

1.3 Gráficos 2D e 3D

Prof. Dr. Sidney Bruce Shiki

E-mail: bruce@ufscar.br

Prof. Dr. Vitor Ramos Franco

e-mail: vrfranco@ufscar.br

UFSCar – Universidade Federal de São Carlos

DEMec - Departamento de Engenharia Mecânica

- Introdução
- Gráficos 2D
 - Gráficos x-y elementares
 - ❖ Comando “plot”
 - Gráficos x-y especializados
- Gráficos 3D
- Exercícios

- Na maioria dos casos, é necessário construir uma figura para mostrar os resultados
 - A figura tem que ser apresentável!!!
- Aqui, o aluno aprenderá:
 - gerar diferentes formas de figuras/gráficos;
 - formatar a figura na linha de comandos,
 - salvar a figura para ser utilizada posteriormente.
- É importante lembrar que existem diversas maneiras de se formatar uma figura.

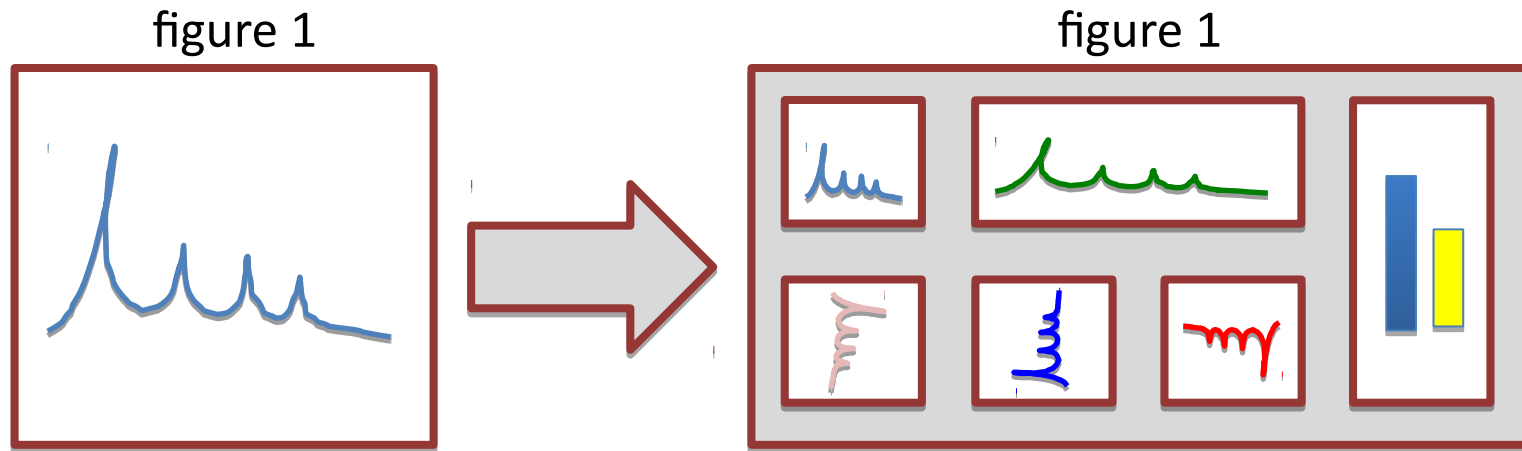
- Primeira coisa a se fazer: gerar a figura (área)

`figure(1)` o nº da figura vai se alterando

- Uma vez criada a figura, o comando de geração pode ser aplicado, podendo ser um gráfico 2D ou 3D.
- Para se gerar mais de uma “curva” na mesma figura, utiliza-se o comando

`hold on` mantém a figura disponível para mais gerações

- Algo bastante utilizado na apresentação de resultados é aproveitar a mesma figura para mostrar sub-figuras (internas)



- O comando para este fim é:

```
subplot(x, y, z)
```

transforma a área da figura em
uma matriz de posições de figuras

x = nº de linhas; y = nº de colunas; z = posição da sub-figura

– Exemplos de criação:

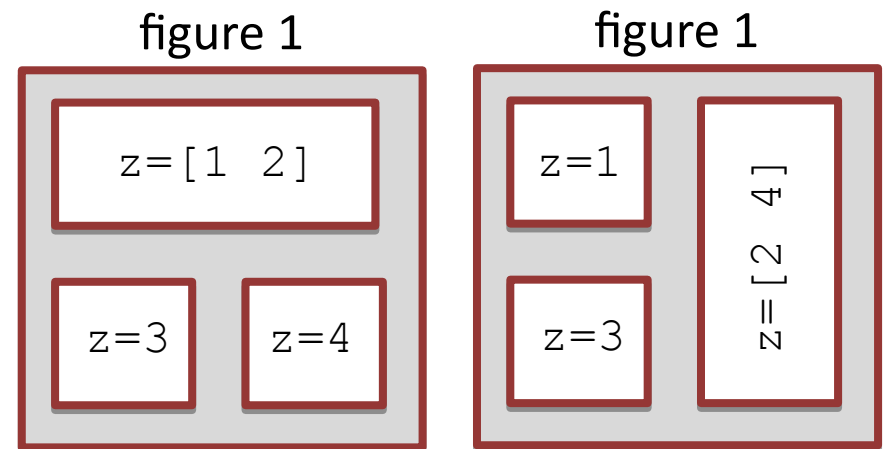
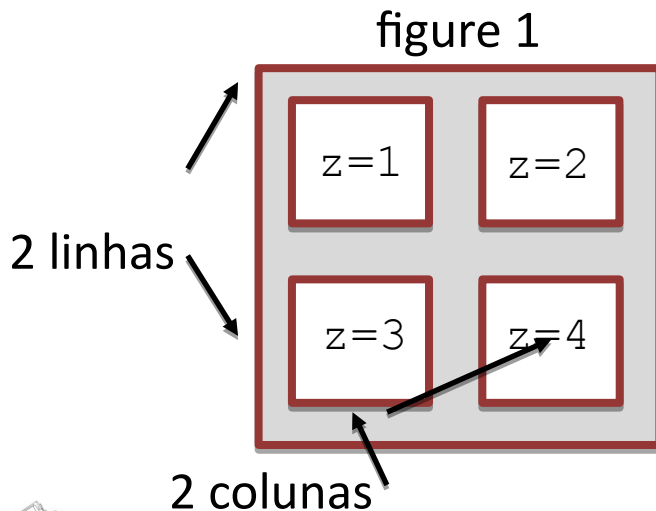
```
subplot(2,2,1)
% gere a sub-figura aqui
subplot(2,2,2)
% gere a sub-figura aqui
subplot(2,2,3)
% gere a sub-figura aqui
subplot(2,2,3)
% gere a sub-figura aqui
```

```
subplot(x,y,z)
```

❖ Para agrupar sub-figuras:

```
subplot(2,2,[1 2])
```

```
subplot(2,2,[2 4])
```



- Para salvar a figura, utiliza-se:

```
saveas(n°_da_figura, 'nome_da_figura.jpg')
```

- Diferentes formatos de figuras podem ser gerados: tif, jpg, eps, pdf, png, etc.
- A qualidade da figura e tamanho do arquivo estão diretamente relacionados ao formato.

