```
% Considerando que o sinal foi amostrado corretamente a freq maxima
 % que pode estar contida no sinal sera metade de Fs -> pelo teorema de nyquist
 Fs = 18000;
 Fmax = Fs/2;
 figure;
 plot(dados);
 title('Sinal Original');
 xlabel('Amostras');
 ylabel('Amplitude');
 grid on;
                             Sinal Original
      300
      250
      200
      150
      100
       50
       -50
      -100
                         200
                100
                                 300
                                          400
                                                  500
                                                          600
                               Amostras
a. Utilize a equação de análise da Série Discreta de Fourier e obtenha os valores dos coeficientes X\sim[n]
 ANÁLISE EM FREQUÊNCIA DE SINAIS DE TEMPO DISCRETO PERIÓDICOS
```

 $\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n]e^{-j(2\pi/N)kn}$ **EQUAÇÃO DE ANÁLISE** function X = eq_analise(sinal, k)

```
X = 0;
     N = length(sinal);
     for n = 0:N-1
         X = X + (sinal(n+1) * exp(-1i * (2*pi/N) * k * n));
     end
 end
 N = length(dados);
 coeficientes = zeros(1, N);
 for k = 0:N-1
     K = [K, k];
     coeficientes(k+1) = eq_analise(dados, k);
 fase = angle(coeficientes);
 magnitude = abs(coeficientes);
 frequencias = (0:N-1)*(Fs/N);
b. Apresente o gráfico de barras dos valores de X\sim[n] e \angle X\sim[n] em função de k. Quais as componentes múltiplas da frequência fundamental existentes?
```

% Gráficos de Magnitude

title('DFT - Magnitude completa');

100

150

200

figure;

subplot(2,1,1); bar(K, magnitude);

% Gráficos de Fase

50

E as frequencias divisiveis por ela são:

for k = 4:length(K(1:N/2))

disp(K(k))

end

 $ylabel('|X^{(k)}|');$

×104

×104

grid on;

end

100

A freq fundamental sera k=2 com magnitude=5.400000e+04

disp("E as frequencias divisiveis por ela são:")

if mod(K(k), K(3)) == 0 && magnitude(k) > 0.1

150

DFT - Magnitude completa

Frequencia (Hz) DFT - Metade positiva do espectro

2000 4000 6000 8000 10000 12000 14000 16000 18000

Frequencia (Hz) Fase em função da frequencia - Metade positiva

A1 = 9.0000000e+01 / A2 = 9.0000000e+01 / A3 = 3.0000000e+01

function X = eq_sintese(coeficientes, n, N)

sinal(n+1) = eq_sintese(coeficientes, n, N);

f1 = 60 / f2 = 180 / f3 = 300

EQUAÇÃO DE SÍNTESE

X = 0;

sample = [];

for n = 0:N-1

subplot(2,1,1);

end

figure;

figure;

hold on;

grid on;

xlabel('n');

legend('show');

300

250

200

150

100

50

bar(K, phase_fft);

 $ylabel(' \angle X \sim (k)');$

xlabel('k');

grid on;

title('Fase ∠X[k] em função de k FFT');

for k = 1:length(frequencias(1:N/2))

fprintf("%d: %d\n", frequencias(k), mag_fft(k)/N);

Sinal original Dados reconstruídos

1.8788

-2.4803

0.0000

-0.0000

-2.1335

-1.5708

-1.5708

if $mag_fft(k) > 0.1$

end

0: 9.000000e+01 60: 9.000000e+01 180: 3.000000e+01 300: 1.500000e+01

300

250

200

f1 = 60 / f2 = 180 / f3 = 300

disp(fase);

disp(coef_fft);

1.0e+04 *

Comparando as coeficientes e fases:

end

ylabel('Amplitude');

title('Real vs Reconstruído');

sinal = zeros(1, N);

sample = [sample, n];

plot(sample, real(sinal));

200

fprintf("A freq fundamental sera k=%d com magnitude=%d", K(3), magnitude(3))

