ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO
Departamento de Engenharia Mecatrônica
Sistemas Dinâmicos II para Mecatrônica
Profs. Larissa Driemeier e Marcilio Alves

Usando MATLAB

A aula de hoje será de estudo dirigido. Para isso você precisará do MATLAB®, ferramenta disponível na máquina virtual que você está logado.

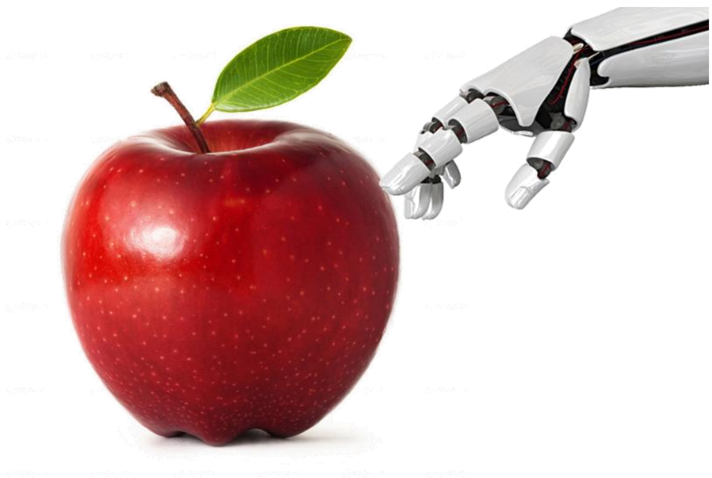
Nas próximas páginas você encontrará nove exercícios, alguns resolvidos, outros parcialmente resolvidos e alguns você terá que encontrar a saída.

Não se preocupe em criar nenhum arquivo com as respostas, pois os exercícios não deverão ser entregues aos professores. A aula é de estudo, não de teste. Suas anotações serão somente para você.

Ao analisar um exercício **com solução**, seja detalhista. Não deixe de entender nenhuma linha do código. Isso será imprescindível para que você consiga fazer os exercícios **sem solução**.

A lista é longa e o tempo curto... Não se distraia com conversas paralelas.

Boa aula!



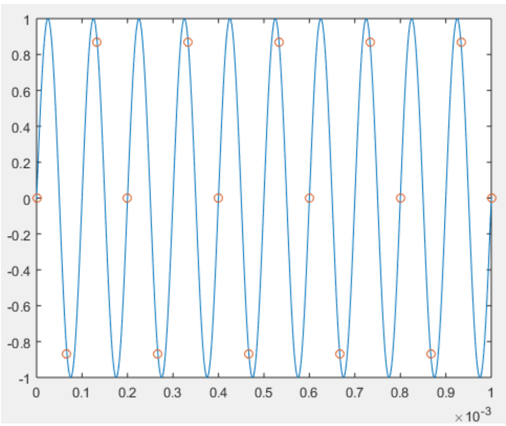
Exercícios

EXERCÍCIO 01

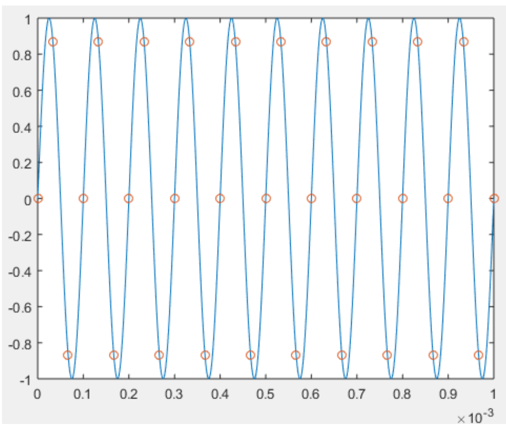
Vamos produzir uma senoide no MatLab e amostrá-la. Aproveite esse exercício para plotar gráficos. Explore especialmente a influência da taxa de amostragem no resultado do sinal amostrado (vetor `sinal_sample`), comparando-o com o sinal contínuo (vetor `sinal`).

```
clear all;
close all
clc
%% dados do sinal
f = 10000;%Freq entrada Hz
fs = 15000;% Frequencia de amostragem Hz
%% gerar sinal
tempo = [0:1/(100*f):10/f];%Tempo amostr ou num amostra
sinal = sin(2*pi*f*tempo); % Geração onda senoidal
%% plotar sinal
plot(tempo,sinal)
hold;
%% sinal amostrado
Ts = 1/fs;
N=31;
n = [0:1:30];
t_sample = [0 : Ts : n(31)*Ts];
sinal_sample = sin (2*pi*f*t_sample);
plot(t_sample, sinal_sample,'o');
set(gca,'FontSize',16)
xlabel('$n$','Interpreter','LaTex','FontSize',18)
ylabel('$x[n]$','Interpreter','LaTex','FontSize',18)

figure(2);
stem(n*Ts*1000,sinal_sample,'r'); % Grafico dos pontos amostrais no tempo
hold on;
plot(n*Ts*1000,sinal_sample); % Grafico do "sinal continuo" no tempo
set(gca,'FontSize',16)
xlabel('$t [ms]$','Interpreter','LaTex','FontSize',18)
ylabel('$x(t)$','Interpreter','LaTex','FontSize',18)
```

Nessas linhas você define o número de pontos amostrados `N`, frequência de amostragem `fs` e, como consequência, o intervalo de tempo do sinal amostrado `n(31)*Ts`. Mantendo `N` constante e aumentando `fs` você diminui o `n(31)*Ts` e vice versa.