- Outro comando para salvar figura é:

```
print(...) => Print figure or save to file
```

- Gráficos x-y elementares
 - Os mais utilizados são:

Descrição	Comando
Gráfico linear	<code>plot(x, y)</code>
Gráfico di-log (escalas x e y logaritmicas)	<code>loglog(x, y)</code>
Gráfico mono-log (escala x logaritmica)	<code>semilogx(x, y)</code>
Gráfico mono-log (escala y logaritmica)	<code>semilogy(x, y)</code>

- Foco no mais utilizado de todos: “plot”
- Exemplo de criação:

```
t = linspace(0, 5, 2^10);  
y = sin(2*pi*2*t);  
figure(1)  
plot(t, y)
```

Alterações:

- Estilo de linha
- Anotações
- Dados do gráfico

- Estilo de linha

O MATLAB permite escolher o tipo de linha, cor e símbolo utilizado na construção do gráfico

- Exemplo de criação:

```
figure(2)  
plot(t,y,'r.','linewidth',2,'markersize',20)
```

- Maiores informações sobre especificações de linha podem ser vistas digitando:

```
help LineSpec
```

- e clicando em:

```
reference page for LineSpec
```

○ Anotações

O MATLAB permite também criar várias anotações no gráfico, tais como: labels, legenda, título, texto, linhas, etc...

– Exemplo de criação:

```
figure(3)
y2 = 0.5*sin(2*pi*2*t);
plot(t,y,'b','linewidth',2)
hold on
plot(t,y2,'r','linewidth',2)
title('Variação temporal das Resistências R_1 e R_2')
xlabel('Tempo [s]')
ylabel('Resistência [\Omega]')
legend('R_1','R_2')
```

Texto também pode ser criado a fim de se anotar alguma informação adicional na figura

– Exemplo de criação:

```
% comando text  
% posiciona em (x,y  
text(1,0.8,'anotação')
```

Além de texto, linhas (retas) podem ser criadas em casos em que se necessitam de referências, faixas de variação, etc.

– Exemplo de criação:

```
% comando line  
% liga os pontos (x1,y1) e (x2,y2)  
% pontos nos vetores [x1 x2] e [y1 y2]  
line([0 5],[0 0],'color',[0 1 0])% alteração (percentagem)  
de cor no padrão RGB -em Red, Green e Blue
```

- Para melhorar a apresentação da figura, pode-se:
 - Alterar a fonte, tamanho da fonte, colocar fonte em negrito, itálico, etc.
 - Reposicionar a legenda (dentro ou fora) além de orientar a legenda (horizontal ou vertical);

```
title('Variação temporal das Resistências R_1 e  
R_2', 'fontname', 'times', 'fontweight', 'bold', 'fontsize', 14)  
xlabel('Tempo [s]', 'fontname', 'times', 'fontweight', 'bold',  
'fontsize', 14)  
ylabel('Resistência [\nOmega]', 'fontname', 'times', 'fontweight', 'bold',  
'fontsize', 14)  
legend('R_1', 'R_2',  
'location', 'northoutside', 'orientation', 'horizontal')
```

- Dados do gráfico

- também podem ser alterados de maneira semelhante a label, title, legend;
- são modificados usando o comando set(gca,...)

- Exemplo de criação:

```
set(gca, 'fontname', 'times', 'fontweight', 'bold',  
        'fontsize', 14, 'xgrid', 'on', 'ygrid', 'on')  
xlim([0 3]) % limita somente a visualização  
ylim([-1.5 1.5]) % limita somente a visualização
```

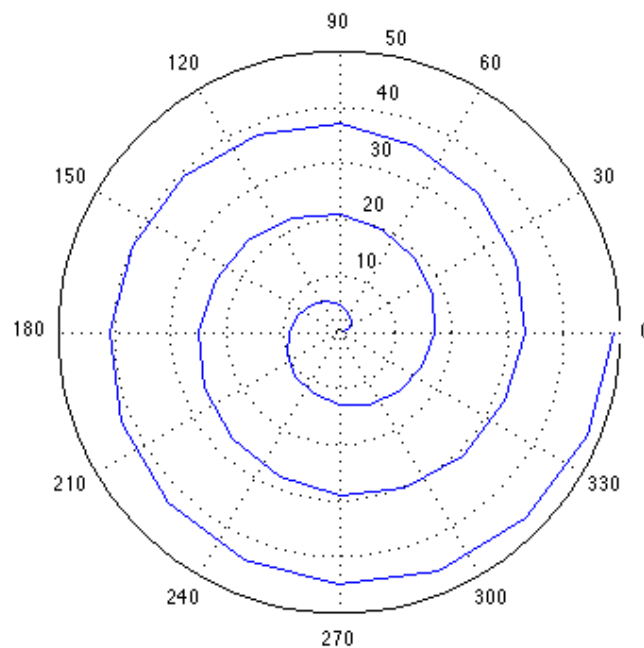
- OBS #1: o “label” (descrição das linhas) da legenda também se altera com esse comando;
- OBS #2: alterações também pode ser realizadas nos valores numéricos atribuídos aos eixos (exemplo usando “barras”);
- OBS #3: outra forma de colocar malha: `grid on, grid minor`

- Gráficos x-y especializados
 - Algumas outras maneiras gráficas de se mostrar resultados, dependendo do estudo a ser realizado.
 - Os mais utilizados são:

Descrição	Comando
Gráfico em coordenadas polares	<code>polar(theta, r)</code>
Gráfico de barras	<code>bar(x, y)</code>
Gráfico de sequência discreta	<code>stem(x, y)</code>
Gráfico degrau (escada)	<code>stairs(x, y)</code>
Histograma	<code>hist(y, N)</code>
Gráfico “pizza”	<code>pie(x)</code>
Gráfico de uma função	<code>fplot(name, lin)</code>
Gráfico de contorno (sombra)	<code>contour(Z)</code>

