Question 1: Discrete Fourier Transform (DFT)

The following 3 values are known from the $2\pi$-periodic function $f$, i.e.

$$f_0 = f(0), \quad f_1 = f\left(\frac{2\pi}{3}\right), \quad f_2 = f\left(\frac{4\pi}{3}\right)$$

Use the DFT method to determine the interpolating trigonometric polynomial.

Solution:

Discrete fourier transform,

$$\tilde{c}_j = \frac{1}{N} \sum_{l=0}^{N-1} f(x_l) e^{-\frac{ik2\pi l}{N}} \implies \tilde{c}^N = \frac{1}{N} W f^N$$

$$W_{kl} = e^{-\frac{ik2\pi l}{N}}, \quad j, l = 0, 1, 2$$

$$W = \begin{pmatrix} 1 & 1 & e^{-\frac{4\pi}{N}} e^{\frac{8\pi}{N}} \\ 1 & e^{-\frac{4\pi}{3}} & e^{-\frac{4\pi}{3}} e^{\frac{16\pi}{3}} \\ 1 & e^{-\frac{4\pi}{3}} e^{\frac{8\pi}{3}} & e^{-\frac{8\pi}{3}} \end{pmatrix}$$

Inverse discrete fourier transform $W^{-1} = W^T \frac{1}{N} \implies f^N = W^T \tilde{c}^N$

$$f_k = \sum_{j=0}^{j=N^{-1}} \tilde{c}_j e^{\frac{ik2\pi j}{N}}$$

$$= \sum_{j=0}^{j=\frac{1}{2}(N-1)} \tilde{c}_j e^{\frac{ik2\pi j}{N}} + \sum_{j=\frac{1}{2}(N-1)+1}^{j=(N-1)} \tilde{c}_j e^{\frac{ik2\pi j}{N}}$$

$$= \alpha + \beta$$

Substitute in $\beta : l = j - N$, i.e.

$$\beta = \sum_{l=-\frac{1}{2}(N-1)}^{-1} \tilde{c}_{l+N} e^{\frac{i2\pi k(l+N)}{N}} = \sum_{l=-\frac{1}{2}(N-1)}^{-1} \tilde{c}_{l+N} e^{\frac{i2\pi kl}{N}} e^{\frac{i2\pi kN}{N}} = \sum_{l=-\frac{1}{2}(N-1)}^{-1} \tilde{c}_l e^{\frac{i2\pi kl}{N}}$$
\[ f_k = \alpha + \beta = \sum_{l=-\frac{1}{2}(N-1)}^{\frac{1}{2}(N-1)} \tilde{c}_l e^{i\frac{2\pi kl}{N}} \]

\[ \tilde{c}_j e^{ilx} + \tilde{c}_{-j} e^{-ilx} = (\tilde{c}_j + \tilde{c}_{-j}) \cos(jx) + (\tilde{c}_j - \tilde{c}_{-j}) \sin(jx) = a_j \cos(jx) + b_j \sin(jx) \]

\[ p(x) = \frac{a_0}{2} + \sum_{j=1}^{\frac{1}{2}(N-1)} (a_j \cos(jx) + b_j \sin(jx)) \]

\[ \tilde{c}_0 = \frac{1}{3}(f_0 + f_1 + f_2) \quad \tilde{c}_0 = \frac{1}{3}(f_0 + f_1 + f_2) \]

\[ \tilde{c}_1 = \frac{1}{3}(f_0 + e^{-\frac{2\pi}{3}} f_1 + e^{-\frac{4\pi}{3}} f_2) \quad \tilde{c}_{-1} = \frac{1}{3}(f_0 + e^{\frac{2\pi}{3}} f_1 + e^{\frac{4\pi}{3}} f_2) \]

\[ \tilde{c}_0 = \frac{1}{3}(f_0 + e^{-\frac{2\pi}{3}} f_1 + e^{-\frac{4\pi}{3}} f_2) \quad \tilde{c}_{-1} = \frac{1}{3}(f_0 + e^{\frac{2\pi}{3}} f_1 + e^{\frac{4\pi}{3}} f_2) \]

\[ a_j = \tilde{c}_j + \tilde{c}_{-j} \implies a_0 = \frac{2}{3}(f_0 + f_1 + f_2) \]

\[ \implies a_1 = \frac{2}{3}(2f_0 - f_1 - f_2) \]

\[ b_j = i\tilde{c}_j - i\tilde{c}_{-j} \implies b_0 = 0 \]

\[ \implies b_1 = \frac{1}{3}(\sqrt{3}f_1 - \sqrt{3}f_2) \]