Sinal de 10 kHz com taxa de 50kHz



Sinal de 10 kHz com taxa de 15kHz

EXERCÍCIO 02

FFT (FAST FOURIER TRANSFORM) é simplesmente uma forma mais rápida de calcular a DFT: A FFT utiliza alguns algoritmos que permitem reduzir o número de operações para $N\log_2 N$. Para utilizar a FFT, é necessário que o número de amostras seja uma potência de 2 – a FFT é executada mais rapidamente com um vetor cujo comprimento é uma potência de 2.

Para $N = 1000,\ DFT = 1\,000\,000,\ FFT = 10\,000$ operações

A FFT no Matlab: Matlab permite o cálculo fácil da DFT via FFT. Se tivermos um vetor A, de n elementos,

$$>>FFTdeA = fft(A);$$

Como exemplo, considera-se uma função cosseno,

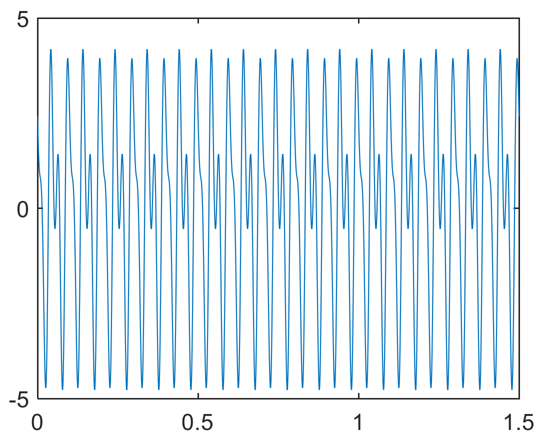
$$x(t) = 3\cos(2\pi f_1 t + 0.2) + \cos(2\pi f_1 t - 0.3) + 2\cos(2\pi f_1 t + 2.4)$$

Considere a frequência de amostragem $f_s = 1000$ amostras/segundo, amostrando-se em um período total de 1.5 ms.

Com esse exemplo, vamos entender a FFT...

```
%% Senoides e aliasing
% Frequencias, em Hz, do sinal
f1=20;
f2=30;
f3=40;
% Taxa de amostagem
fs=1000; Ts=1/fs;
%Sinal
N=31;
t=0:1/fs:1.5-1/fs;
x=3*cos(2*pi*f1.*t+0.2)+cos(2*pi*f2.*t-0.3)+2*cos(2*pi*f3.*t+2.4);
figure;
plot(t,x)
```

O sinal plotado no domínio do tempo será:

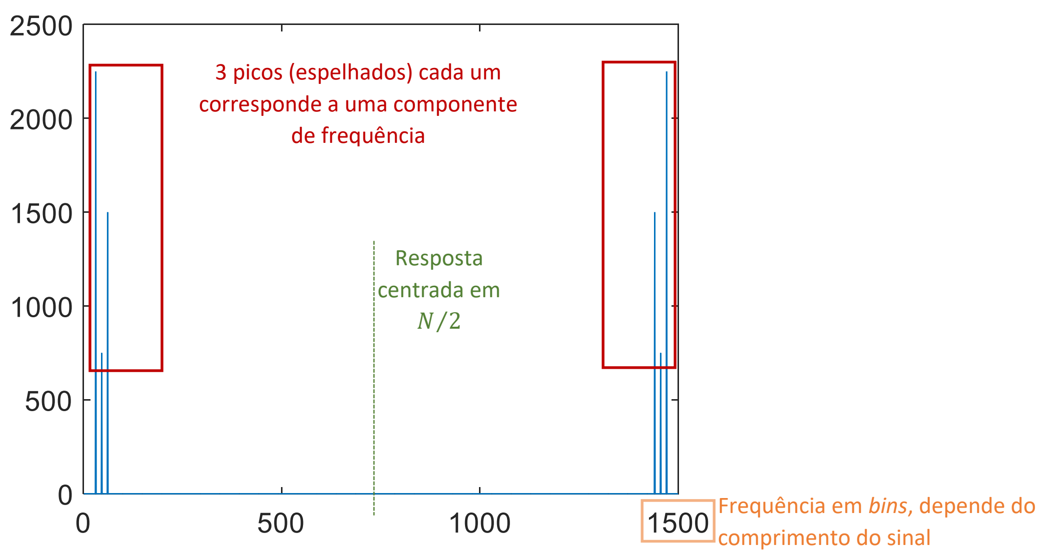


```
% os valores de x são valores reais, enquanto os valores de X são
% complexos. Veja alguns exemplos:
x(2:6)
X(30:34)
%X são valores complexos porque representam magnitude e fase!!!!
%Magnitude
figure;
X_mag=abs(X);
X_mag(30:34)
```

```
ans =

    2.0717    1.7535    1.4794    1.2572    1.0884


ans =

   1.0e+03 *

  -0.0000 - 0.0000i   2.2051 + 0.4470i   0.0000 - 0.0000i   0.0000 + 0.0000i  -0.0000 - 0.0000i

N =

        1500


ans =

        1500
```

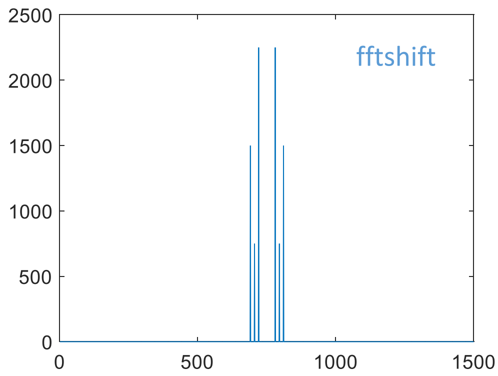
```
%X são valores complexos porque representam magnitude e fase!!!!
%Magnitude
figure;
X_mag=abs(X);
X_mag(30:34)
plot(X_mag)
set(gca,'FontSize',16)
```

```
ans =

   1.0e+03 *

    0.0000    2.2500    0.0000    0.0000    0.0000
```



