

---

# Implementação da DFT

## Table of Contents

Inicialização .....	1
Função de cálculo da DFT .....	1
Testes das funções .....	2

Aluno: Pedro Henrique Garcia - R.A. 1829696

## Inicialização

```
clear; clc;
M = 200; % tamanho do vetor x
x = randn(M,1); % vetor x aleatório
N = randi([1 100*M]); % escolhemos um numero de pontos da DFT
% Para evitar aliasing, N >= M
```

## Função de cálculo da DFT

```
% function X = phg_dft(x, N)
% função para o cálculo da dft. Recebe um vetor x e o numero de pontos
da
% dft N.

% M é o tamanho do x
M = numel(x);

% para um vetor maior que o numero de amostras da dft, o Matlab trunca
o
% sinal. Por isso, calculamos a dft de N pontos então.
if M>N
    M=N;
end
% pre-alocamos a dft
X(N)=0;

% utilizamos dois laços para calcular a somatória da dft pela
definição
for k=0:N-1
    for n=0:M-1
        X(k+1) = X(k+1) + (x(n+1)*exp( -1i*(2*pi/N)*k*n ));
    end
end

% assegura que a dft será compatível com a do matlab. Se x é coluna, X
% deverá ser coluna também.
if (iscolumn(x) == 1)
    X = transpose(X);
```

