

SLCO4A - Aula Prática 2

Sumário

Plot com o MATLAB.....	1
Plotando em duas dimensões.....	1
Comando linspace e logspace.....	1
Plotando várias funções em uma figura.....	2
Configurando opções do plot.....	2
Comando hold.....	2
Formatando uma figura.....	2
Plotando em diferentes figuras.....	3
Funções por partes.....	3
Outros comandos para plotar uma função.....	4
Plot de funções de tempo discreto.....	4
Gráfico em coordenadas polares.....	4
Gráfico tridimensional.....	5

Plot com o MATLAB

Plotando em duas dimensões

a) Plot a função $y(x) = x^2$, $-2 \leq x \leq 2$

```
clear all; close all
clc
x0=-2:2
length(x0)
y0=x0.^2;
length(y0)
plot(x0,y0)
```

Observação 1: Note que a representação gráfica correspondeu a uma quantidade de cinco pares ordenados (x,y), que resultou em um gráfico distorcido da representação desejada para a função. Neste tipo de caso deve ser aumentar o número de pontos da variável independente para obter uma melhor representação gráfica.

b) Aumente o número de ponto da variável independente x , considerando passo de incremento de 0.1 e refaça o item anterior

```
x=-2:0.1:2
y=x.^2
length(y)
plot(x,y)
plot(y)
y=x.^2;
length(y)
```

Observação 2: Há algumas opções disponíveis no canto superior direito das figuras. As opções permitem exportar, visualizar valor numérico, dar zoom e editar em janela gráfica externa.

Comando linspace e logspace

```
x=0:0.25:1
x=linspace(0,1,5)
%Criar um vetor com cinco pontos espaçados logaritmicamente entre 10^0 até
%10^1
logspace(0,1,5)
```

Plotando várias funções em uma figura

c) Plot as seguintes funções em uma mesma figura. $y(x) = x^2 \cos(x)$, $g(x) = x \cos(x)$ e $f(x) = 2^x \sin(x)$, $0 \leq x \leq 2\pi$, considerando 100 pontos.

```
x=linspace(0,2*pi,100)
y=(x.^2).*cos(x)
g=x.*cos(x)
f=(2.^x).*sin(x)
plot(x,y,x,g,x,f)
```

Observação 3: Note que a figura foi criada com cores predefinidas e linha do tipo sólido. É possível criar gráficos usando cores, símbolos e tipos de linhas personalizados por meio da inserção de outros argumentos na função `plot`, vide opções por meio do comando `help plot`.

Configurando opções do plot

d) Plot as funções $y_1(x) = \sin(x)$ e $y_2(x) = \cos(x)$, $-1 \leq x \leq 1$ com 15 elementos. Para a primeira função utilize cor preta, linha pontilhada e marcadores do tipo pentagrama. Enquanto que para segunda função utilize cor vermelha, linha tracejada e marcadores do tipo círculo.

```
x=linspace(-1,1,15)
y1=asin(x)
y2=acos(x)
plot(x,y1,'k:p',x,y2,'r--o')
```

Outra forma de sintaxe para o comando `plot` que especifica cor e tipo de linha é da seguinte forma

```
plot(x,y1,'Color',[1,0,1],'LineWidth',2,'Marker','diamond','Linestyle','-.')
```

Comando hold

Ao repetir a execução do comando `plot` para uma figura prévia, a representação gráfica obtida anteriormente é descartada. Para manter o gráfico anterior e acrescentar um novo gráfico por meio da repetição do comando `plot` deve-se utilizar o comando `hold on`. Veja o exemplo

```
x=linspace(0,2*pi,150)
plot(x,cos(x))
hold on
plot(x,sin(x))
```

Formatando uma figura

Há alguns comandos úteis para formatação de figuras, tais como:

grid-title-xlabel-ylabel-legend-text-gtext-axis-xlim-ylim

```
x=linspace(0,2*pi,150)
plot(x,cos(x), 'r*',x,sin(x), 'k')
legend('cos(x)', 'sen(x)')
grid
xlabel('Eixo x')
ylabel('Eixo y')
title('Gráfico de cos(x) e sen(x)')
text(3,0.3, 'sen(x)')
text(2,0.3, 'cos(x)')
axis([2,4,-1,0])
```

Plotando em diferentes figuras

Para criar diferentes figuras pode-se especificar a partir do comando `figure(1) ... figure(k)` e utilizar para cada figura o comando `plot`.