Além disso, faz sentido analisar somente metade do sinal (lembre-se da frequência de Nyquist!!!).

```
%veja a diferença de se usar o comando fftshift
Y = fftshift(X)
Y_mag=abs(Y)
figure;
plot(Y_mag)
set(gca,'FontSize',16)
```

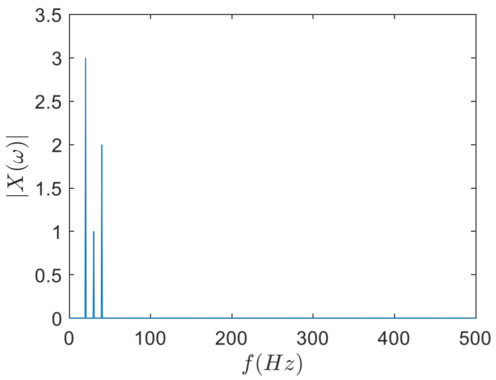


O eixo das abscissas corresponde a frequências, mas seu valor depende do comprimento do sinal. Os valores de X variam 0-1499, que é o tamanho do sinal criado por nós. Essas são chamadas de *bin frequencies*. Essa frequência em *bins* pode ser facilmente convertida em frequência em Hz,

$$f = F_{bins}\frac{f_s/2}{N/2}$$

No eixo das ordenadas, a magnitude está multiplicada também por $N/2$.

```
figure;
n=0:1:N-1
plot((fs/2)/(N/2).*n,X_mag/(N/2))
xlim([0 fs/2])
set(gca,'FontSize',16)
xlabel('$f(Hz)$','Interpreter','LaTex','FontSize',18)
ylabel('$|X(\omega)|$','Interpreter','LaTex','FontSize',18)
```



A fase do sinal também é uma informação importante, que pode ser tirada de X.

```
%Fase
X_fase=angle(X);
%se você verificar a fase em cada componente verá que
coincide com a
% fase do sinal
%
X_fase(31)
X_fase(46)
X_fase(61)
```

De onde saíram esses números, 31, 46 e 61? ☺

```
ans =

    0.2000


ans =

   -0.3000


ans =

    2.4000
```

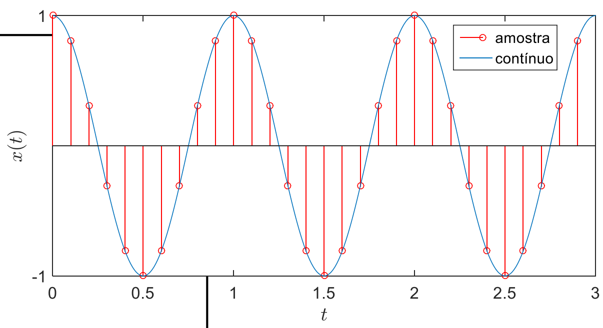
Exercício 3

O sinal de frequência 1Hz

$$x = \cos(2\pi t)$$

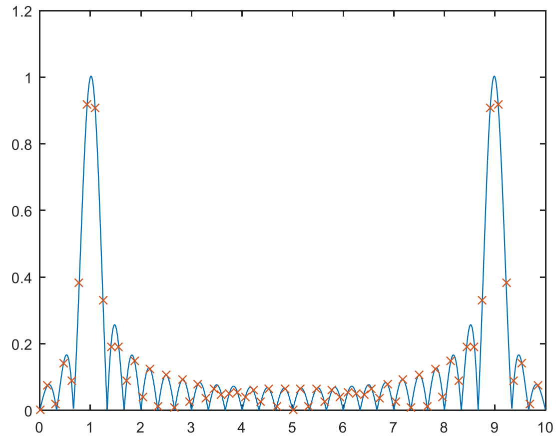
é amostrado a uma taxa de 10Hz, pelo período total de 3s (isto é, foram amostrados 3 períodos do sinal).

```
%% Senoide e fft - zero padding
N=30;
fs=10;
n = [0:N-1];
t=0:0.01:N/fs;
x_cont= cos(2*pi*t);
x = cos(2*pi*n/10);

figure;
stem(n.*1/fs,x,'r');
hold on
plot(t,x_cont);
set(gca,'FontSize',16)
xlabel('$t$','Interpreter','LaTex','FontSize',18)
ylabel('$x(t)$','Interpreter','LaTex','FontSize',18)
set(gca, 'YTick', [-1 1])
legend('amostra','contínuo')
```



Serão analisadas três situações:

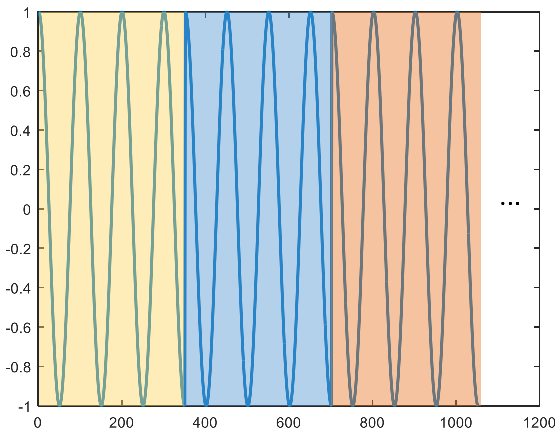
a. **Truque do Zero padding**: Entendendo diferentes valores de N na fft: `fft(x,Np)` - use os valores $N_p = 30, 64, 128, 256$ **mantendo a frequência de amostragem**. Avalie o resultado comparando as respostas. Para comparar as respostas, considere a resposta com $N_p = 1024$ como contínua e faça quatro gráficos, comparando a resposta contínua com aquelas com diferentes valores de N_p . Abaixo, como ilustração, o gráfico de comparação para $N_p = 64$. Responda: qual a influência de acrescentar zeros no final do sinal amostrado (*truque do zero-padding*)?



