

João Pedro Dias Nunes 10705846

Modelagem e Introdução ao Scilab/Matlab
PME 3380 - Modelagem de Sistemas Dinâmicos - Lista 1

Brasil

2020

Sumário

1	COMANDOS DIRETOS	3
1.1	Números	3
1.1.1	Definição de uma constante	3
1.1.2	Escrevendo números complexos	3
1.1.3	Espressão Booleana	3
1.2	Matrizes e vetores	4
1.2.1	Vetor constante	4
1.2.2	Matriz constante	4
1.2.3	Definição de matrizes com operações de objetos definidos anteriormente	5
1.2.4	Uso do ponto e vírgula	5
1.2.5	Elemento zero	5
1.2.6	Matriz de zeros com 2 linhas e 3 colunas	5
1.2.7	Matriz de zeros com as mesmas dimensões da matriz A	5
1.2.8	Matriz de uns com 2 linhas e 3 colunas	6
1.2.9	Matriz diagonal com os elementos da diagonal principal indo de 1 a 5	6
1.2.10	Extraindo elementos da diagonal principal	6
1.2.11	Formando uma matriz diagonal com os elementos da diagonal principal de uma matriz	6
1.3	Operações	7
1.3.1	Matriz identidade	7
1.3.2	Somar matrizes	7
1.3.3	Somar 1 a todos elementos de uma matriz	7
1.3.4	Multiplicação de matrizes	7
1.3.5	Multiplicação elemento a elemento	8
1.3.6	Extração da linha 2	8
1.3.7	Extração da coluna 3	8
1.3.8	Extração da última linha	8
1.3.9	Traço de uma matriz	8
1.3.10	Rank (característica) de uma matriz	8
1.3.11	Matriz transposta	9
1.3.12	Inversa da matriz	9
1.3.13	Determinante de uma matriz	9
1.4	Polinômios	9
1.4.1	Polinômio em x com raízes em 0 e -1	9
1.4.2	Polinômio com coeficientes 1, 2 e 1	10

1.5	Funções racionais	10
1.5.1	Função racional	10
1.5.2	Extração dos coeficientes	10
1.5.3	Cálculo de raízes	10
1.5.4	Autovalores e autovetores	11
1.6	Funções	11
1.6.1	Definindo uma função	11
1.6.2	Plotando o resultado	11
1.6.3	Criando uma nova janela gráfica	12
1.6.4	Plotando o resultado com asteriscos	12
1.6.5	Criando uma nova janela gráfica	12
1.6.6	Aumentando o tamanho dos asteriscos	13
2	UTILIZAÇÃO DE MACROS	14
2.1	Utilizando macros para funções	14
2.2	Várias funções	14

1 Comandos diretos

Para todos os exercícios dessa lista, o programa utilizado foi o Matlab.

1.1 Números

1.1.1 Definição de uma constante

```
1 >> a=1;
2 >> a
3
4 a =
5
6     1
```

1.1.2 Escrevendo números complexos

```
1 >> a=2+i
2
3 a =
4
5     2.0000 + 1.0000 i
6
7 >> b=-5-3i;
8 >> b
9
10 b =
11
12    -5.0000 - 3.0000 i
```

1.1.3 Expressão Booleana

Verificação se a é igual a 1:

```
1 >> a==1
2
3 ans =
4
5     logical
6
7     0
```

O resultado foi 0 (falso), pois a variável a havia sido definida como $2 + i$. Ao testarmos $a == 2 + 1$:

```
1 >> a==2+i
2
3 ans =
4
5     logical
6
7     1
```

o resultado é 1 (verdadeiro).