Substituting for \(a_0, b_0, a_1, b_1\) in the trigonometric polynomial

\[ p(x) = \frac{1}{3}(f_0 + f_1 + f_2) + \frac{2}{3}(2f_0 - f_1 - f_2) \cos x + \frac{1}{3}(\sqrt{3}f_1 - \sqrt{3}f_2) \sin x \]
Question 2: Discrete Fourier Transform

Two $2\pi$-periodic functions $f(x), g(x)$ are approximated by $N$ Fourier coefficients $\tilde{c}_j, \tilde{d}_j, j = 0, ..., N - 1$

a) Show that the product $h(x) = f(x)g(x)$ is approximated by $2N - 1$ Fourier coefficients as discrete convolution

$$\tilde{e}_l = \sum_{j=0}^{l} \tilde{c}_j \tilde{d}_{l-j}, \quad l = 0, ..., 2N - 1$$

Solution:

$$f_k = f(x_k) = \sum_{j=0}^{N-1} \tilde{c}_j e^{ijx_k}$$

$$g_k = g(x_k) = \sum_{n=0}^{N-1} \tilde{c}_j e^{inx_k}; \quad x_k = \frac{2\pi k}{N}$$

$$h_k = f_k \cdot g_k = \sum_{j=0}^{N-1} \sum_{n=0}^{N-1} \tilde{c}_j \tilde{d}_n e^{i(j+n)x_k}$$

Put $l = j + n \implies h_k = \sum_{j=0}^{N-1} \sum_{l=j}^{N+j-1} \tilde{c}_j \tilde{d}_{l-j} e^{ilx_k}$  \(\ast\)

Assuming $\tilde{c}_r = 0, \tilde{d}_r = 0$ for $r > N - 1$. This makes $\tilde{c}_j = 0$ for $j > N - 1$ and $\tilde{d}_{l-j} = 0$ for $l > N + j - l$. Under these constraints, we can then represent $\ast$ - the shaded region in the figure below as

$$h_k = \sum_{l=0}^{2N-2} \sum_{j=0}^{l} \tilde{c}_j \tilde{d}_{l-j} e^{ilx_k}$$

$$\tilde{e}_l = \sum_{j=0}^{l} \tilde{c}_j \tilde{d}_{l-j} \implies h_k = \sum_{l=0}^{2N-2} \sum_{j=0}^{l} \tilde{e}_l e^{ilx_k}; \quad x_k = \frac{2k\pi}{N}$$
b) Formulate an algorithm for computing \( \tilde{c}_l \) with computational complexity \( O(N \log N) \)

**Question 3: Fast Fourier Transform (FFT)**

Task 2 has an application in the fast multiplication of power series. Given,

\[
a = (a_0, a_1, \ldots, a_n)^T \\
b = (b_0, b_1, \ldots, b_n)^T, \quad a_i, b_i \in \mathbb{R},
\]

the coefficients of the 2 polynomials of degree \( n \). Determine the product \( c = a \star b \) using the FFT. To do this, select \( a, b \) as \texttt{rand}(1, n+1) for \( n = 63 \). Check your results with MATLAB function \texttt{conv}(a,b) and compute the difference in 2-Norm.

**Solution:**

Compute the FFT of coefficients \( a, b \) to get \( \tilde{a}_j, \tilde{b}_k \), respectively. From the previous task we see that

\[
c = a \cdot b \implies \tilde{c}_l = \sum_{j=0}^{l} \tilde{a}_j \tilde{b}_{l-j}
\]

We can then compute the inverse-FFT(IFT) to compute \( c \), i.e.

\[
\tilde{c}_l \xrightarrow{iFFT} c
\]

**Question 4: Programming Task**

For a periodic function \( N \) values \( f_0, f_1, \ldots, f_{N-1} \) are given. Write a MATLAB-program, which

- plots \( \left( \frac{2k\pi}{N}, f_k \right) \) for \( 0 \leq k \leq N - 1 \).
- Calculate the complex Fourier coefficients \( \tilde{c}_k \) of the interpolating trigonometric polynomial \( F \) with \( f \left( \frac{2k\pi}{N} \right) = f_k \).
- plot \( f \) in the interval \( 0 \leq x \leq 2\pi \)