the Discrete Fourier Transform

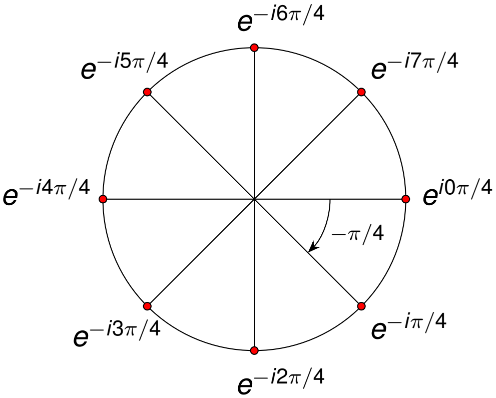
1. The Discrete Fourier Transform
   - roots of unity
   - definition of the DFT
2. Convolutions and the DFT
   - convolutions become componentwise products
   - applying the discrete Fourier transform
3. Interpolation by the DFT
   - the DFT interpolation theorem
   - applied to filter design

MCS 472 Lecture 9
Industrial Math & Computation
Jan Verschelde, 29 January 2024

the Discrete Fourier Transform

roots of unity

The eight roots of unity, generated by $\omega = e^{-i2\pi/8} = e^{-i\pi/4}$, are



$$\omega = e^{-i\pi/4} = \cos(\pi/4) - i\sin(\pi/4), \quad i = \sqrt{-1}, \quad \text{(Euler's formula)}$$

roots of unity and primitive roots

Definition (roots of unity)

The number z is an *nth root of unity* if $z^n - 1 = 0$.

Definition (primitive root of unity)

An nth root of unity is *primitive*
if it is not a kth root of unity for any $k < n$.

Exercise 1:

For $n = 8$, write all nth roots that are primitive.

Verify that for each primitive root z,
all other eight roots can be generated by taking powers.

sums of powers of the *n*th root of unity

For $n > 0$, the nth primitive root of unity is $\omega = e^{-i2\pi/n}$.

ω is a root of the equation

$$x^n - 1 = 0,$$

the other $n - 1$ roots are the powers ω^k, $k = 2, 3, \ldots, n$, with $\omega^n = \omega^0 = 1$.

The root 1 makes the polynomial $x^n - 1$ factor:

$$\omega^n - 1 = (\omega - 1)\left(1 + \omega + \omega^2 + \cdots + \omega^{n-1}\right) = 0.$$

For $\omega \neq 1$, we have then

$$1 + \omega + \omega^2 + \cdots + \omega^{n-1} = 0.$$

sums of the powers of the roots of unity

As the other $n-2$ roots are powers ω^k, $k = 1, 2, \ldots, n-1$:

$$1 + \omega^k + \omega^{2k} + \cdots + \omega^{(n-1)k} = 0.$$

For $k = n$, we have

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

Proposition (the Gauss relation)

Let ω be a primitive n root of unity and let k be an integer.

$$\sum_{j=0}^{n-1} \omega^{j\,k} = \begin{cases} n & \text{if } k/n \text{ is an integer,} \\ 0 & \text{otherwise.} \end{cases}$$

the Discrete Fourier Transform

1. The Discrete Fourier Transform
   - roots of unity
   - definition of the DFT
2. Convolutions and the DFT
   - convolutions become componentwise products
   - applying the discrete Fourier transform
3. Interpolation by the DFT
   - the DFT interpolation theorem
   - applied to filter design

the discrete Fourier transform

Definition (the Discrete Fourier Transform (DFT))

Let $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^T$ be an n-dimensional vector.

The *Discrete Fourier Transform* of $\mathbf{x}$ is $\mathbf{y} = [y_0, y_1, \ldots, y_{n-1}]^T$, where

$$y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{j\,k}, \quad \omega = e^{-i2\pi/n}.$$

