Numerical Analysis
FMN011

Carmen Arévalo
Lund University
carmen@maths.lth.se

The Fourier Transform

Arévalo FMN011
Complex numbers

- $z = a + bi$ with $i = \sqrt{-1}$
- $|a + bi| = \sqrt{a^2 + b^2}$
- Conjugate: $\bar{z} = a - bi$
- Euler formula: $e^{i\theta} = \cos \theta + i \sin \theta$
- All $z = e^{i\theta}$ lie on the complex unit circle
- Polar representation: $a + bi = re^{i\theta}$, where $r = \sqrt{a^2 + b^2}$ and $\theta = \arctan b/a$
Roots of unity

A complex number $z$ is an $n$th root of unity if $z^n = 1$

An $n$-th root of unity is primitive if $z^k \neq 1$ for $k = 1, 2, 3, \ldots, n - 1$.

Examples:

- $-1$ is a primitive second root of unity and a nonprimitive 4-th root of unity

- $\omega_n = e^{-i2\pi/n}$ is a primitive $n$th root of unity.
Unit circle and 8th roots of unity

\[ \omega_8 = e^{-i \frac{2\pi}{8}} \]

\[ \omega_8^0 = 1 \]

\[ \omega_8^1 = \omega \]

\[ \omega_8^2 \]

\[ \omega_8^3 \]

\[ \omega_8^4 \]

\[ \omega_8^5 \]

\[ \omega_8^6 \]

\[ \omega_8^7 = \pi/4 \]
Properties of primitives roots of unity

Let $\omega$ denote the $n$th root of unity, $\omega = e^{-i2\pi/n}$, $n > 1$.

- For $1 \leq k \leq n-1$, $1 + \omega^k + \omega^{2k} + \omega^{3k} + \cdots + \omega^{(n-1)k} = 0$

- $1 + \omega^n + \omega^{2n} + \omega^{3n} + \cdots + \omega^{(n-1)n} = n$

- $\omega^{-1} = \omega^{n-1}$

- If $n$ is even, $\omega^{n/2} = -1$
Fourier matrix

The DFT of \( x = [x_0, \ldots, x_{n-1}]^T \) is

\[
\begin{pmatrix}
  y_0 \\
  y_1 \\
  y_2 \\
  \vdots \\
  y_{n-1}
\end{pmatrix}
= \frac{1}{\sqrt{n}}
\begin{pmatrix}
  \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\
  \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\
  \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2}
\end{pmatrix}
\begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  \vdots \\
  x_{n-1}
\end{pmatrix}
\]

where \( \omega = e^{-i2\pi/n} \).
Discrete Fourier Transform

\[
y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}
\]

If \( p(t) = x_0 + x_1 t + \cdots + x_{n-1} t^{n-1} \),

\[
\begin{pmatrix}
y_0 \\
y_1 \\
\vdots \\
y_{n-1}
\end{pmatrix} = \frac{1}{\sqrt{n}} \begin{pmatrix}
p(\omega^0) \\
p(\omega^1) \\
\vdots \\
p(\omega^{n-1})
\end{pmatrix}
\]
Inverse Fourier matrix

It is possible to calculate the inverse of the Fourier matrix,

\[ F_n^{-1} = \frac{1}{\sqrt{n}} \begin{pmatrix}
\omega^0 & \omega^0 & \omega^0 & \ldots & \omega^0 \\
\omega^0 & \omega^{-1} & \omega^{-2} & \ldots & \omega^{-(n-1)} \\
\omega^0 & \omega^{-2} & \omega^{-4} & \ldots & \omega^{-2(n-1)} \\
\omega^0 & \omega^{-3} & \omega^{-6} & \ldots & \omega^{-3(n-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \ldots & \omega^{-(n-1)^2}
\end{pmatrix} \]

If \( z = re^{-i\theta} \), its complex conjugate is \( \bar{z} = re^{i\theta} \).

Notice that

\[ F_n^{-1} = \bar{F}_n \]
Unitary matrices

The magnitude of a complex number is

$$\|z\| = \sqrt{\bar{z}^T z}$$

A complex matrix $F$ is unitary if

$$F^{-1} = \bar{F}^T$$

$$\|Fv\|_2 = \sqrt{\bar{v}^T \bar{F}^T Fv} = \sqrt{v^T v} = \|v\|_2.$$  

Note that the Fourier matrix and its inverse are unitary matrices.

If $A$ is a (real) orthogonal matrix, then it is unitary.
Operation count and the DFT in MATLAB

Applying the DFT to a vector of dimension \( n \) requires one square root, one division, one matrix-vector multiplication and one multiplication of a vector by a scalar.

The number of arithmetic operations is

\[
2 + n(2n - 1) + n = 2n^2 + 2
\]

Applying the iDFT requires the same number of operations.

