1. **Discrete Fourier Transform Algorithm**
For a set of N equidistant data points y_k with distance h, the *Discrete Fourier Transform* (DFT) is equivalent to a matrix-vector multiplication:

$$Y_n = \sum_{k=1}^{N} A_{nk} y_k, \tag{1}$$

with

$$A_{nk} = \frac{1}{\sqrt{2\pi}} Z^{nk}, \tag{2}$$

where Z^{nk} is the $(n \times k)$-th power of $Z = \exp(-2\pi i/N)$. The corresponding frequencies are then given by $\omega_n = n\frac{2\pi}{Nh}$. To get the inverse transform you just have to discretize the integral of the continuous form (in the same fashion as the forward transformation) obtaining:

$$y_n = \sum_{k=1}^{N} A'_{nk} Y_k, \tag{3}$$

with

$$A'_{nk} = \frac{\sqrt{2\pi}}{N} Z^{-nk}, \tag{4}$$

Implement the transform and its inverse and use it to compute the DFT of several functions.

(a) Transform the sawtooth function,

$$f(t) = \begin{cases} t & t \leq T/2 \\ t - T & t > T/2 \end{cases}, \tag{5}$$

and plot the real and imaginary part of Y_n. Check that the back transformation works in this case.

(b) Transform the following functions, and transform them back. Plot again the real and imaginary part of Y_k and compare them for the different functions. Use $N = 8$ for these functions.

$$\begin{aligned} g_1(t) &= \sin(t) \\ g_2(t) &= \cos(t) \\ g_3(t) &= 3 + \cos(t) \\ g_4(t) &= 3 + \cos(5 + t) \end{aligned}$$

(c) Transform $y(t) = \exp\left(-\frac{t}{2\pi}\right)$.

(d) Choose an even and an odd function to transform and study the magnitude of the real and imaginary part of Y_k for both cases. Use $N = 128$ for these functions.

(e) (Optional) Implement the *Fast Fourier Transform* (FFT) as it was covered in the lecture. Compare the results to the DFT algorithm.