the matrix representation

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

For example, for $n = 8$, in matrix form the DFT is

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \frac{1}{\sqrt{n}} \begin{bmatrix}
\omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} & \omega^{0\cdot 0} \\
\omega^{1\cdot 0} & \omega^{1\cdot 1} & \omega^{1\cdot 2} & \omega^{1\cdot 3} & \omega^{1\cdot 4} & \omega^{1\cdot 5} & \omega^{1\cdot 6} & \omega^{1\cdot 7} \\
\omega^{2\cdot 0} & \omega^{2\cdot 1} & \omega^{2\cdot 2} & \omega^{2\cdot 3} & \omega^{2\cdot 4} & \omega^{2\cdot 5} & \omega^{2\cdot 6} & \omega^{2\cdot 7} \\
\omega^{3\cdot 0} & \omega^{3\cdot 1} & \omega^{3\cdot 2} & \omega^{3\cdot 3} & \omega^{3\cdot 4} & \omega^{3\cdot 5} & \omega^{3\cdot 6} & \omega^{3\cdot 7} \\
\omega^{4\cdot 0} & \omega^{4\cdot 1} & \omega^{4\cdot 2} & \omega^{4\cdot 3} & \omega^{4\cdot 4} & \omega^{4\cdot 5} & \omega^{4\cdot 6} & \omega^{4\cdot 7} \\
\omega^{5\cdot 0} & \omega^{5\cdot 1} & \omega^{5\cdot 2} & \omega^{5\cdot 3} & \omega^{5\cdot 4} & \omega^{5\cdot 5} & \omega^{5\cdot 6} & \omega^{5\cdot 7} \\
\omega^{6\cdot 0} & \omega^{6\cdot 1} & \omega^{6\cdot 2} & \omega^{6\cdot 3} & \omega^{6\cdot 4} & \omega^{6\cdot 5} & \omega^{6\cdot 6} & \omega^{6\cdot 7} \\
\omega^{7\cdot 0} & \omega^{7\cdot 1} & \omega^{7\cdot 2} & \omega^{7\cdot 3} & \omega^{7\cdot 4} & \omega^{7\cdot 5} & \omega^{7\cdot 6} & \omega^{7\cdot 7}
\end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix}$$

The n-by-n matrix in this representation is the *Fourier matrix*.

the Fourier matrix

For $\omega = e^{-i2\pi/n}$, the Fourier matrix is

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}.$$

Then for $\mathbf{x}$ and $\mathbf{y}$ two n-dimensional vectors for which

$$\mathbf{y} = F_n \mathbf{x},$$

we have that $\mathbf{y}$ is the DFT of $\mathbf{x}$.

the inverse of the Fourier matrix

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^1 & \omega^2 & \cdots & \omega^{n-1} \\ \omega^0 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2} \end{bmatrix}$$

has as inverse (by the Gauss relation):

$$F_n^{-1} = \frac{1}{\sqrt{n}} \begin{bmatrix} \omega^0 & \omega^0 & \omega^0 & \cdots & \omega^0 \\ \omega^0 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ \omega^0 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-(2(n-1))} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \omega^0 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)^2} \end{bmatrix}.$$

Observe: $F_n^{-1} = \overline{F_n}$, the complex conjugate of F. $\omega = e^{-i2\pi/n}$

the inverse discrete Fourier transform

Definition (the inverse Discrete Fourier Transform (iDFT))

Let $\mathbf{y} = [y_0, y_1, \ldots, y_{n-1}]^T$ be an n-dimensional vector.

The *inverse Discrete Fourier Transform* of $\mathbf{y}$ is $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^T$,

$$\mathbf{x} = F_n^{-1}\mathbf{y},$$

where F_n is the n-by-n Fourier matrix.

Componentwise, we have the formulas:

$$x_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(\omega^{-k}\right)^j y_k, \quad \omega = e^{-i2\pi/n}, \quad j = 0, 1, \ldots, n-1.$$

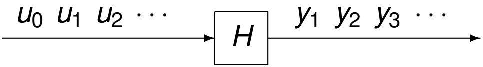
verify with Julia functions

Exercise 2:

1. Write a Julia function `FourierMatrix` with takes on input n and which returns the Fourier matrix F_n.
2. Write a Julia function `inverseFourierMatrix` with takes on input n and which returns the inverse Fourier matrix F_n^{-1}.
3. Verify for $n = 8$ that the product of the output of your `FourierMatrix`(n) with the output of your `inverseFourierMatrix`(n) is indeed the identity matrix.

the Discrete Fourier Transform

filters and convolutions



A linear, time invariant, causal filter is determined by the impulse response $\left\{ h_k \right\}_{k=0}^{\infty}$ and the transfer function is $H(z) = \displaystyle\sum_{k=0}^{\infty} h_k z^{-k}$.