b. **Número de períodos de amostragem**: Mantendo o valor de $N_p = 1024$ constante, plote a fft para diferentes números de períodos amostrados. Você amostrou 3 períodos no item (a). Compare essa resposta com a resposta obtida amostrando 6, 9 e 12 períodos. Existe uma maneira rápida de aumentar o número de períodos no MatLab,

```
%% Aumentando o número de períodos
x1 = cos(2*pi*n/10); % 3 periodos
x2 = [x1 x1]; % 6 periodos
x3 = [x1 x1 x1]; % 9 periodos
x4 = [x1 x1 x1 x1]; % 12 periodos
```

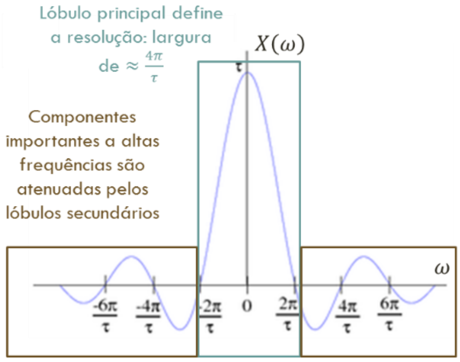
c. Se a amostragem do sinal não cobrir ciclos inteiros, o espectro calculado irá apresentar o fenômeno de vazamento espectral ("leakage"), que se traduz no seguinte erro básico: as amplitudes calculadas sofrem um achatamento e um espalhamento em torno das raias espectrais originais. Veja na figura abaixo a ilustração de como o sinal "é visto" pela transformada. Discuta o fenômeno de *leakage*, definindo períodos não completos de amostragem do exemplo.



```
%% Senoide e fft - leakage
N=35;
fs=10;
n = [0:N-1];
t=0:0.01:N/fs;
x_cont= cos(2*pi*t);
x = cos(2*pi*n/10);
%% Aumentando o número de períodos
x1 = cos(2*pi*n/10); % 3 periodos
x2 = [x1 x1]; % 6 periodos
x3 = [x1 x1 x1]; % 9 periodos
x4 = [x1 x1 x1 x1]; % 12 periodos
x_cont3=[x_cont x_cont x_cont];
figure;
plot(x_cont3,'Linewidth',2);
```

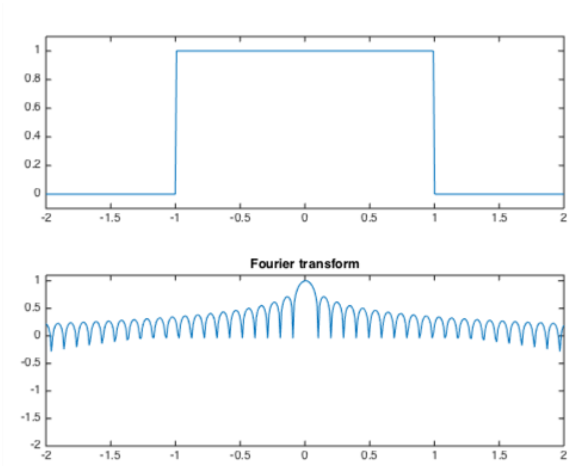
Exercício 4

Uma solução para o *leakage* é o janelamento (windowing). Diferentes tipos de janelas podem ser utilizados. A mais simples é a retangular, que é igual a 1 durante o intervalo de tempo que se pretende analisar, e igual a zero fora desse intervalo. Já aprendemos que a janela retangular no domínio do tempo, resulta em uma função *sinc* no domínio da frequência (figura ao lado).



Quanto mais estreito for o lóbulo principal, melhor a resolução frequencial. No entanto, quanto mais estreito o lóbulo principal, mais altos se tornam os lóbulos laterais, que aparecem como ruído de fundo no espectrograma.

A janela retangular fornece boa resolução frequencial, mas os lóbulos laterais são muito altos, resultando em muito ruído de fundo. Além disso, ocorre o fenômeno de leakage, discutido anteriormente, se os períodos de amostragem não forem completos.



Existem várias janelas, além da retangular, cujo objetivo é diminuir a influência dos extremos da amostragem. Veja, por exemplo, a janela de Hanning (também chamada de Hann), em homenagem ao vienense Julius Ferdinand von Hann (1839-1921). A janela Hanning é modelada como:

$$w(k) = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi k}{M-1}\right)\right] \qquad k = 0, \dots, M-1$$

No MatLab,

$w \ = \ hanning(M);$ ou $w \ = \ hann(M);$

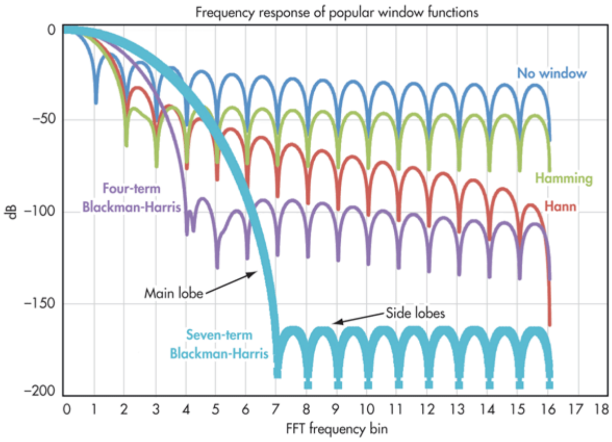
O janelamento Hamming começa em 0,08, sobe para 1 no meio do período, e depois cai novamente até 0,08 no final.