1.2 Matrizes e vetores

1.2.1 Vetor constante

```
1 >> v=[1 2 3 4 5]
2
3 v =
4
5     1     2     3     4     5
6
7 >> v=1:5
8
9 v =
10
11     1     2     3     4     5
```

1.2.2 Matriz constante

```
1 >> A=[2 2 3
2 0 0 7
3 5 9 -1]
4
5 A =
6
7     2     2     3
8     0     0     7
9     5     9    -1
10
11 >> A=[2 2 3;0 0 7;5 9 -1];
12 >> A
13
14 A =
15
16     2     2     3
17     0     0     7
18     5     9    -1
```

1.2.3 Definição de matrizes com operações de objetos definidos anteriormente

```

1 >> a=1;b=2;
2 >> A = [a+b pi 3
3         b^2 0 atan(a)
4         5 sin(b) -1]
5
6 A =
7
8     3.0000     3.1416     3.0000
9     4.0000         0     0.7854
10    5.0000     0.9093    -1.0000

```

1.2.4 Uso do ponto e vírgula

Quando usamos ponto e vírgula no final, há uma definição apenas, e, portando, o resultado não aparece na tela. Quando não se usa, o resultado é mostrado:

```

1 >> V = [1 2 3]
2
3 V =
4
5     1     2     3
6
7 >> V = [1,2,3];

```

1.2.5 Elemento zero

```

1 >> B=zeros()
2
3 B =
4
5     0

```

1.2.6 Matriz de zeros com 2 linhas e 3 colunas

```

1 >> B=zeros(2,3)
2
3 B =
4
5     0     0     0
6     0     0     0

```

1.2.7 Matriz de zeros com as mesmas dimensões da matriz A

```

1 >> A=[2 2 3;0 0 7;5 9 -1];
2 B = zeros (size(A));
3 >> B

```

```
4
5 B =
6
7     0     0     0
8     0     0     0
9     0     0     0
```

1.2.8 Matriz de uns com 2 linhas e 3 colunas

```
1 >> C=ones(2,3)
2
3 C =
4
5     1     1     1
6     1     1     1
```

1.2.9 Matriz diagonal com os elementos da diagonal principal indo de 1 a 5

```
1 >> D=diag(1:5)
2
3 D =
4
5     1     0     0     0     0
6     0     2     0     0     0
7     0     0     3     0     0
8     0     0     0     4     0
9     0     0     0     0     5
```

1.2.10 Extraindo elementos da diagonal principal

```
1 >> A=[1 2 3
2       4 5 6
3       7 8 9];
4 >> B=diag(A)
5
6 B =
7
8     1
9     5
10    9
```

1.2.11 Formando uma matriz diagonal com os elementos da diagonal principal de uma matriz

```
1 >> C=diag(diag(A))
2
3 C =
4
```

```
5      1      0      0
6      0      5      0
7      0      0      9
```

1.3 Operações

1.3.1 Matriz identidade

```
1 >> A=diag(ones(1,3))
2
3 A =
4
5      1      0      0
6      0      1      0
7      0      0      1
```

1.3.2 Somar matrizes

```
1 >> B=A+A
2
3 B =
4
5      2      0      0
6      0      2      0
7      0      0      2
```

1.3.3 Somar 1 a todos elementos de uma matriz

```
1 >> C=B+1
2
3 C =
4
5      3      1      1
6      1      3      1
7      1      1      3
```

1.3.4 Multiplicação de matrizes

```
1 >> A=[1 2 3;4 5 6;7 8 9];
2 >> C=[1 2 0;0 0 1;0 2 3];
3 >> D=A*C
4
5 D =
6
7      1      8     11
8      4     20     23
9      7     32     35
```


1.3.5 Multiplicação elemento a elemento

```
1 >> A=[1 0 0;0 2 3;5 0 4];
2 >> B=[2 0 0;0 2 2;0 0 3];
3 >> C=A.*B
4
5 C =
6
7     2     0     0
8     0     4     6
9     0     0    12
```

1.3.6 Extração da linha 2

```
1 >> a=C(2,:)
2
3 a =
4
5     0     4     6
```

1.3.7 Extração da coluna 3

```
1 >> b = C(:,3)
2
3 b =
4
5     0
6     6
7    12
```

1.3.8 Extração da última linha

```
1 >> b = C(end,:)
2 b =
3
4     0     0    12
```

1.3.9 Traço de uma matriz

```
1 >> A=[1 2 3; 4 5 6;7 8 9];
2 >> t=trace(A)
3
4 t =
5
6    15
```

1.3.10 Rank (característica) de uma matriz

```
1 >> r=rank(A)
2
3 r =
4
5     2
```

1.3.11 Matriz transposta

```
1 >> B=A'
2
3 B =
4
5     1     4     7
6     2     5     8
7     3     6     9
```

1.3.12 Inversa da matriz

```
1 >> A=[0 1;-2 -3];
2 >> B=inv(A)
3
4 B =
5
6    -1.5000    -0.5000
7     1.0000         0
8
9 >> A*B
10
11 ans =
12
13     1     0
14     0     1
```