For input u, the k-th element in the output y is

$$y_k = h_k u_0 + h_{k-1} u_1 + \cdots + h_1 u_{k-1} + h_0 u_k = \sum_{j=0}^{k} h_{k-j} u_j.$$

The convolution is denoted by the operator $\star$, as $y = h \star u$.

filtering a periodic signal

$$u_0, u_1, u_2, u_3, u_0, \ldots \rightarrow (h_0, h_1, h_2, h_3) \rightarrow y_0, y_1, y_2, y_3, y_0, \ldots$$

Let us compute $y = h \star u$, applying $y_k = \sum_{j=0}^{k} h_{k-j} u_j$:

$$y_0 = h_0 u_0 + h_{-1} u_1 + h_{-2} u_2 + h_{-3} u_3.$$

By the periodicity: $h_{-1} = h_3$, $h_{-2} = h_2$, and $h_{-3} = h_1$.

$$\begin{aligned} y_0 &= h_0 u_0 + h_3 u_1 + h_2 u_2 + h_1 u_3 \\ y_1 &= h_1 u_0 + h_0 u_1 + h_3 u_2 + h_2 u_3 \\ y_2 &= h_2 u_0 + h_1 u_1 + h_0 u_2 + h_3 u_3 \\ y_3 &= h_3 u_0 + h_2 u_1 + h_1 u_2 + h_0 u_3 \end{aligned}$$

the Discrete Fourier Transform

applying the discrete Fourier transform

Apply $y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{j\,k}$, for $n = 4$, $k = 1$, on (y_0, y_1, y_2, y_3) as input:

$$\widehat{y}_1 = \frac{1}{\sqrt{4}} \left(y_0\omega^0 + y_1\omega^1 + y_2\omega^2 + y_3\omega^3 \right).$$

Multiply the formulas with powers of ω:

$$\begin{array}{rcl}
\omega^0 y_0 & = & \omega^0 h_0 u_0 + \omega^0 h_3 u_1 + \omega^0 h_2 u_2 + \omega^0 h_1 u_3 \\
\omega^1 y_1 & = & \omega^1 h_1 u_0 + \omega^1 h_0 u_1 + \omega^1 h_3 u_2 + \omega^1 h_2 u_3 \\
\omega^2 y_2 & = & \omega^2 h_2 u_0 + \omega^2 h_1 u_1 + \omega^2 h_0 u_2 + \omega^2 h_3 u_3 \\
\omega^3 y_3 & = & \omega^3 h_3 u_0 + \omega^3 h_2 u_1 + \omega^3 h_1 u_2 + \omega^3 h_0 u_3
\end{array}$$

Adding up the above left hand sides leads to

$$\omega^0 y_0 + \omega^1 y_1 + \omega^2 y_2 + \omega^3 y_3 = 2\widehat{y}_1.$$

collecting terms

$$
\begin{array}{rcl}
\omega^0 y_0 & = & \omega^0 h_0 u_0 + \omega^0 h_3 u_1 + \omega^0 h_2 u_2 + \omega^0 h_1 u_3 \\
\omega^1 y_1 & = & \omega^1 h_1 u_0 + \omega^1 h_0 u_1 + \omega^1 h_3 u_2 + \omega^1 h_2 u_3 \\
\omega^2 y_2 & = & \omega^2 h_2 u_0 + \omega^2 h_1 u_1 + \omega^2 h_0 u_2 + \omega^2 h_3 u_3 \\
\omega^3 y_3 & = & \omega^3 h_3 u_0 + \omega^3 h_2 u_1 + \omega^3 h_1 u_2 + \omega^3 h_0 u_3
\end{array}
$$

Adding up the above right hand sides and collecting terms gives

$$
\begin{array}{rl}
& u_0 \left(\omega^0 h_0 + \omega^1 h_1 + \omega^2 h_2 + \omega^3 h_3 \right) \\
+ & u_1 \left(\omega^0 h_3 + \omega^1 h_0 + \omega^2 h_1 + \omega^3 h_2 \right) \\
+ & u_2 \left(\omega^0 h_2 + \omega^1 h_3 + \omega^2 h_0 + \omega^3 h_1 \right) \\
+ & u_3 \left(\omega^0 h_1 + \omega^1 h_2 + \omega^2 h_3 + \omega^3 h_0 \right)
\end{array}
$$

the Fourier transform of h_0, h_1, h_2, h_3

Apply $y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{jk}$, for $n = 4$, $k = 1$, on (h_0, h_1, h_2, h_3) as input:

$$\widehat{h}_1 = \frac{1}{\sqrt{4}} \left(h_0\omega^0 + h_1\omega^1 + h_2\omega^2 + h_3\omega^3 \right).$$