Mas também é possível criar múltiplas figuras dentro de uma única figura por meio do comando `subplot(m,n,p)`. Com este comando cria-se dentro de uma mesma janela de figura que estará dividida em $m \times n$ pequenas subfiguras.

e) Utilizando o comando `subplot` plot as seguintes funções $f(x) = \cos(x)$, $g(x) = e^{-\frac{x}{2\pi}}\cos(x)$, $f(x) = e^{\frac{x}{2\pi}}\cos(x)$ em uma única figura contendo 3 subfiguras disposta em um arranjo 3x1

```
figure
x=0:0.1:4*pi
```

```
x = 1x126
      0      0.1000      0.2000      0.3000      0.4000      0.5000      0.6000      0.7000 ...
```

```
subplot(3,1,1)
plot(x,cos(x))
grid on
subplot(3,1,2)
plot(x, exp(-x/(2*pi)).*cos(x))
grid on
subplot(3,1,3)
plot(x, exp(x/(2*pi)).*cos(x))
grid on
```

Funções por partes

f) Obtenha a representação gráfica de uma função definida por partes, tal que

$$f(t) = \begin{cases} 1, & -2 \leq t \leq 2 \\ 0, & 2 < t < 5 \\ t \sin(4\pi t) & 5 \leq t \leq 8 \end{cases}$$

```

t1=-2:0.01:2
t2=2.01:0.01:4.99
t3=5:0.01:8
f1=ones(size(t1))
f2=zeros(size(t2))
f3=t3.*sin(4*pi*t3)
t=[t1 t2 t3]
f=[f1 f2 f3]
figure
plot(t,f)

```

Outros comandos para plotar uma função

Além do comando `plot` há outros comandos que podem ser empregados para obter uma representação gráfica de uma função, tais como:

`loglog-semilogx-semilogy-area-fplot-ezplot`

f) Considere a função $y(x) = 2x + 30$, $1 \leq x \leq 1000$. Obtenha diferentes representações gráficas utilizando outros comandos além do `plot`.

```

x=1:1000
y=2*x+30
figure
plot(x,y)
figure
loglog(x,y)
figure
semilogy(x,y)
figure
semilogx(x,y)
figure
area(x,y)
figure
fplot(@(x)2.*x+30,[1,1000])
%fplot('2*x+30',[1,1000])
%ezplot('2*x+30',[1,1000])

```

Plot de funções de tempo discreto

g) Considere a função de tempo discreto $f(k) = k^2$, $k \in \mathbb{Z} : -3 \leq k \leq 3$.

```

k=-3:3;
f=k.^2;
stem (k,f)
stem(k,f,"r*")

```

Gráfico em coordenadas polares

```

theta = linspace(0,4*pi,50)

```

```

theta = 1x50

```

0	0.2565	0.5129	0.7694	1.0258	1.2823	1.5387	1.7952 ...
---	--------	--------	--------	--------	--------	--------	------------

```
c=theta
```

```
c = 1x50
```

0	0.2565	0.5129	0.7694	1.0258	1.2823	1.5387	1.7952 ...
---	--------	--------	--------	--------	--------	--------	------------

```
polarplot(theta, c, '--*')
```

Gráfico tridimensional

```
x=0:.1:100
```

```
x = 1x1001
```

0	0.1000	0.2000	0.3000	0.4000	0.5000	0.6000	0.7000 ...
---	--------	--------	--------	--------	--------	--------	------------

```
y= cos(x)
```

```
y = 1x1001
```

1.0000	0.9950	0.9801	0.9553	0.9211	0.8776	0.8253	0.7648 ...
--------	--------	--------	--------	--------	--------	--------	------------

```
z= sin(x)
```

```
z = 1x1001
```

0	0.0998	0.1987	0.2955	0.3894	0.4794	0.5646	0.6442 ...
---	--------	--------	--------	--------	--------	--------	------------

```
plot3(x,y,z)
xlabel('x')
ylabel('y=cos(x)')
zlabel('z=sin(x)')
```