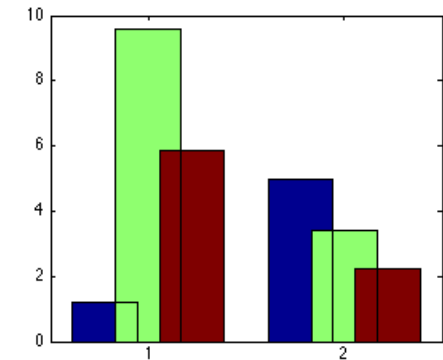
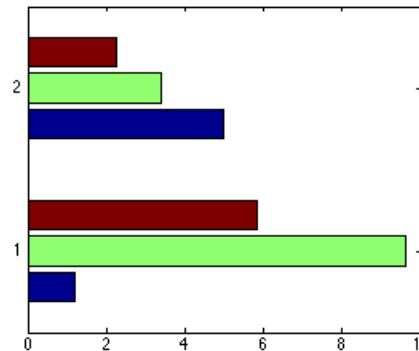
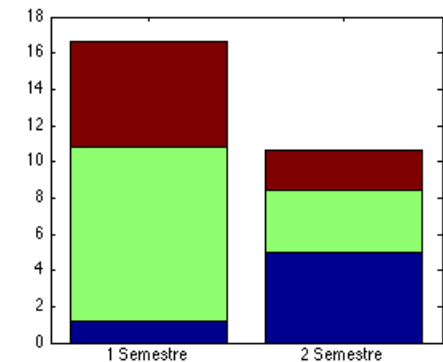
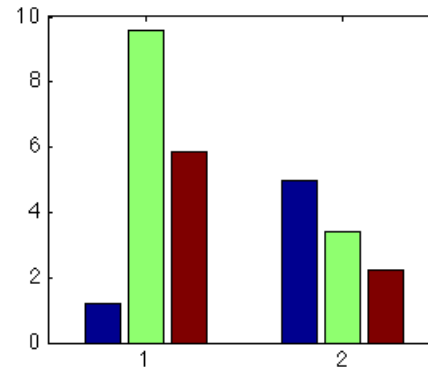
- Exemplos de criação de gráficos especializados
 - Gráfico polar:

```
theta=0:pi/8:6*pi;  
r=1:length(theta);  
figure(4)  
polar(theta,r)
```



– Gráfico de barras:

```
y=rand(2,3)*10;
figure(5)
subplot(2,2,1)
bar(y,'group')
subplot(2,2,2)
bar(y,'stack')
subplot(2,2,3)
barh(y,'group')
% barra na horizontal
subplot(2,2,4)
bar(y,1.5)
% largura da barra 1.5
```

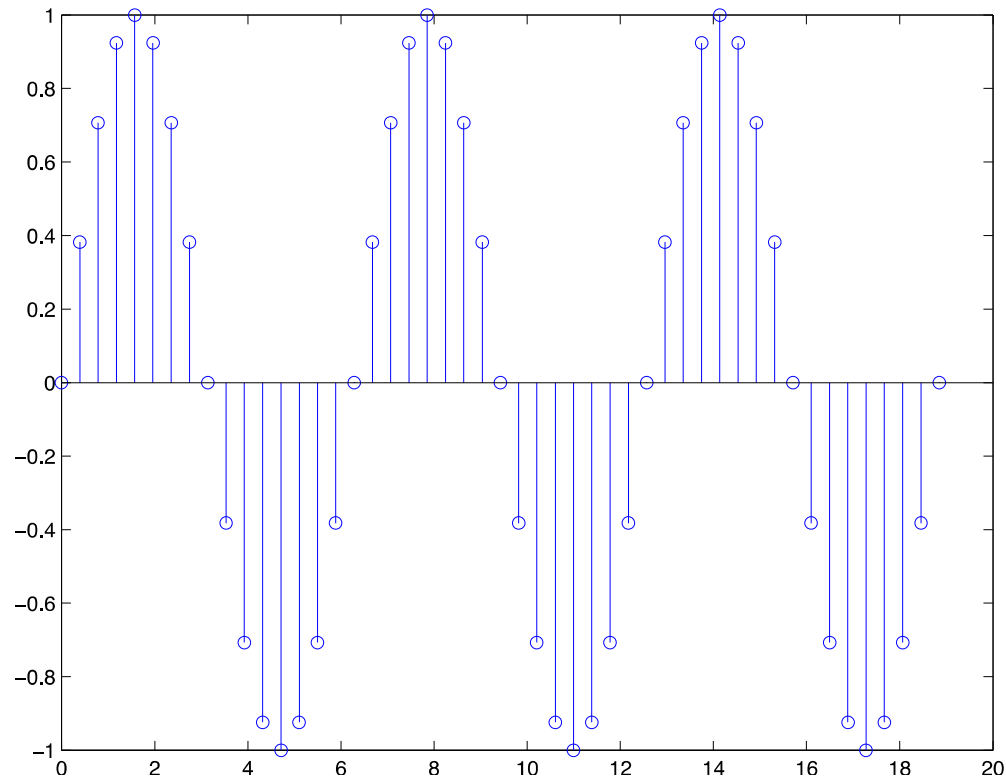


Alterações nos valores
numéricos atribuídos
aos eixos

```
periodo = {'1 Semestre','2 Semestre'};
set(gca,'xticklabel',periodo)
```

