Discrete Fourier Transform: Interpolation

Problem (General Interpolation Task)

For a given function $g : \mathbb{R} \to \mathbb{R}$ with given points

$$(x_j, g(x_j)), \quad j = 1, \ldots, n,$$

we search for a function p, s.t. the graph of p goes through these given points.

p is chosen within a class of reasonable (and manageable) functions, e.g.

- *the class of real polynomials,*
- *the class of trigonometric polynomials in case of $T-$periodic functions.*

Trigonometric Approximation Revisited

In the later case we consider a function $g : [0, 2\pi] \to \mathbb{R}$, periodically extended with $T = 2\pi$.
Let

$$x_0 = 0, x_1 = \frac{2\pi}{n}, \ldots, x_{n-1} = (n-1)\frac{2\pi}{n}, x_n = 2\pi.$$

We search for a trigonometric polynomial

$$p_n(x) := \sum_{k=0}^{n-1} c_k \exp(ikx)$$

such that

$$p_n(x_j) = g(x_j) \quad \text{for } j = 0, \ldots, n-1.$$

Remark: Due to the periodicity, we have $g(x_0) = g(x_n)$.

By inserting:

$$\sum_{k=0}^{n-1} \exp(ikx_j) c_k = g(x_j) \quad \text{for } j = 0, \ldots, n-1.$$

This is a linear equation system (LES) with coefficient matrix

$$f_{jk} = \exp\left(2\pi i k \frac{j}{n}\right) = \zeta^{jk} \quad \text{with } \zeta := \exp\left(2\pi i \frac{1}{n}\right).$$

Fourier Matrix

This motivates the definition of the so-called **Fourier matrix** :

$$F_n = (f_{ij})_{i,j=1,\ldots,n}$$

Properties of the Fourier matrix:

- F_n is symmetric
- $\frac{1}{\sqrt{n}} F_n$ is unitary, i.e. $\left(\frac{1}{\sqrt{n}} F_n\right)^{-1} = \overline{\left(\frac{1}{\sqrt{n}} F_n\right)}^{\top}$ (details later in LA)

Thus, we may rewrite the LES

$$F_n \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{pmatrix}$$

that uniquely determines the trigonometric polynomial p_n by

$$\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} = \frac{1}{n} \overline{F}_n^{\top} \begin{pmatrix} g(x_1) \\ \vdots \\ g(x_n) \end{pmatrix}$$

Discrete Fourier Transform

The mapping

$$\mathbb{C}^n \to \mathbb{C}^n, v \mapsto F_n v$$

is called **discrete Fourier transform (DFT)**, i.e.

$$(F_n v)_j = \sum_{k=0}^{n-1} v_k \exp\left(2\pi i \frac{jk}{n}\right).$$

This may be interpreted as a discretization of the (continuous) Fourier transform (at the end of lecture).

The inverse DFT solves the trigonometric interpolation problem.

By an algorithm, called **fast Fourier transform (FFT)** the DFT may be computed with only "few" operations!

Example: Fourier Matrix

Let $v^\top = \left(1, -1, 0, 1\right)$. $= \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix}$ $\qquad \zeta_{(4)} = e^{i\frac{2\pi}{4}} = i$

Compute

$$F_4\, v = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & \zeta^1 & \zeta^2 & \zeta^3 \\ 1 & \zeta^2 & \zeta^4 & \zeta^6 \\ 1 & \zeta^3 & \zeta^6 & \zeta^9 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & i & i^2 = -1 & -i \\ 1 & -1 & 1 & -1 \\ 1 & -i & -1 & i \end{pmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 - 1 + 0 + 1 \\ 1 - i + 0 - i \\ 1 + 1 + 0 - 1 \\ 1 + i + 0 + i \end{pmatrix} = \begin{pmatrix} 1 \\ 1 - 2i \\ 1 \\ 1 + 2i \end{pmatrix}$$

Image & Audio-Video Compression

An image with decreasing quality from the left to the right. (Source: Wikipedia)

Typical standards: `jpeg`, `mpeg`

An important step (among others) in these data compression methods is the discrete Fourier cosine transform that maps into the frequency domain.

Theoretically, this step is invertible, i.e. without loss of information possible.

Outlook Fourier Analysis

At the end of a lecture (if time permits) we will have a short look at

- (Continuous) Fourier transform
- Laplace transform
- Short-time Fourier transform
- Wavelet