250

300

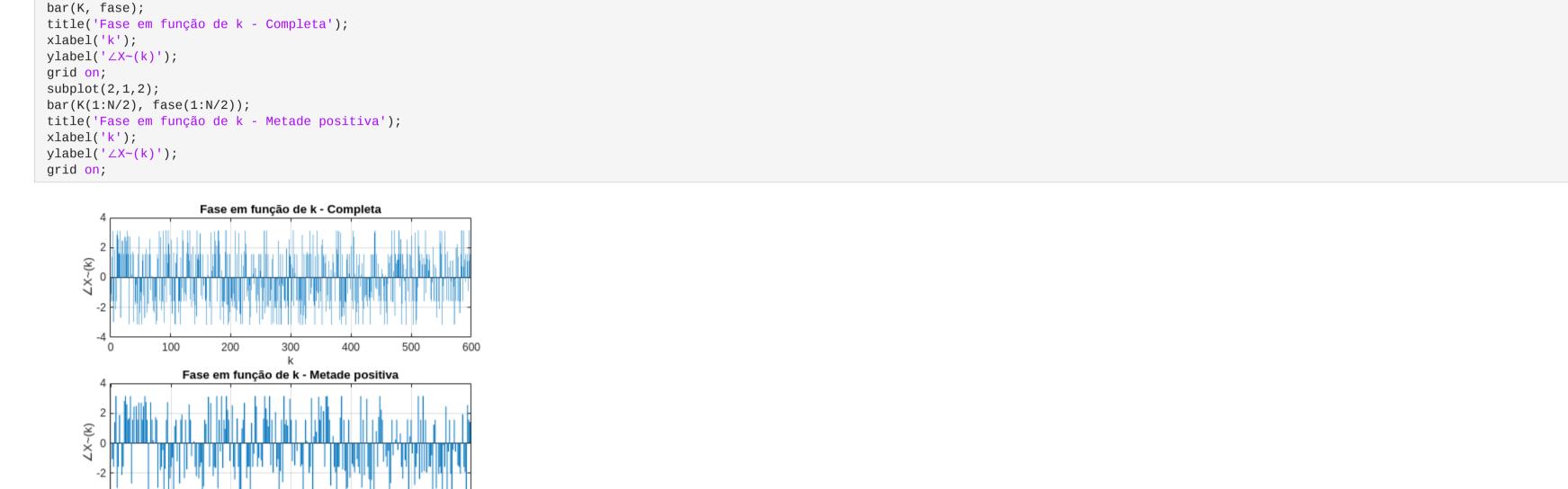
subplot(2,1,1);

figure;

xlabel('k'); $ylabel('|X^{(k)}|');$ grid on;

300

```
subplot(2,1,2);
bar(K(1:N/2), magnitude(1:N/2));
title('DFT - Metade positiva do espectro');
xlabel('k');
ylabel('|X^{(k)}|');
grid on;
       6 ×104
                        DFT - Magnitude completa
                                 300
                     DFT - Metade positiva do espectro
         ×104
```



6 10 c. Apresente o gráfico de barras dos valores de $|X^{-}[n]|$ em função de f(Hz). Que frequências compõem o sinal com base na análise de Fourier? Quais suas amplitudes?

```
% Gráficos de Magnitude
figure;
subplot(2,1,1);
bar(frequencias, magnitude);
title('DFT - Magnitude completa');
xlabel('Frequencia (Hz)');
ylabel('|X^{(k)}|');
grid on;
subplot(2,1,2);
bar(frequencias(1:N/2), magnitude(1:N/2));
title('DFT - Metade positiva do espectro');
xlabel('Frequencia (Hz)');
```





for k = 0:N-1X = X + (coeficientes(k+1) * exp(1i * (2*pi/N) * k * n));end X = X/N;end

e. Com base em $X\sim[n]$ e na equação de síntese da Série Discreta de Fourier, reconstrua o sinal x[n] apresentando seu gráfico.