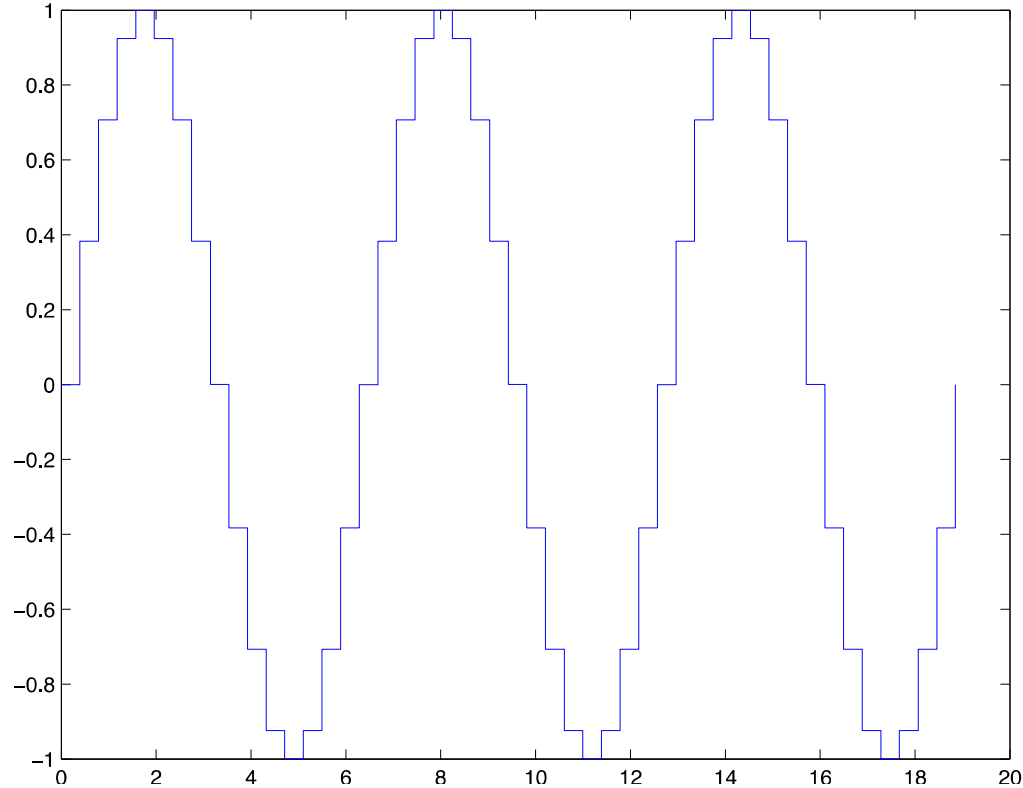

– Gráfico de sequência discreta:

```
x=0:pi/8:6*pi;  
y=sin(x);  
figure(6)  
stem(x,y)
```



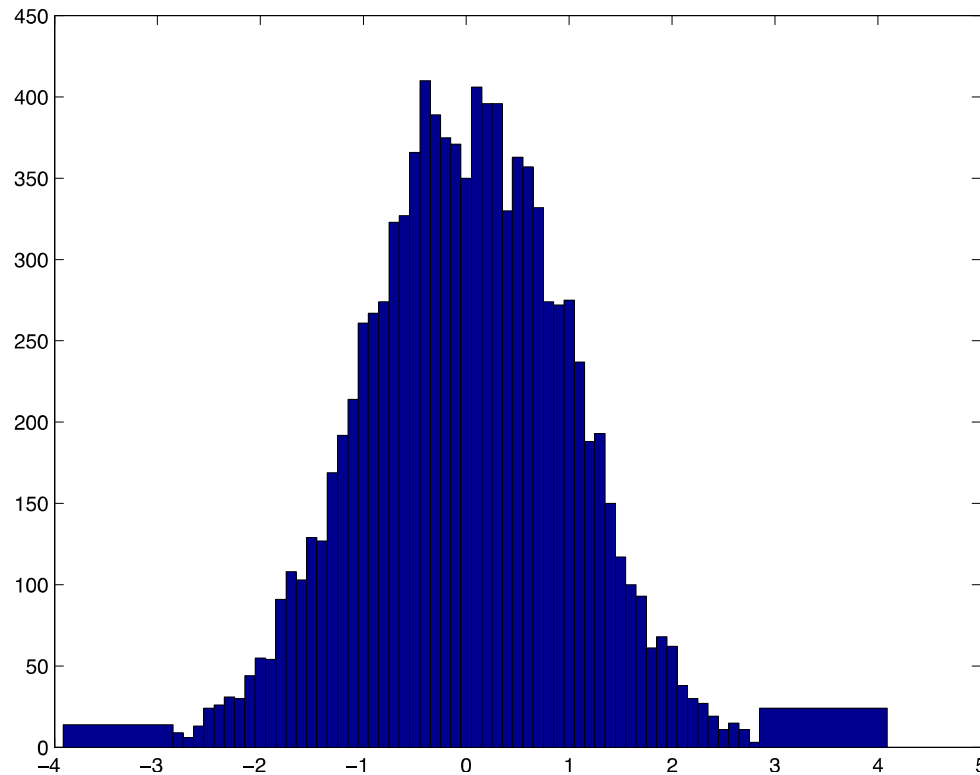
– Gráfico de degrau (escada):

```
x=0:pi/8:6*pi;  
y=sin(x);  
figure(7)  
stairs(x,y)
```



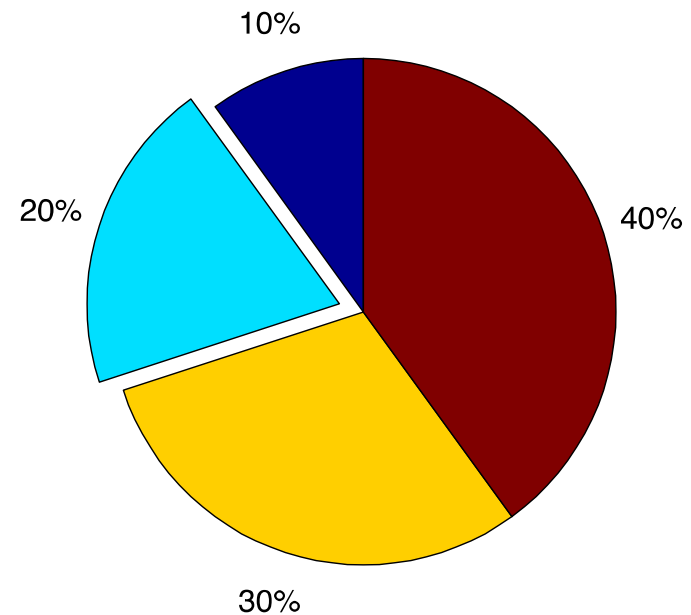
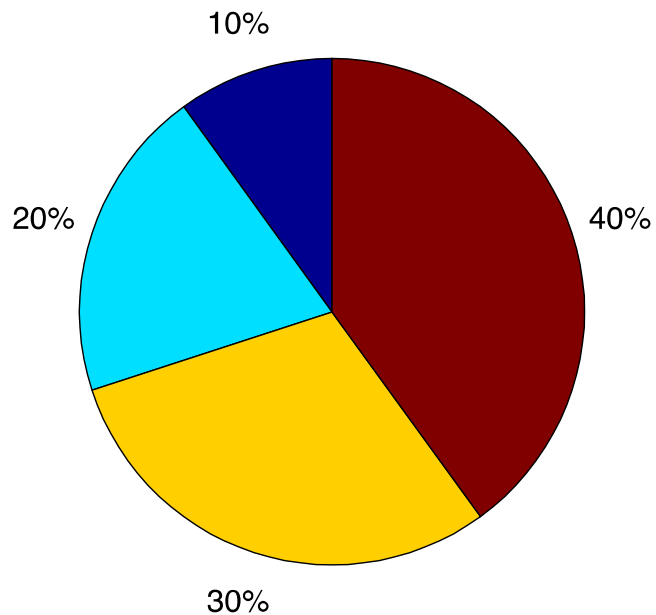
– Gráfico histograma:

```
x=-2.9:0.1:2.9;  
y=randn(10000,1);  
figure(8)  
hist(y,x)
```