In MATLAB:
\( F_n(x) \) is computed by the command \( \text{fft}(x)/\sqrt{n} \).
\( F_n^{-1}(x) \) is computed by the command \( \text{ifft}(x)\star\sqrt{n} \).
The DFT of a real vector

If all the entries of $x$ are real,

$$y_0 = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^0 = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \in \mathbb{R}$$

Note that for $k = 1, \ldots, n - 1$,

$$\omega^{n-k} = e^{-i2\pi(n-k)/n} = e^{i2\pi k/n} = \omega^k$$
\[ y_{n-k} = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^j (n-k) \]

\[ = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j (\omega^k)^j \]

\[ = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} \bar{x}_j \omega^{jk} = \bar{y}_k \]
The DFT of a real vector if $n$ is even

$$
F_{10} = \begin{pmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 \\
  x_6 \\
  x_7 \\
  x_8 \\
  x_9 \\
\end{pmatrix}
= \begin{pmatrix}
  a_0 \\
  a_1 + ib_1 \\
  a_2 + ib_2 \\
  a_3 + ib_3 \\
  a_4 + ib_4 \\
  a_5 \\
  a_4 - ib_4 \\
  a_3 - ib_3 \\
  a_2 - ib_2 \\
  a_1 - ib_1 \\
\end{pmatrix}
= \begin{pmatrix}
  y_0 \\
  y_1 \\
  \vdots \\
  y_{\frac{n}{2}-2} \\
  y_{\frac{n}{2}-1} \\
  y_{\frac{n}{2}} \\
  \overline{y_{\frac{n}{2}-1}} \\
  \overline{y_{\frac{n}{2}-2}} \\
  \ldots \\
  \overline{y_1} \\
\end{pmatrix}
$$
The Fast Fourier Transform

The Cooley and Tukey algorithm (FFT) reduces the complexity of the DFT from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$.

Recall that

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}, \quad k = 0, 1, \ldots, n - 1$$

and let’s concentrate on the computation of

$$z_k = \sum_{j=0}^{n-1} x_j \omega^{jk}$$
Computation of $z_k$ when $n = 2^N$

$$z_k = \sum_{j=0}^{n/2-1} x_j \omega^{jk} + \sum_{j=n/2}^{n-1} x_j \omega^{jk}$$

$$= \sum_{j=0}^{n/2-1} x_j \omega^{jk} + \sum_{j=0}^{n/2-1} x_{j+n/2} \omega^{(j+n/2)k}$$

$$= \sum_{j=0}^{n/2-1} \left( x_j + \omega^{kn/2} x_{j+n/2} \right) \omega^{jk}$$

For $k = 0, 1, \ldots, n - 1$, there is a plus sign when $k$ is even and a negative sign when $k$ is odd.
Transformation into two half-length vectors

\[ z_{2k} = \sum_{j=0}^{n/2-1} \left( x_j + x_{j+n/2} \right) g_j \omega^{2jk} \]

\[ z_{2k+1} = \sum_{j=0}^{n/2-1} \left( x_j - x_{j+n/2} \right) h_j \omega^{2jk} \]

for \( k = 0, 1, \ldots, n/2 - 1 \).

We have two transforms of length \( n/2 = 2^{N-1} \) instead of one of length \( n = 2^N \).
Butterfly network

If \( n = 2^N \), the FFT can be completed in \( n(2 \log_2 n - 1) + 3 \) arithmetic operations.
Complexity when $n$ is not a power of 2

Computational Effort for FFT in Dependency on Prime Factors

<table>
<thead>
<tr>
<th>Prime</th>
<th>Number of Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,5,409</td>
<td>10^9</td>
</tr>
<tr>
<td>2,3,11,31</td>
<td>10^8</td>
</tr>
<tr>
<td>2,23,89</td>
<td>10^7</td>
</tr>
<tr>
<td>3,5,7,13</td>
<td>10^6</td>
</tr>
<tr>
<td>2</td>
<td>10^5</td>
</tr>
<tr>
<td>17,241</td>
<td>10^5</td>
</tr>
<tr>
<td>2,3,683</td>
<td>10^4</td>
</tr>
</tbody>
</table>

Arévalo FMN011
Fast Inverse Fourier Transform

As the inverse Fourier matrix is the complex conjugate matrix,

\[ F_n^{-1} = \bar{F}_n, \]

to carry out the inverse Fourier transform of (complex) vector \( y \):

1. Conjugate: \( y \to \bar{y} \)

2. Apply the FFT: \( \bar{y} \to F_n\bar{y} \)

3. Conjugate: \( F_n\bar{y} \to \bar{F}_n\bar{y} = \bar{F}_n y = F_n^{-1} y \)