 $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}$

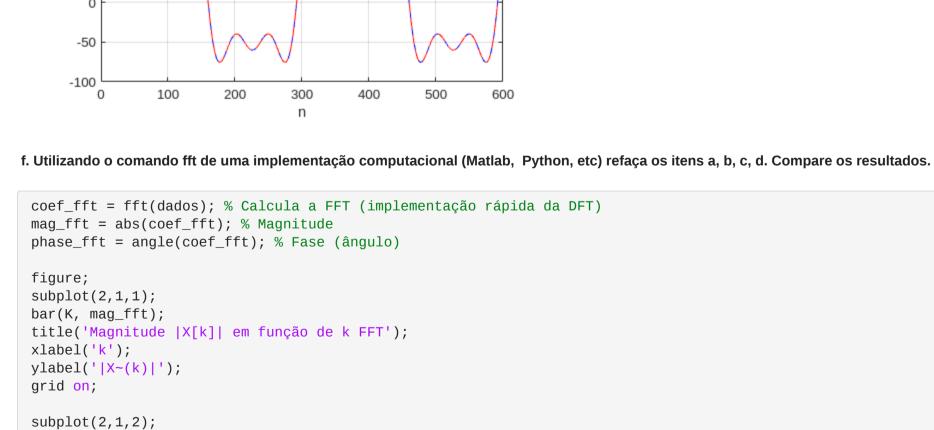
```
title('Parte Real');
xlabel('n');
ylabel('Amplitude');
subplot(2,1,2);
plot(sample, imag(sinal));
title('Parte Imaginária');
xlabel('n');
ylabel('Amplitude');
                               Parte Real
     300
     200
     100
     -100
                 100
                         200
                                           400
                                  300
         ×10<sup>-11</sup>
                             Parte Imaginária
                         200
                                   300
```

plot(real(sinal), 'b', 'DisplayName', 'Sinal original');

plot(dados, 'r--', 'DisplayName', 'Dados reconstruídos');

Real vs Reconstruído

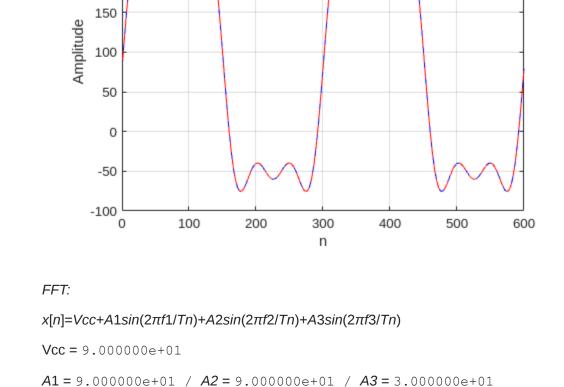
Sinal original - Dados reconstruídos



Magnitude |X[k]| em função de k FFT $\times 10^{4}$

```
100
                        200
                                 300
                                          400
                                                   500
                     Fase ∠X[k] em função de k FFT
                100
                        200
                                  300
                                          400
                                                   500
                                                           600
fprintf("FFT: A freq fundamental sera %d com magnitude=%d", frequencias(3), mag_fft(3)/N)
FFT: A freq fundamental sera 60 com magnitude=9.000000e+01
disp("E as frequencias divisiveis por ela são:")
E as frequencias divisiveis por ela são:
```

sinal_fft = ifft(coef_fft); figure; plot(real(sinal_fft), 'b', 'DisplayName', 'Sinal original'); hold on; plot(dados, 'r--', 'DisplayName', 'Dados reconstruídos'); title('Real vs Reconstruído iFFT'); xlabel('n'); ylabel('Amplitude'); legend('show'); grid on; Real vs Reconstruído iFFT



0 -1.0670 -1.5708 1.4064 3.1416 -2.9832 -1.5708 disp(phase_fft); 0 -2.5687 -1.5708 1.9141 -3.1416 0.5083

%angulos da fase não estão batendo disp(coeficientes);

1.0e+04 * 5.4000 + 0.00001 - 0.0000 - 0.00001 - 0.0000 - 0.00001 - 0.0000 + 0.00001 - 0.00001

2.8285

-1.3517

3.1416

-3.1416

2.5904

-2.6143

1.5708

1.6760

-0.1918

3.1416

-2.6933

1.5708

2.4580

0.0000

2.4898

-2.3303

1.5708

1.5708

2.712

2.306

5.4000 + 0.0000i -0.0000 - 0.0000i -0.0000 - 0.0000i -0.0000 - 0.0000i -0.0000i -0.000i -0.000i -0.000i -0.000i %coeficientes estão iguais, mas não estou entendo o por que a fase esta %dando difença

Comparando minha implementação da DFT com a FFT implementada pelo matlab posso dizer que, em relação aos resultados, os algoritmos apenas se divergiram na fase calculada. Talvez seja por algum arredondamento, ou apenas um erro cometido de minha parte no cálculo.