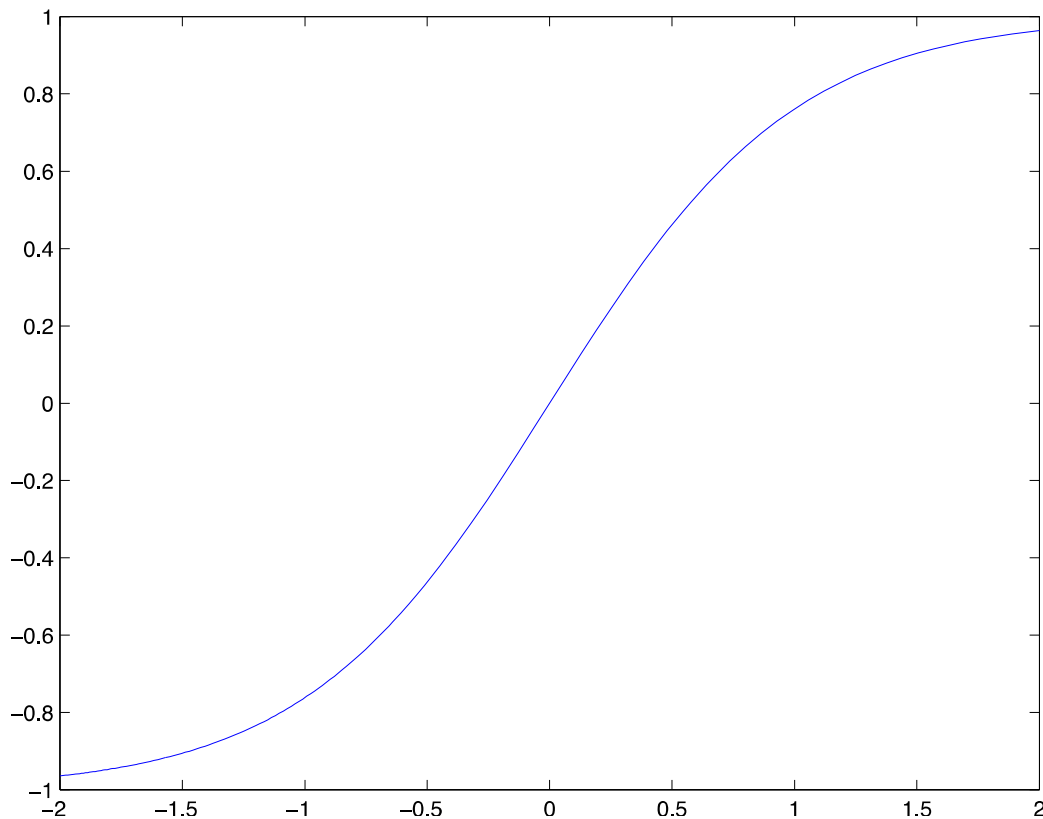
– Gráfico de degrau (escada):

```
x = [1 2 3 4];  
figure(9)  
subplot(1,2,1)  
pie(x)  
subplot(1,2,2)  
destaque = [0 1 0 0]; % destaca a fatia 2 de 10 (20%)  
pie(x,destaque)
```



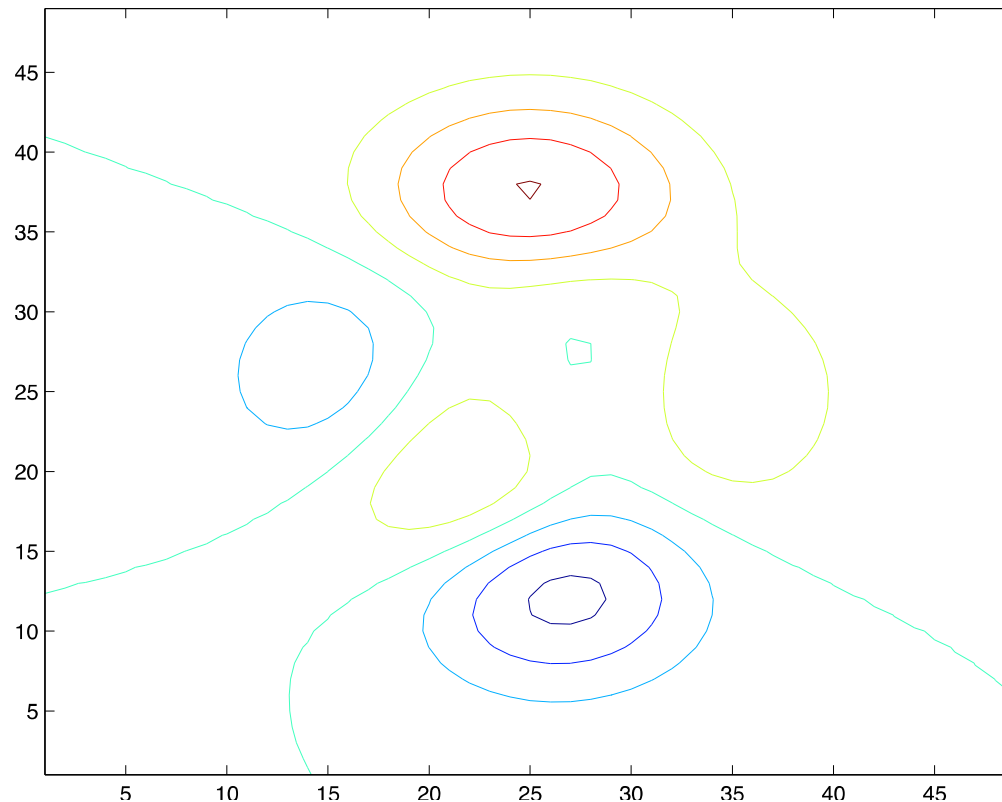
– Gráfico de uma função:

```
figure(10)  
fplot('tanh', [-2 2]) % tangente hiperbólica
```