$$w(k) = 0{,}54 - 0{,}46\cos\left(\frac{2\pi k}{M-1}\right), \quad k = 0, \dots, M-1$$

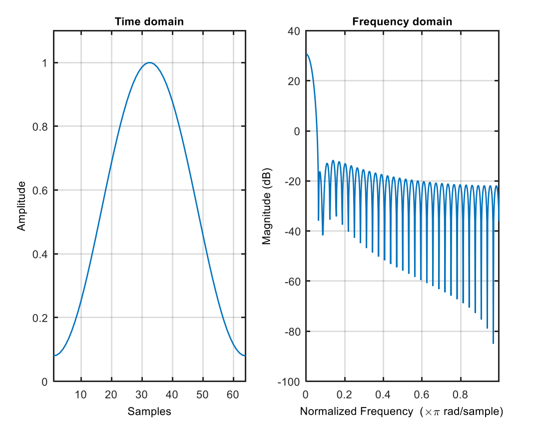
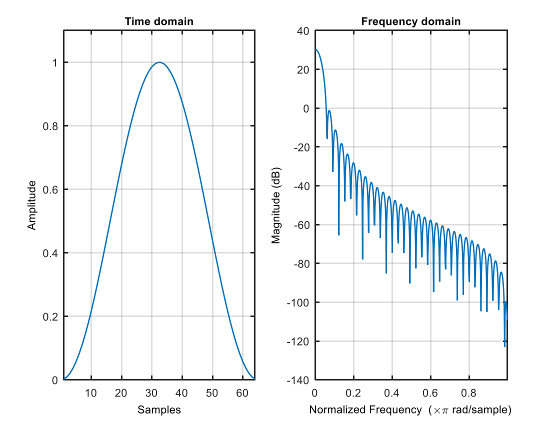
Ou seja, os valores iniciais e finais da amostragem são atenuados. No MatLab,

$$w \ = \ hamming(M);$$

A figura abaixo ilustra o comportamento de algumas janelas conhecidas.



Use os seguintes comandos para entender as janelas Hamming e Hanning,

`L = 64;` `wvtool(hamming(L))`	`L = 64;` `wvtool(hanning(L))`
	

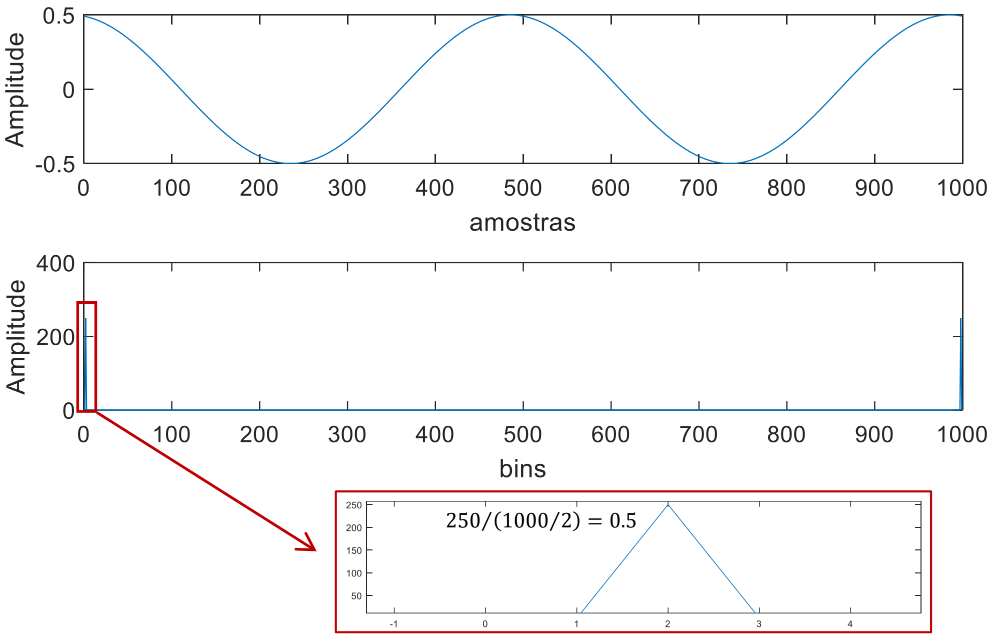
Exercício 5

Considere o sinal,

$$x = 0.5\cos(2\pi f_1 n + 0.2)$$

Segundo o código MATLAB abaixo,

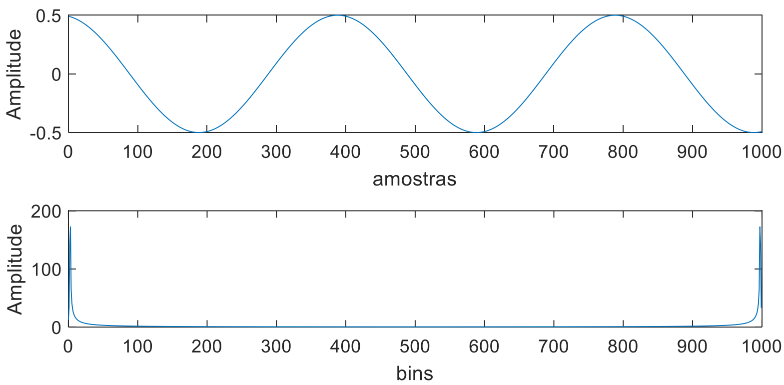
```
fs=1000;
t=0:1/fs:1-1/fs;
f1=2;
x1=0.5*cos(2*pi*f1*t+0.2);
subplot(2,1,1)
plot(x1)
set(gca,'FontSize',14)
xlabel('amostras','FontSize',16)
ylabel('Amplitude','FontSize',16)
%
X1=fft(x1);
subplot(2,1,2)
plot([0:length(X1)-1],abs(X1))
set(gca,'FontSize',14)
xlabel('bins','FontSize',16)
ylabel('Amplitude','FontSize',16)
```