Now we can rewrite the added right hand sides:

$$\begin{array}{rcl} u_0 \left(\omega^0 h_0 + \omega^1 h_1 + \omega^2 h_2 + \omega^3 h_3 \right) & = & u_0\, 2\widehat{h}_1 \\ + \;\; u_1 \left(\omega^0 h_3 + \omega^1 h_0 + \omega^2 h_1 + \omega^3 h_2 \right) & = & u_1\, 2\widehat{h}_1\, \omega^1 \\ + \;\; u_2 \left(\omega^0 h_2 + \omega^1 h_3 + \omega^2 h_0 + \omega^3 h_1 \right) & = & u_2\, 2\widehat{h}_1\, \omega^2 \\ + \;\; u_3 \left(\omega^0 h_1 + \omega^1 h_2 + \omega^2 h_3 + \omega^3 h_0 \right) & = & u_3\, 2\widehat{h}_1\, \omega^3 \end{array}$$

the Fourier transform of u_0, u_1, u_2, u_3

Apply $y_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{j\,k}$, for $n = 4$, $k = 1$, on (u_0, u_1, u_2, u_3) as input:

$$\widehat{u}_1 \quad = \quad \frac{1}{\sqrt{4}} \left(u_0 \omega^0 + u_1 \omega^1 + u_2 \omega^2 + u_3 \omega^3 \right).$$

So we found

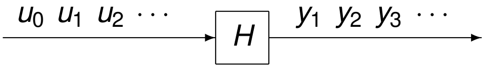
$$\begin{aligned} 2\widehat{y}_1 &= u_0\, 2\widehat{h}_1 + u_1\, 2\widehat{h}_1\, \omega^1 + u_2\, 2\widehat{h}_1\, \omega^2 + u_3\, 2\widehat{h}_1\, \omega^3 \\ &= 2\widehat{h}_1 \left(u_0 + u_1 \omega^1 + u_2 \omega^2 + u_3 \omega^3 \right) \\ &= 2\widehat{h}_1 2\widehat{u}_1, \end{aligned}$$

or, more in general:

$$\widehat{y}_1 = \sqrt{n}\, \widehat{h}_1 \widehat{u}_1,$$

where $\widehat{y} = \mathrm{DFT}(y)$, $\widehat{h} = \mathrm{DFT}(h)$, and $\widehat{u} = \mathrm{DFT}(u)$.

the DFT convolution property



The filter H has impulse response $\left\{ h_k \right\}_{k=0}^{\infty}$.

$$y = h \star u \xrightarrow{\text{DFT}} \widehat{y} = \sqrt{n}\,\widehat{h} \cdot \widehat{u}$$

convolution $\qquad$ componentwise product

where $\widehat{y} = \mathrm{DFT}(y)$, $\widehat{h} = \mathrm{DFT}(h)$, and $\widehat{u} = \mathrm{DFT}(u)$.

Theorem (the DFT convolution property)

The discrete Fourier transform of $h \star u$ is $\sqrt{n}$ times the componentwise product of the discrete Fourier transforms of h and u.

verify numerically and symbolically

Exercise 3:

Verify the DFT convolution property
on two random vectors **x** and **y**, for $n = 8$.

1. Use your `FourierMatrix` of Exercise 2
   to compute the DFT of **x** and **y**, $\widehat{\mathbf{x}} = \mathrm{DFT}(\mathbf{x})$ and $\widehat{\mathbf{y}} = \mathrm{DFT}(\mathbf{y})$.
2. Verify that $\sqrt{8}$ times the componentwise product of $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$
   equals the DFT of $\mathbf{x} \star \mathbf{y}$.

Exercise 4:

We derived the statement of the DFT convolution property
for $n = 4$ and $k = 1$.