– Gráfico de contorno (sombra):

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
figure(14)  
contour(Z)
```

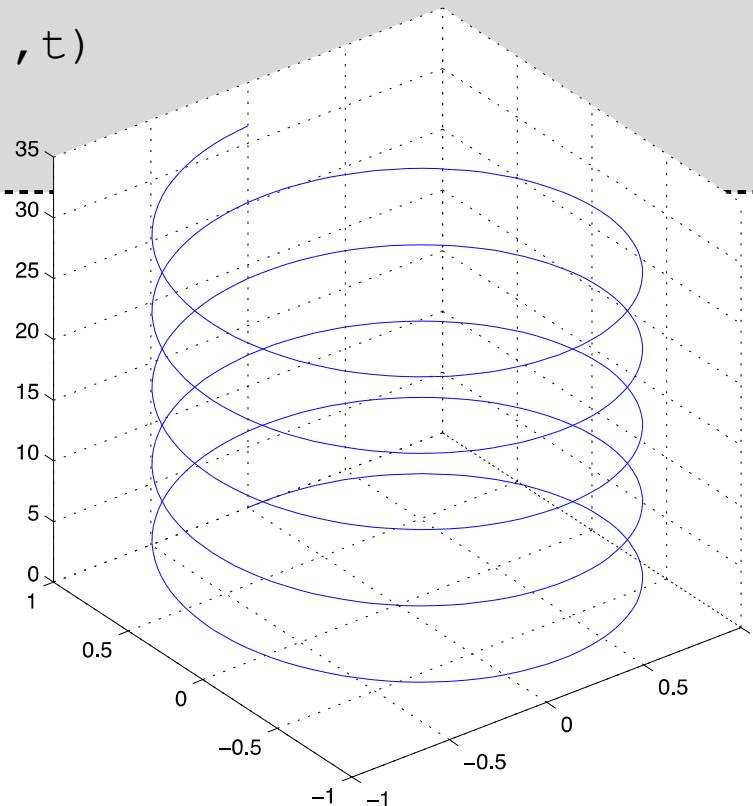




- Para gerar gráficos 3D, é necessário que os dados sejam agrupados de forma correta
 - Uma das formas é a matriz (2D ou 3D)
 - Algumas vezes, são necessários três vetores como entrada para o comando
 - A formatação das figuras são semelhantes ao utilizado em gráficos 2D

- Exemplos de criação de gráficos 3D
 - Plotando no espaço:

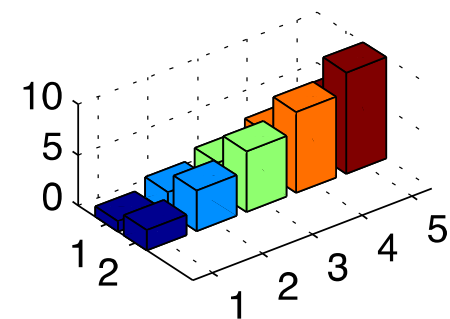
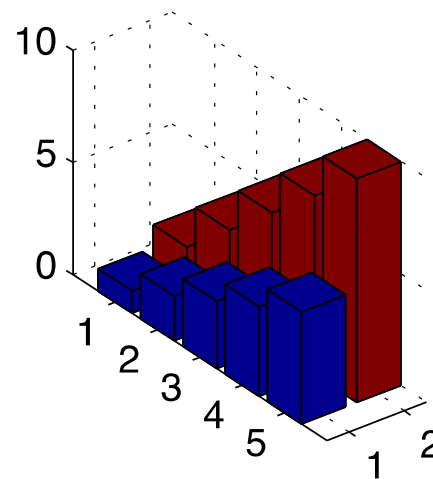
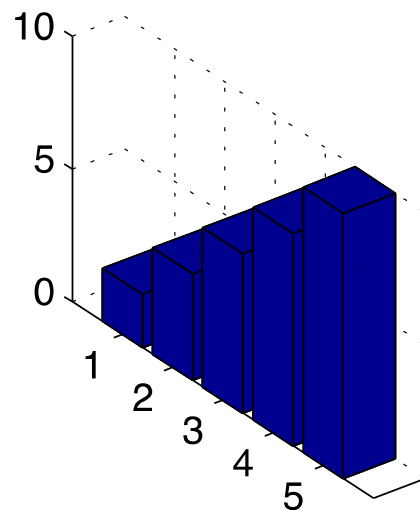
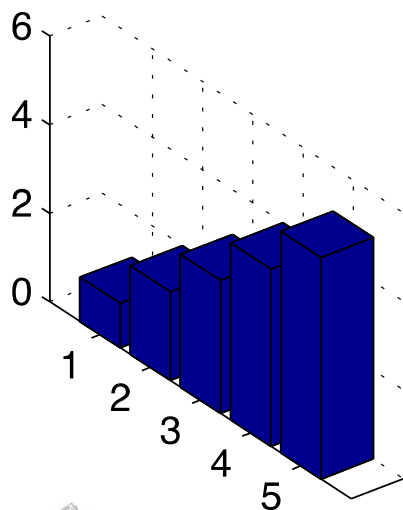
```
t=0:pi/50:10*pi;  
figure(11)  
plot3(sin(t),cos(t),t)  
grid on  
axis square
```



- Gráfico de “barras” 3D
 - Exemplo de criação:

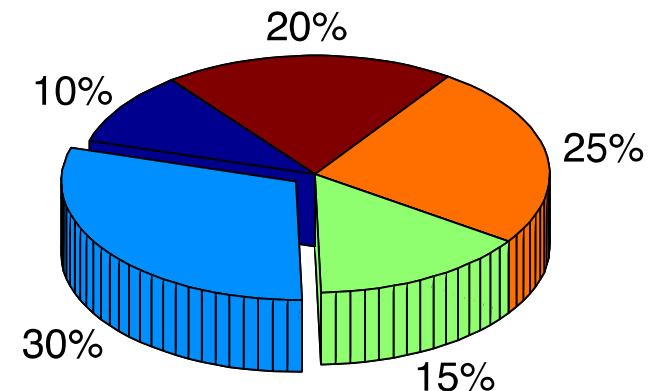
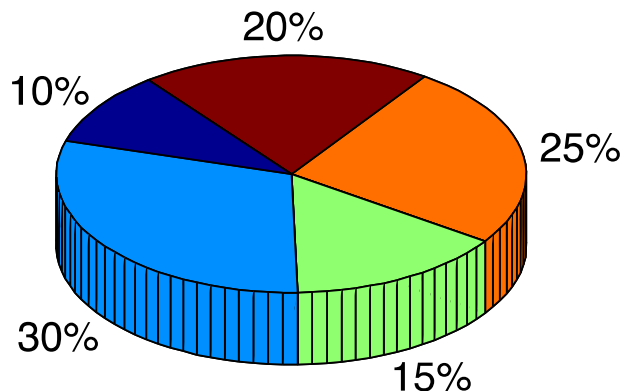
```
x = 1:5; y = 2*x; A = [x;y];  
figure(12)  
subplot(1,4,1)  
bar3(x)  
subplot(1,4,2)  
bar3(y)
```