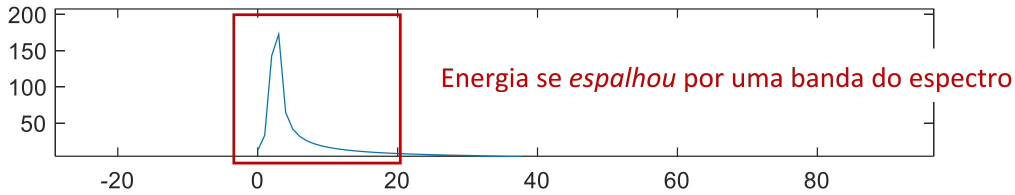


Pico em 2.0 bins (coincide com $2.0 * \frac{500}{500} = 2.0\ Hz$)

Se mudarmos a frequência de 2 para 2.5Hz, a amostragem não será de período inteiro e a resposta muda radicalmente.

```
fs=1000;
t=0:1/fs:1-1/fs;
f1=2.5;
x1=0.5*cos(2*pi*f1*t+0.2);
subplot(2,1,1)
plot(x1)
set(gca,'FontSize',14)
xlabel('Amostras','FontSize',16)
ylabel('Amplitude','FontSize',16)
%
X1=fft(x1);
subplot(2,1,2)
plot([0:length(X1)-1],abs(X1))
set(gca,'FontSize',14)
xlabel('bins','FontSize',16)
ylabel('Amplitude','FontSize',16)
```

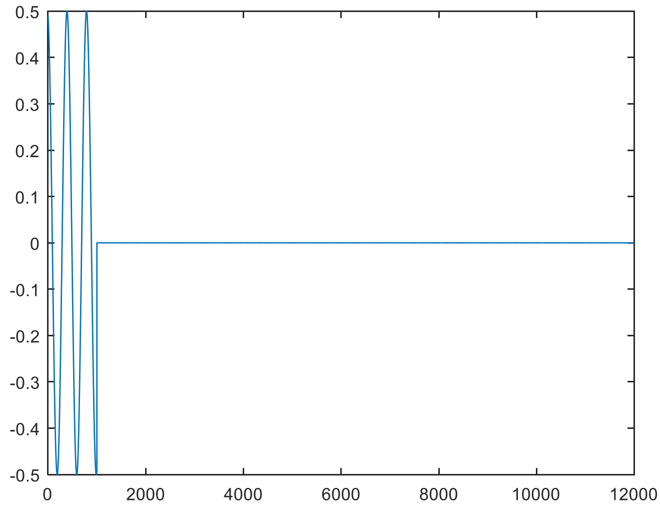


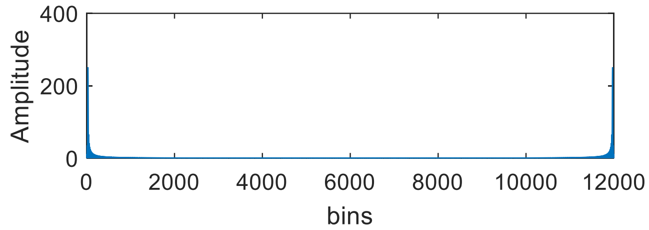


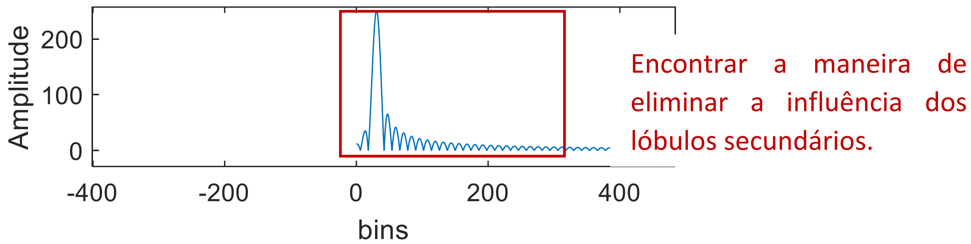
O uso do truque *zero padding* já entendemos que, nesse caso, não resolve. Veja que foi adicionado um número de zeros ao final do sinal (sem usar a variável N_p da fft) para termos controle do sinal.

```
x1=[x1 zeros(1,11000)];
X1=fft(x1);
subplot(2,1,2)
plot([0:length(X1)-1],abs(X1))
```

