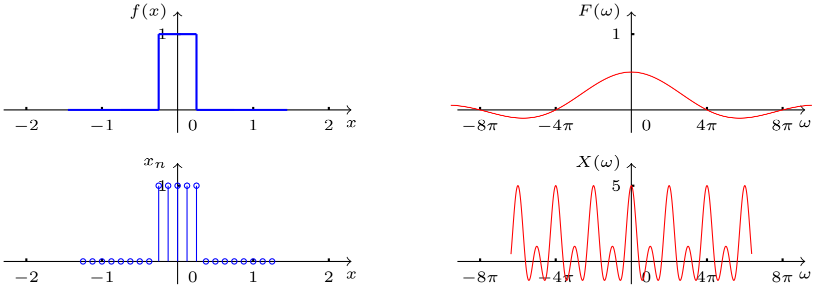


DTFT

- Discrete-time Fourier transform (DTFT) refers to Fourier analysis on the uniformly-spaced samples of a continuous function, i.e., a Fourier series with x_n as coefficients:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-in\omega}$$

Here, the frequency variable ω has normalized units of $radians/sample$.



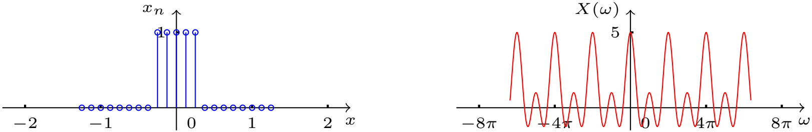
- DTFT is itself a periodic function of frequency $X(\omega)$. From this function, the original samples x_n can be readily recovered as below:

$$x_n = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{in\omega}\mathrm{d}\omega$$

- For example, the DTFT is $X(\omega) = 1 + 2\cos\omega + 2\cos 2\omega$. The original samples can be recovered as:

$$x_0 = \frac{1}{2\pi}\int_{-\infty}^{\infty}(1 + 2\cos\omega + 2\cos 2\omega)\mathrm{d}\omega = 1$$

Similarly, we obtained $x_{-1} = x_1 = x_2 = x_{-2} = 1$.



DTFT and DFT

- From these samples, DTFT produces a function of frequency that is a periodic summation of the Fourier transform of the original continuous function.
- The sampling theorem states the theoretical conditions under which the original function can be perfectly recovered from DTFT of the samples.
- When the input data sequence x_n is N-periodic, DTFT reduces to DFT, i.e.,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi}{N} kni}$$

- Alternatively, DTFT is itself a continuous function, and the discrete samples of it can be efficiently calculated using DFT.

Appendix: Dirac's δ function

- Dirac's δ function has the following two properties:
  1. $\delta(x) = \begin{cases} \infty & x = 0 \\ 0 & \text{otherwise} \end{cases}$
  2. $\int_{-\infty}^{+\infty} \delta(x) \mathrm{d}x = 1$
- We can prove the following properties:
  - For any contineous function $f(x)$,
    $$\int_{-\infty}^{+\infty} f(x)\delta(x - x_0)\mathrm{d}x = f(x_0)$$
    .
  - $\delta(x)$ is the Fourier transform of 1 since
    $$\mathcal{F}\{\delta(x)\} = \int_{-\infty}^{+\infty} \delta(x)e^{-ix\omega}\mathrm{d}x = 1$$
    .
  - According to the inverse Fourier transform of 1, we have:
    $$\delta(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{i\omega x}\mathrm{d}\omega = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega x}\mathrm{d}\omega$$
    .