```
subplot(1,4,3)  
bar3(A')  
subplot(1,4,4)  
bar3(A)
```



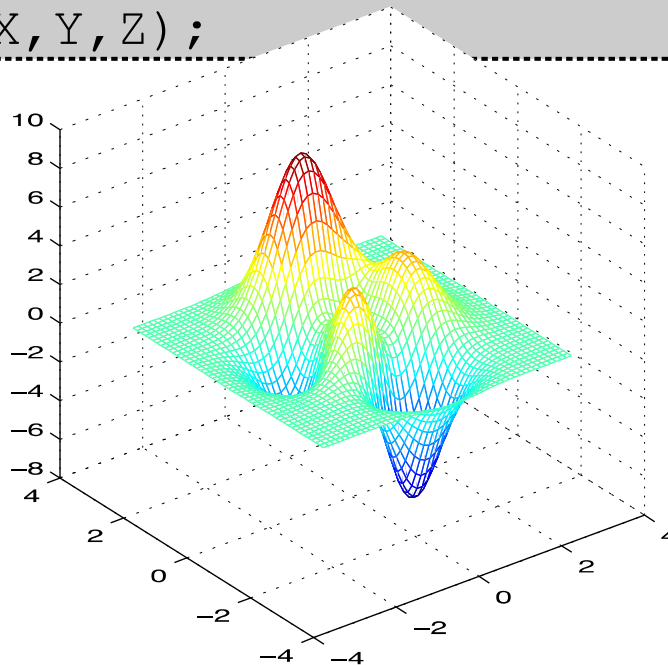
- Gráfico tipo “Pizza” 3D
 - Exemplo de criação:

```
dados = [1 3 1.5 2.5 2]; % soma=10 (100%)  
figure(13)  
subplot(1,2,1)  
pie3(dados);  
subplot(1,2,2)  
destaque = [0 1 0 0 0]; % destaca o 3 de 10 (30%)  
pie3(dados,destaque);
```

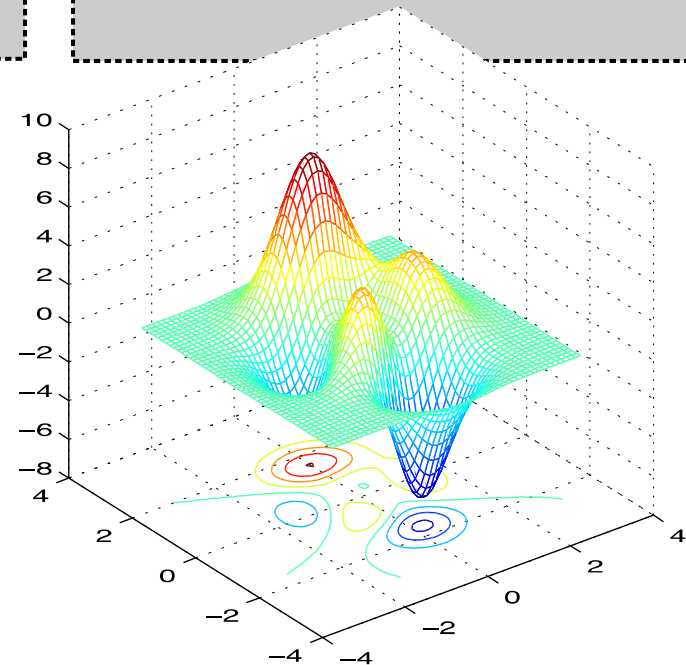


- Superfícies 3D
 - Exemplos de criação:

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
figure(14)  
subplot(1,2,1)  
mesh(X,Y,Z);
```

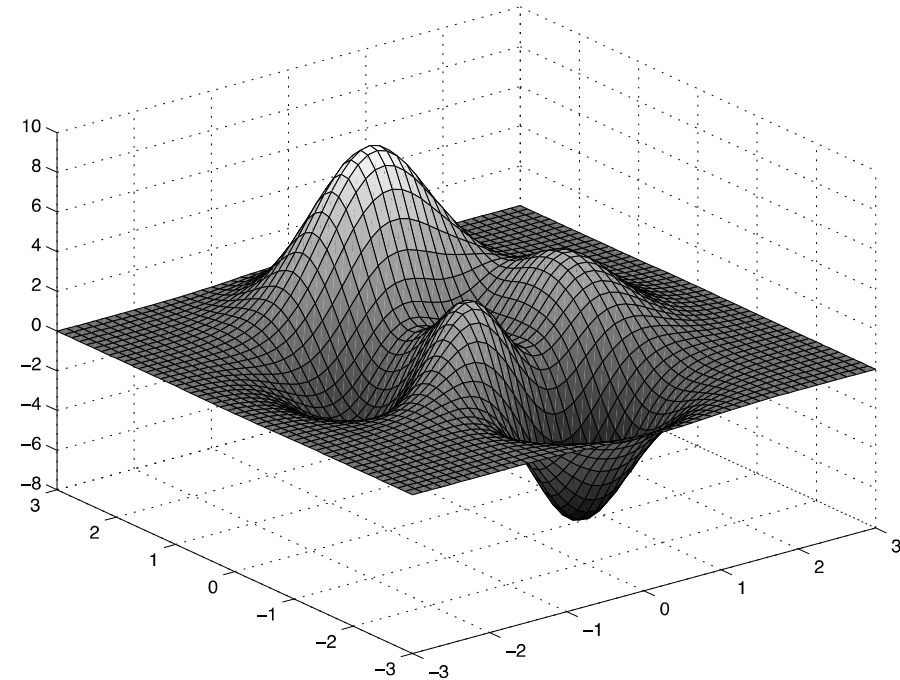
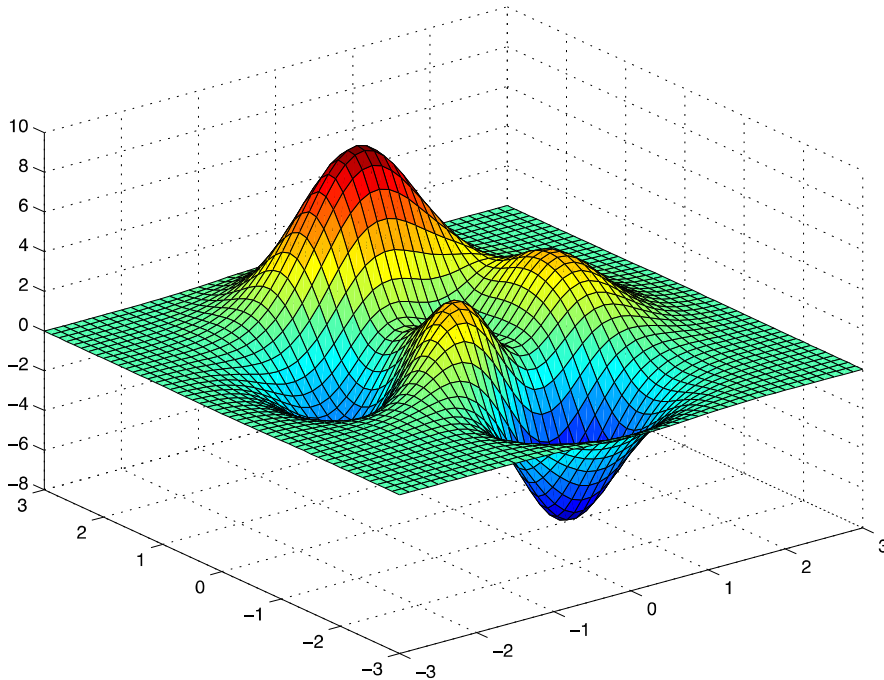


```
subplot(1,2,2)  
meshc(X,Y,Z);  
% com contorno
```

