

## SIMULAÇÃO 2: Transformada de Fourier de Tempo Discreto

1. O MATLAB pode ser utilizado para aproximar a transformada de Fourier de tempo discreto (TFTD) de sinais com duração finita por meio da seguinte expressão:

$$\mathbf{X} = \mathbf{x} \left[ \exp \left( -j \frac{\pi}{M} \mathbf{n}^T \mathbf{k} \right) \right], \quad (1)$$

em que,  $\mathbf{X}$  é um vetor linha com  $M$  amostras, equi-espaciaadas entre  $0$  e  $\pi$  radianos, da TFTD da sequência  $\mathbf{x}$ , e  $\mathbf{n}$  e  $\mathbf{k}$  são vetores linhas dos índices de tempo e frequência, respectivamente, sendo ainda  $\mathbf{n}^T$  a transposta do vetor linha  $\mathbf{n}$ .

Escreva uma função no MATLAB para estimar a TFTD de uma sequência de duração finita. O formato da função deve ser:

```
function [X] = dtft(x,n,k)
% X = valores da TFTD calculados em w pontos de frequências
% x = sequência de duração finita sobre n
% n = vetor de índices de tempo
% k = vetor de índices de frequência
```

Use essa função para estimar a TFTD da sequência  $x[n] = (0.9 \exp(j\pi/3))^n$ ,  $0 \leq n \leq 10$  e investigar sua periodicidade. As funções de magnitude e fase de  $X(e^{j\omega})$  possuem simetria par? Justifique.

**Resposta:** Analisando os gráficos, não é possível identificar simetria par na magnitude ou na fase, bem como não foi possível observar comportamento ímpar para esses gráficos.

Código em python e gráficos:

```
import numpy as np
import matplotlib.pyplot as plt

# Função que faz a estimativa da TFTD
def dtft(x, n, k):
    M = 150
    X = np.zeros_like(k, dtype=np.complex128)
    for i, freq in enumerate(k):
        X[i] = np.sum(x*np.exp(-1j*(np.pi/M)*n*freq))
    return X

n = np.arange(0,10)
x = (0.9*np.exp(1j*np.pi/3))**n
k = np.arange(-400,401)
X = dtft(x,n,k)

# Plota a magnitude de  $X(e^{j\omega})$ 
plt.subplot(2,1,1)
plt.plot(k, np.abs(X))
plt.title('Magnitude  $X(e^{j\omega})$ ')
plt.xlabel('Frequência (k)')
plt.ylabel('| $X(e^{j\omega})$ |')
plt.grid(True)

# Plota a fase de  $X(e^{j\omega})$ 
plt.subplot(2,1,2)
plt.plot(k, np.angle(X))
plt.title('Fase de  $X(e^{j\omega})$ ')
plt.xlabel('Frequência (k)')
plt.ylabel('Ângulo  $X(e^{j\omega})$ (rad)')
plt.grid(True)

plt.tight_layout()
plt.show()
```