end

## Testes das funções

```
% O matlab calcula a dft através do algoritmo da fft, que é
% computacionalmente mais eficiente pois realiza menos multiplicações
% complexas. Por isso, a fft é mais rápida.

tic; X_fft = fft(x,N); toc
tic; X_phg = phg_dft(x,N); toc

% plot do erro, que é de magnitude insignificante.
stem(abs(X_fft - X_phg));
title('Erro')

% podemos verificar também o maior erro entre as dfts
max(abs(X_fft - X_phg))

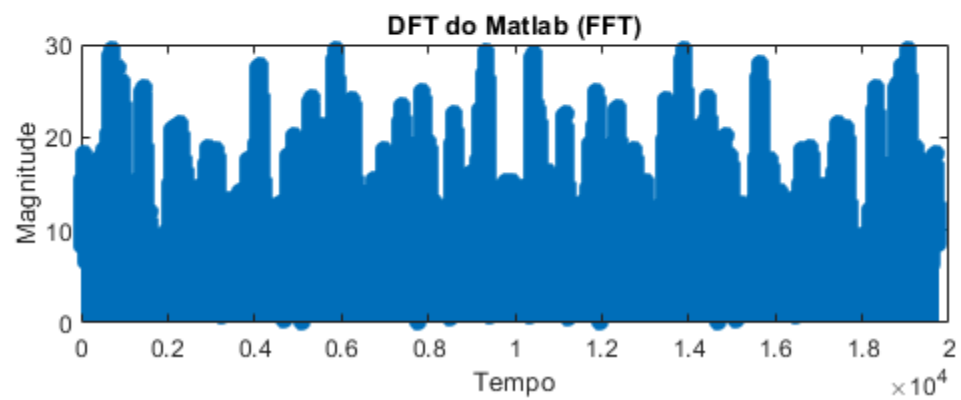
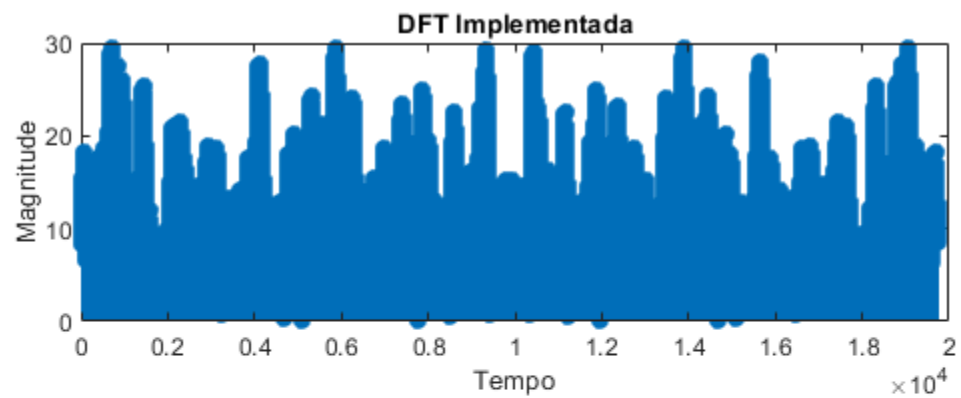
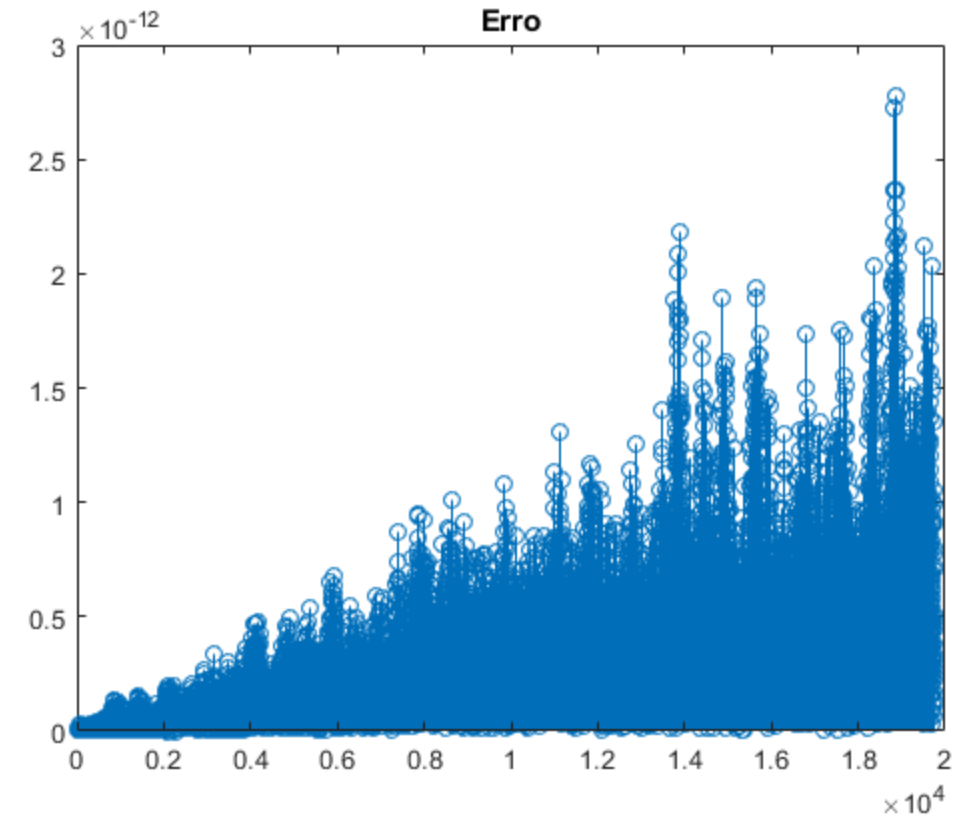
% comparando os dois resultados:

figure;
ax1 = subplot(2,1,1);
stem(ax1,abs(X_phg), 'filled');
title(ax1, 'DFT Implementada');
xlabel(ax1, 'Tempo');
ylabel(ax1, 'Magnitude');

xk_m = fft(x,N);

ax2 = subplot(2,1,2);
stem(ax2,abs(X_fft), 'filled');
title(ax2, 'DFT do Matlab (FFT)');
xlabel(ax2, 'Tempo');
ylabel(ax2, 'Magnitude');

Elapsed time is 0.018714 seconds.
Elapsed time is 0.993350 seconds.
ans =
    2.7786e-12
```



*Published with MATLAB® R2018b*