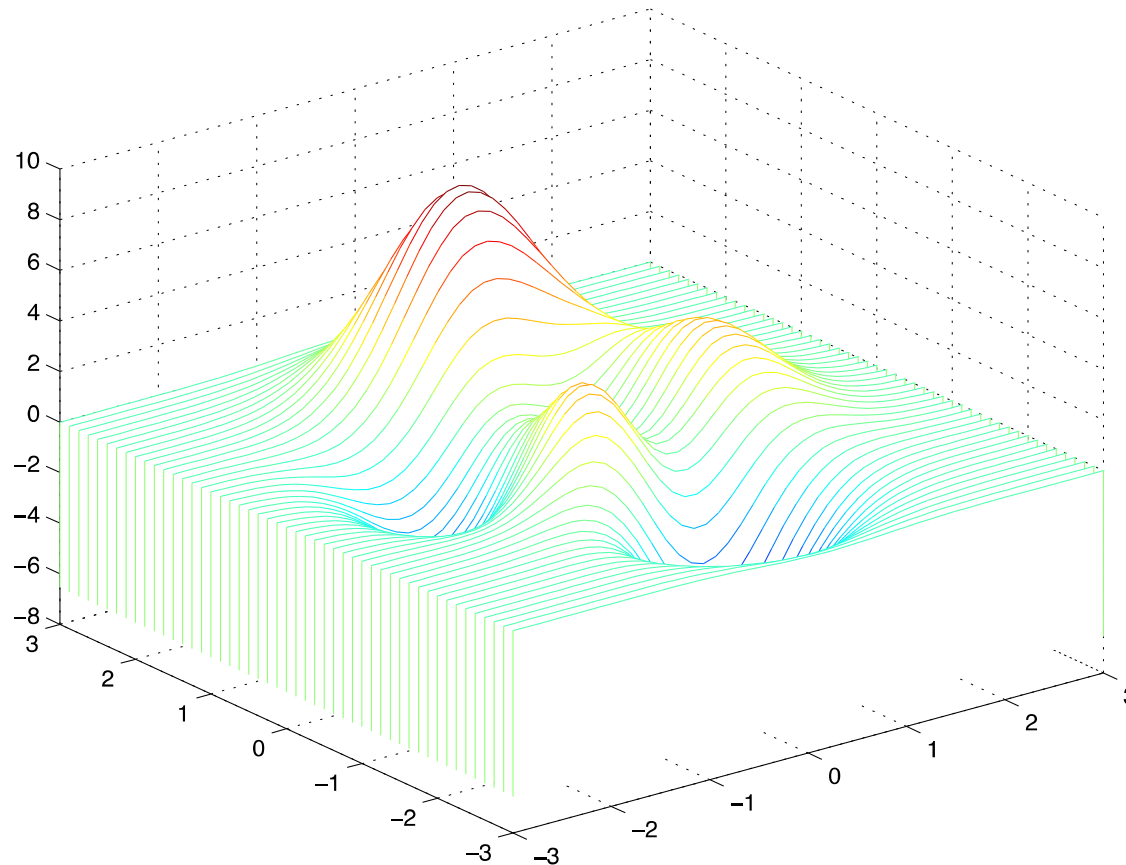


```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
figure(15)  
surf(X,Y,Z);
```

```
figure(16)  
surf(X,Y,Z);  
colormap(gray)
```



```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
figure(17)  
waterfall(X,Y,Z);
```





Crie:

- Um vetor x de $-\pi$ até π com 700 elementos;
- Um vetor:
$$y = x \sin \left(\frac{0.872\pi \sin(x)}{x} \right)$$
- Outro vetor:
$$z = -|x| \cos \left(\frac{\pi \sin(x)}{x} \right)$$
- Crie um gráfico tridimensional com linha vermelha e espessura 3 com os 3 vetores e coloquem alguma descrição nos eixos X, Y e Z com fonte de tamanho 16.

Exercício: vendas anuais de uma empresa



- Gere um vetor de números aleatórios contendo um elemento por mês do ano. O valor de cada um desses números corresponderá ao número de vendas multiplicados por 1000;
 - Apresente as vendas mensais de três formas diferentes na mesma figura (forma de barras, pizza, destacando para este caso o mês em que se vendeu mais e a outra fica a seu critério;
 - Formate a figura da maneira mais apresentável que puder;
 - Salve a figura em dois formatos diferentes, sendo um deles de forma vetorizada (.pdf ou .eps) e compare as figuras resultantes (tamanho e visual).
- Não use recursos como `if`, `for`, `while`, etc.

Perguntas ?