1.3.13 Determinante de uma matriz

```
1 >> d=det(A)
2
3 d =
4
5     2
```

1.4 Polinômios

1.4.1 Polinômio em x com raízes em 0 e -1

```
1 >> v=[0 -1];
2 >> p1=poly(v)
3
```

```
4 p1 =  
5  
6      1      1      0
```

O Matlab representa um polinômio a partir do vetor de seus coeficientes, dessa forma, o vetor acima representa o polinômio $p_1(x) = 1x^2 + 1x + 0$.

1.4.2 Polinômio com coeficientes 1, 2 e 1

```
1  
2 >> p = [1 2 1]  
3  
4 p =  
5  
6      1      2      1
```

1.5 Funções racionais

1.5.1 Função racional

```
1 p1 =  
2  
3      1      1      0  
4  
5 >> p2 = [5 2 1]  
6  
7 p2 =  
8  
9      5      2      1  
10  
11 >> f=p1/p2  
12  
13 f =  
14  
15      0.2333
```

1.5.2 Extração dos coeficientes

```
1 >> p2  
2  
3 p2 =  
4  
5      5      2      1
```

1.5.3 Cálculo de raízes

```
1 >> p=roots(p1)  
2
```

```
3 p =  
4  
5     0  
6    -1
```

1.5.4 Autovalores e autovetores

d - matriz diagonal cujos elementos são os autovalores

v - matriz cujas colunas são os autovetores

```
1 >> [v,d] = eig(A)  
2  
3 v =  
4  
5     0.7071    -0.4472  
6    -0.7071     0.8944  
7  
8  
9 d =  
10  
11     -1     0  
12     0    -2
```

1.6 Funções

1.6.1 Definindo uma função

```
1 >> syms y(x)  
2 y(x) = piecewise(x<0,-x^2,x>=0, sin(x));  
3 >> y(0.5*pi)  
4  
5 ans =  
6  
7 1
```

1.6.2 Plotando o resultado

```
1 >> x = [-4:0.5:4];  
2 >> h = y(x);  
3 plot(x,h)
```

Cujo resultado é mostrado na figura 1:

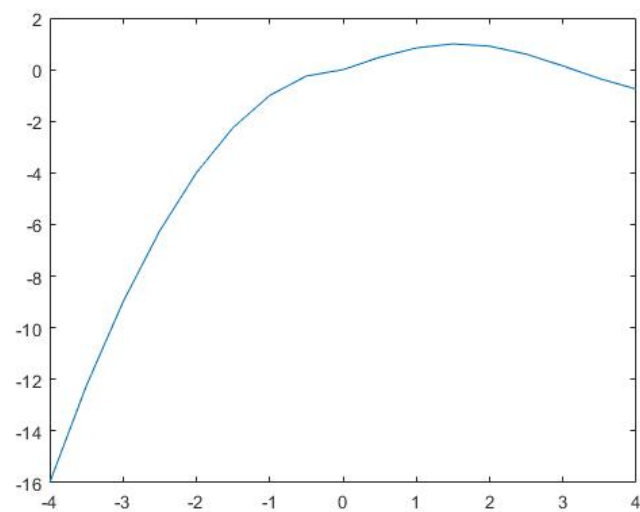


Figura 1 – Figura resultado do código acima

1.6.3 Criando uma nova janela gráfica

```
1 f1=figure;
```

1.6.4 Plotando o resultado com asteriscos

```
1 >> plot(x,h,'*')
```

A expressão acima remete ao seguinte resultado:

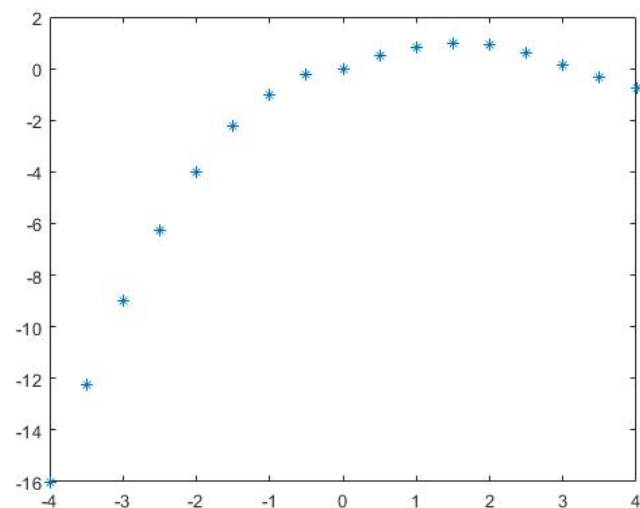


Figura 2 – Figura resultado do código acima

1.6.5 Criando uma nova janela gráfica

```
1 >> f2=figure;
```

1.6.6 Aumentando o tamanho dos asteriscos

```
1 >> plot(x,h, '*', 'MarkerSize',4)
```

O código acima retorna o seguinte resultado:

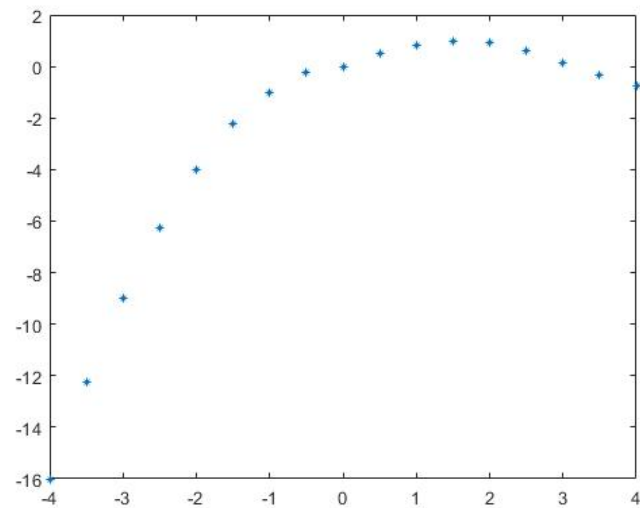


Figura 3 – Figura resultado do código acima

2 Utilização de macros

2.1 Utilizando macros para funções

Definimos a função em um arquivo separado

```
1 function y=teste(x)
2     y=x+x^2+sin(x*2*pi);
3 end
```

e, testando:

```
1 >> teste(0.5*pi)
2
3 ans =
4
5     3.6079
```

2.2 Várias funções

Pode-se criar uma macro, como o código abaixo:

```
1 test0 = @(x) x + x.^2+sin(2*x.*pi);
2 test1 = @(x) -x+ x.^2+x.^3;
3 test2 = @(x) x.^0.5;
4
5 x=[-2:0.5:3];
6
7 a=1;
8 b=0;
9 t1=(a==1);
10 t2=(b>0.5);
11
12 if and(t1,t2)
13     y=test0(x);
14 elseif or(t1,t2)
15     y=test1(x);
16 else
17     y=test2(x);
18 end
19
20 plot(x,y,'*')
21
22 f1 = figure
23 plot(x,y,'*','MarkerSize',2)
```

```
24  
25 f2 = figure  
26 plot(x,y, '*', 'MarkerSize',4)  
27  
28 f3 = figure  
29 plot(x,y, '*', 'MarkerSize',5)
```

cujas figuras resultantes são:

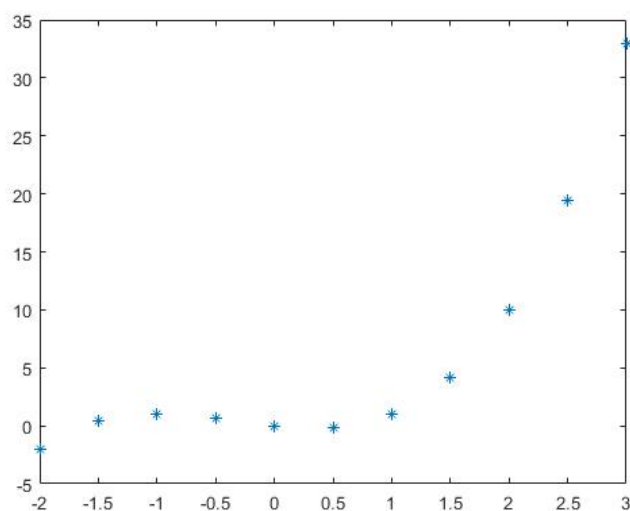


Figura 4 – Primeira figura resultante do código acima

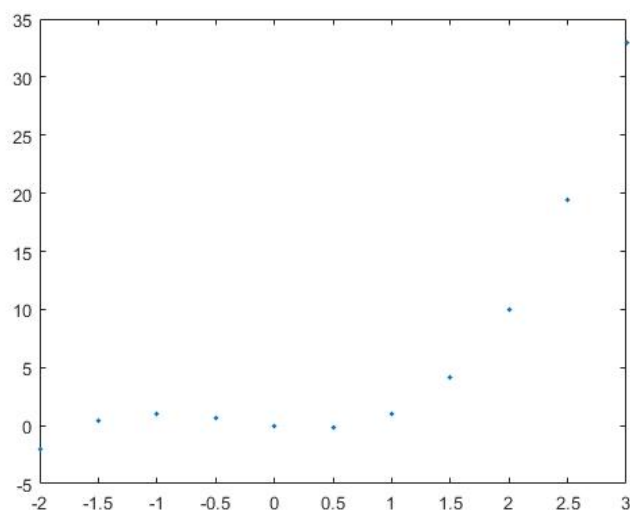


Figura 5 – Segunda figura resultante do código acima

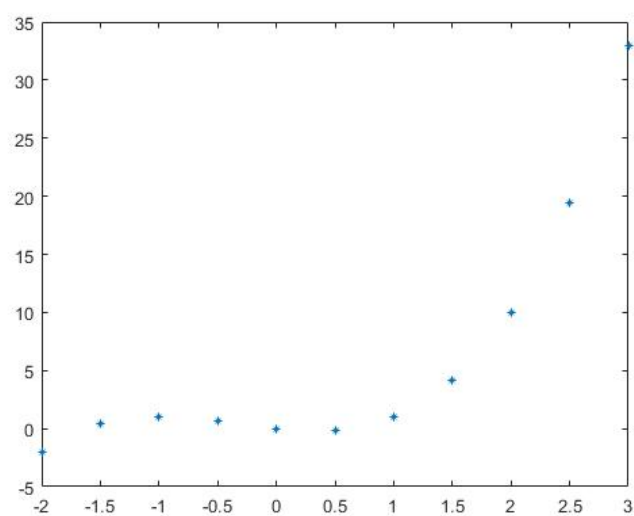


Figura 6 – Terceira figura resultante do código acima

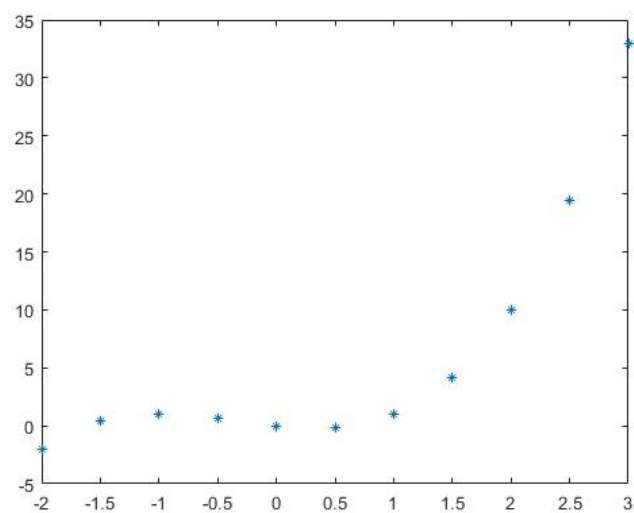


Figura 7 – Quarta figura resultante do código acima