Nesse caso, nosso sinal será,

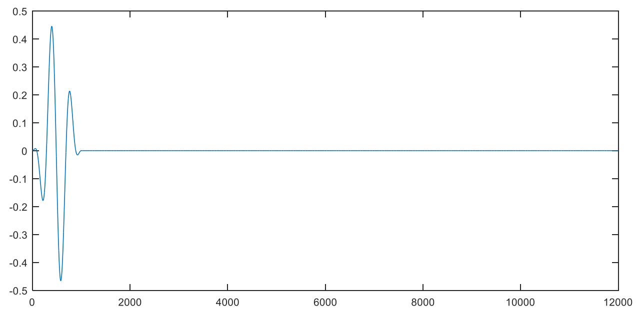


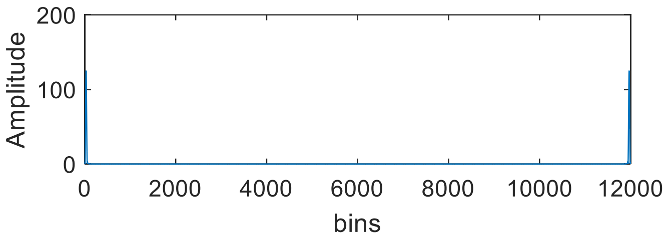


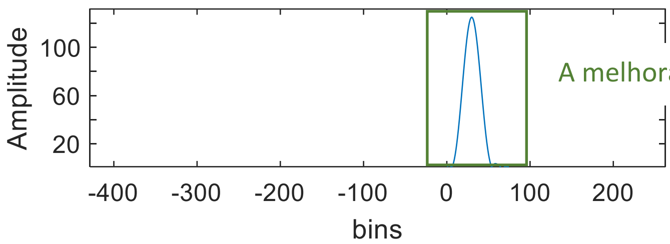


Se usarmos o janelamento, antes do truque do zero-padding,

```
L=length(x1)
x1=x1.*hanning(L)';
x1=[x1 zeros(1,11000)];
X1=fft(x1);
subplot(2,1,2)
plot([0:length(X1)-1],abs(X1))
```

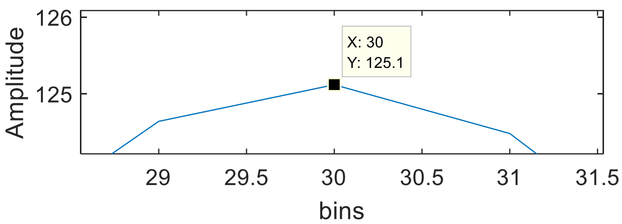






A melhora é significativa.

A magnitude é dada por,



Para janela Hanning, a regra é que a magnitude é o valor multiplicado por $N/4$, e a fase é o valor encontrado. A frequência segue a regra já definida:

```
angle(X1(31))
abs(X1(31))/(N/4)
30*fs/N
```

```
ans =

    0.2000

ans =

    0.5005


ans =

    2.5000
```

Exercício 6

Analise o sinal sinusoidal composto de três frequências,

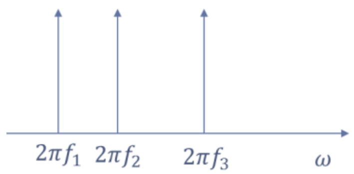
$$x = \cos(2\pi f_1 n) + \cos(2\pi f_2 n T_s) + \cos(2\pi f_3 n T_s)$$

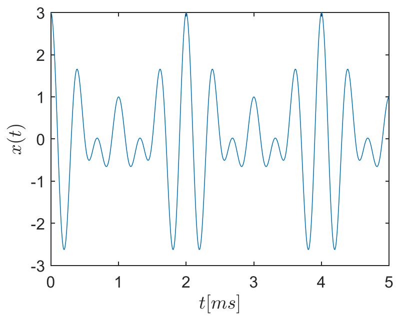
Onde,

$f_1 = 2000$ Hz

$f_2 = 2500$ Hz

$f_3 = 3000$ Hz





$f_s = 1000$ Hz, onde f_s é a taxa de amostragem, e $T_s = \frac{1}{f_s}$ é o período de amostragem.

Nesse exercício você deve definir uma frequência de amostragem constante, f_s, e utilizar um número diferente de amostras: $N = 10, 20, 40\ e\ 100$.

Utilize sempre as janelas retangular e hamming, conforme parte do código abaixo.

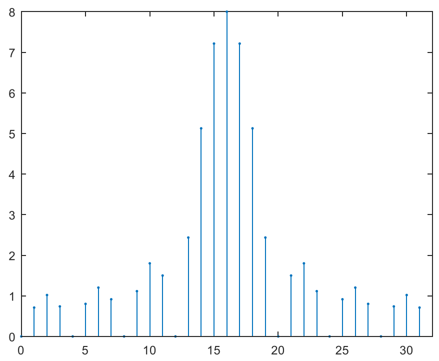
Analise a resposta das duas janelas a medida em que o período de amostragem (isto é, N) aumenta.

Discuta os resultados.

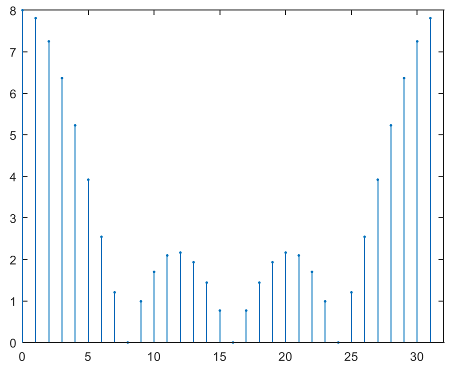
```
fs=10000; T=1/fs;  % Frequencia e período de amostragem
%Frequencias do sinal
f1=2e3;
f2=2.5e3;
f3=3e3;
% Frequencias para cálculo da DFT
w = 0:pi/1024:pi; %comprimento de w é usado em zero padding
figure(1)

L = 10;              % Define Numero de amostras
n=0:L-1;
x=cos(2*pi*f1*n*T)+cos(2*pi*f2*n*T)+cos(2*pi*f3*n*T); % Calcular x(n)*w(n)
X=fft(x,length(w));
subplot(221)
plot(w/pi,abs(X))
axis([0 1 0 50])
title('Rectangular Window -- L = 10')

 h = hamming(L);       % Hamming window
 xh=x.*h';
 Xh=fft(xh,length(w));
 subplot(222)
 plot(w/pi,abs(Xh))
 axis([0 1 0 50])
 title('Hamming Window -- L = 10')
```