Verify the DFT convolution property
by symbolic calculation for $n = 4$ and $k = 2$.

the DFT convolution theorem

Theorem (the DFT convolution property)

Let $\mathbf{x}$ *and* $\mathbf{y}$ *be two n-dimensional vectors.*
The discrete Fourier transform of $\mathbf{x} \star \mathbf{y}$ *is* $\sqrt{n}$ *times the componentwise product of the discrete Fourier transforms of* $\mathbf{x}$ *and* $\mathbf{y}$*.*

Theorem (the DFT convolution theorem)

Let $\mathbf{x}$ *and* $\mathbf{y}$ *be two n-dimensional vectors.*
The convolution $\mathbf{x} \star \mathbf{y}$ *can be computed as*

$$\mathbf{x} \star \mathbf{y} = \mathrm{iDFT}(\sqrt{n}\, \mathrm{DFT}(\mathbf{x}) \cdot \mathrm{DFT}(\mathbf{y})),$$

where DFT *is the discrete Fourier transform and*
iDFT *is the inverse discrete Fourier transform.*
$\cdot$ *is the componentwise product of two vectors.*

the Discrete Fourier Transform

the DFT interpolation theorem

Theorem (the DFT Interpolation Theorem)

Consider n points $t_j = j/n$, for $j = 0, 1, \ldots, n-1$.
Let $\mathbf{x} = [x_0, x_1, \ldots, x_{n-1}]^T$, $\mathbf{y} = F_n \mathbf{x}$, where F_n is the Fourier matrix.
Then

$$f(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{i2\pi kt}$$

satisfies $f(t_j) = x_j$, for $j = 0, 1, \ldots, n-1$.

The coefficients y_k of the discrete Fourier transform are the coefficients of an interpolating function $f(t)$ in a trigonometric basis.

proof by the inverse DFT

$$\mathbf{y} = F_n \mathbf{x}, \quad f(t) = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} y_k e^{i2\pi kt}, \quad f(t_j) = x_j, t_j = j/n, \quad j = 0, 1, \ldots, n$$

Proof: we use $\mathbf{x} = F_n^{-1} \mathbf{y}$, for $j = 0, 1, \ldots, n-1$:

$$\begin{aligned} x_j &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(\omega^{-k}\right)^j y_k, \quad \omega = e^{-i2\pi/n} \\ &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(e^{i2\pi k\, j/n}\right) y_k \\ &= \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} \left(e^{i2\pi k\, t_j}\right) y_k = f(t_j). \end{aligned}$$

Q.E.D.

the 8th roots of unity again

Exercise 5:

Verify the DFT interpolation property for $n = 8$.

1. Generate a random vector $\mathbf{x}$ of size 8.
2. Compute $\mathbf{y} = F_n \mathbf{x}$, with your `FourierMatrix` of Exercise 2.
3. Define the function $f(t)$.
4. Verify that $f(j/n) = x_j$, for $j = 0, 1, \ldots, n-1$.

the Discrete Fourier Transform

1. The Discrete Fourier Transform
   - roots of unity
   - definition of the DFT

2. Convolutions and the DFT
   - convolutions become componentwise products
   - applying the discrete Fourier transform

3. Interpolation by the DFT
   - the DFT interpolation theorem
   - applied to filter design

applied to filter design

Theorem (amplitude gain and phase shift of filter)

Let $H(z) = \sum_{k=0}^{\infty} h_k z^{-k}$ *be the transfer function of a filter F.*

For input $u = \left\{ u_k = \sin(\omega k T) \right\}_{k=0}^{\infty}$, $y = \left\{ y_k = r \sin(\omega k T + \phi) \right\}_{k=0}^{\infty}$ *is the output, where* ($\omega = 2\pi n$, *T is the sampling rate),*

- $r = |H(e^{i\omega T})|$ *is the amplitude gain, and*
- $\phi = \arg H(e^{i\omega T})$ *is the phase shift.*

Filter design in three steps:

1. Make the desired gain $r = r(t)$ and phase shift $\phi = \phi(t)$.
2. Evaluate the desired gains and phase shifts at equidistant angles $\theta_k \in [0, 2\pi]$, $r_k = r(\theta_k)$, $\phi_k = \phi(\theta_k)$, $\widehat{h}_k = r_k e^{i\phi_k}$.
3. $h = \mathrm{iDFT}(\widehat{h})$ is the impulse response, which defines $H(z)$.

summary and bibliography

We defined the Discrete Fourier Transform (DFT), derived the DFT Convolution Theorem and proved the DFT Interpolation Theorem.

The main references for this lecture:

- Charles R. MacCluer:
  Industrial Mathematics. Modeling in Industry, Science, and Government. Prentice Hall, 2000.

  We started Chapter 4.

- Timothy Sauer: *Numerical Analysis*,
  second edition, Pearson, 2012.

  Chapter 10 deals with the discrete Fourier transform.