Exercício 7

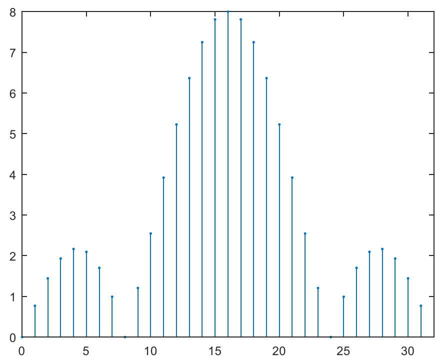
Ligue o código MATLAB com o respectivo gráfico. De preferência, não use o MatLab, use seu conhecimento…



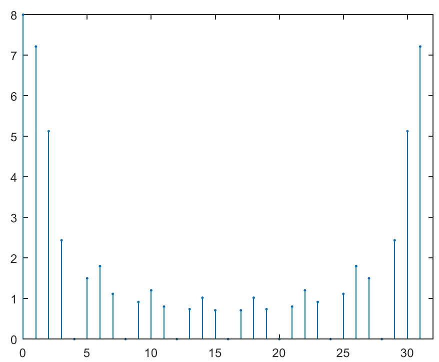

```
%% Codigo n0. 01
h=[1 1 1 1 1 1 1 1];
H=fft(h,32);
figure;
stem(0:31,abs(H),'.')
xlim([0 32])
```




```
%% Codigo n0. 02
h=[1 1 1 1 1 1 1 1];
H=fft(h,32);
figure;
stem(0:31,abs(fftshift(H)),'.')
xlim([0 32])
```



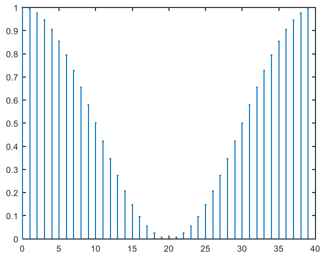

```
%% Codigo n0. 03
h=[2 2 2 2 0 0 0 0];
H=fft(h,32);
figure;
stem(0:31,abs(H),'.')
xlim([0 32])
```



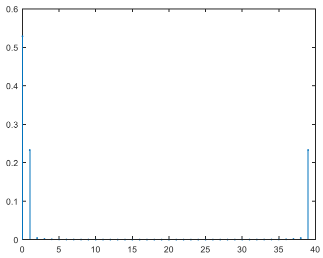

```
%% Codigo n0. 04
h=[2 2 2 2 0 0 0 0];
H=fft(h,32);
figure;
stem(0:31,abs(fftshift(H)),'.')
xlim([0 32])
```

Exercício 8

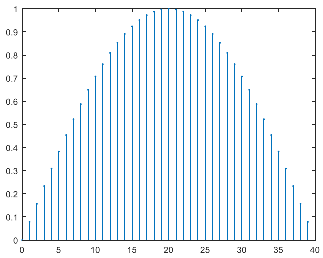
Ligue o código MATLAB com o respective gráfico. De uma breve explicação de sua escolha.



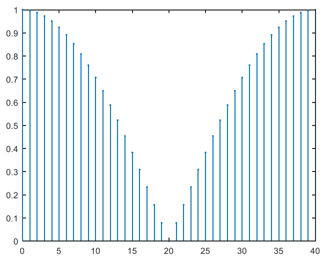
```
%% Codigo no. 1
N=40; n=0:N-1;
h=hamming(N)/N;
H=fft(h,N);
figure;
stem(n,abs(H),'.')
```



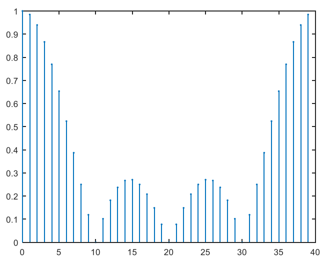
```
%% Codigo no. 2
N=40; n=0:N-1;
h=[0.5 0.5];
H=fft(h,N);
figure;
stem(n,abs(H),'.')
```



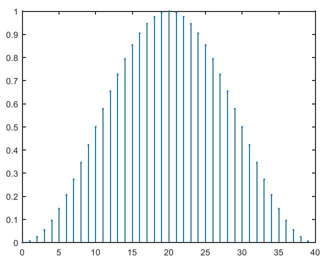
```
%% Codigo no. 3
N=40; n=0:N-1;
h=[0.5 -0.5];
H=fft(h,N);
figure;
stem(n,abs(H),'.')
```



```
%% Codigo no. 4
N=40; n=0:N-1;
h=[0.25 0.5 0.25];
H=fft(h,N);
figure;
stem(n,abs(H),'.')
```



```
%% Codigo no. 5
N=40; n=0:N-1;
h=[-0.25 0.5 -0.25];
H=fft(h,N);
figure;
stem(n,abs(H),'.')
```



```
%% Codigo no. 6
N=40; n=0:N-1;
h=ones(1,4)/4;
H=fft(h,N);
figure;
```

Exercício 9

Os arquivos '.wav' na pasta *NotasMusicais* deveriam estar numeradas em ordem crescente de frequência, que coincide com as notas musicais **Dó(261.63 Hz), Ré(293.66 Hz), Mi(329.63 Hz), Fá(349.23 Hz), Sol (392 Hz), Lá (440 Hz), Si (493.88 Hz)**. Alguém não fez o trabalho de forma adequada. Sua tarefa consiste em criar um programa Matlab® que reconheça a sequência de notas, de forma a completar a tabela:

Nota musical	Nome do arquivo
Dó	
Ré	
Mi	
Fá	
Sol	
Lá	
Si	

Use os comandos `audiowrite` e `audioread`, do MATLAB.