

Escola Politécnica da Universidade de São Paulo

PME 3380 – Modelagem de Sistemas Dinâmicos

## Lista A



Gabriela Vasconcelos Araujo - 10771497

Turma 1  
Prof. Agenor de Toledo Fleury  
São Paulo, 2020

# SUMÁRIO

---

1. Familiarização com Comandos do Scilab .....	2
1.1 Instruções - Código .....	2
1.2 Saídas .....	7
2. Macros .....	8
2.1 Primeiro Teste – teste.sci .....	8
2.1.1 Instruções – Código .....	8
2.1.2 Saída .....	8
2.2 Segundo Teste – teste.sce .....	8
2.2.1 Instruções – Código .....	8
2.2.2 Saídas.....	9

# 1. FAMILIARIZAÇÃO COM COMANDOS DO SCILAB

---

## 1.1 INSTRUÇÕES - CÓDIGO

```
// ~~~~~ Familiarização com Comandos do SciLab ~~~~~

disp("----- Objetos -----");

// DEFINIÇÃO DE UMA CONSTANTE
disp("Definição de uma constante:");
disp("a = 1;");
a = 1;
pause

// ESCRIVENDO NÚMERO COMPLEXOS
disp("Escrevendo números complexos:");
disp("a = 2 + %i;");
disp("b = -5 -3*%i;");
a = 2 + %i;
b = -5 -3*%i;
pause

disp("----- Expressão Booleana -----");

// VERIFICAÇÃO SE A É IGUAL A 1
disp("Verificação se a é igual a 1:");
disp(a == 1); // console retorna "F" (false)
pause

disp("----- Matrizes e Vetores -----");

// VETOR CONSTANTE
disp("Vetor constante:");
disp("v = [1 2 3 4 5] ou v = 1:5");
v = [1 2 3 4 5];
v = 1:5;
pause

// MATRIZ CONSTANTE
disp("Matriz constante:");
disp("A = [2 2 3");
disp("  0 0 7");
disp("  5 9 -1];");
A = [2 2 3
      0 0 7
      5 9 -1];
disp("ou");
disp("A = [2 2 3; 0 0 7; 5 9 -1];");
A = [2 2 3; 0 0 7; 5 9 -1];
pause

// MATRIZES FORMATADAS ATRAVÉS DE OPERAÇÕES
disp("Podemos formar matrizes usando operações com objetos definidos anteriormente:");
disp("a=1;b=2;");
a=1;b=2;
```

```

disp("Observe que se colocarmos ponto e vírgula ao final da expressão, o resultado não é mostrado na tela, o
que pode ser conveniente em algumas situações.");
disp("A = [a+b %pi 3]");
disp(" b^2 0 atan(a)");
disp(" 5 sin(b) -1");
A = [a+b %pi 3
      b^2 0 atan(a)
      5 sin(b) -1];
pause

// MATRIZES E VETORES DE ZEROS
disp("Podemos formar matrizes e vetores de zeros:");
disp("Elemento zero:");
disp("B = zeros();");
B = zeros();
pause
disp("Matriz de zeros com 2 linhas e 3 colunas:");
disp("B = zeros(2,3);");
B = zeros(2,3);
pause
disp("Matriz de zeros com as mesmas dimensões da matriz A:");
disp("A = [2 2 3;0 0 7; 5 9 -1];");
disp("B = zeros(A);");
A = [2 2 3;0 0 7; 5 9 -1];
B = zeros(A);
pause

// FMATRIZES E VETORES DE UNS
disp("Matriz de uns com 2 linhas e 3 colunas:");
disp("C = ones(2,3);");
C = ones(2,3);
pause

// MATRIZES DIAGONAIS
disp("Matrizes diagonais:");
disp("Matriz diagonal com os elementos da diagonal principal indo de 1 a 5:");
disp("D = diag(1:5);");
D = diag(1:5);
pause

disp("Extraindo os elementos da diagonal principal:");
disp("A = [1 2 3]");
disp(" 4 5 6");
disp(" 7 8 9");
disp(" B = diag(A)");
A = [1 2 3
      4 5 6
      7 8 9];
B = diag(A);
pause

disp("Formando uma matriz diagonal com os elementos da diagonal principal de uma matriz:");
disp("C = diag(diag(A))");
C = diag(diag(A));
pause

disp("----- Operações -----");

// MATRIZ IDENTIDADE
disp("Matriz identidade:");

```

```

disp("A = diag(ones(1,3))");
A = diag(ones(1,3));
pause

// SOMA DE MATRIZES
disp("Soma de matrizes:");
disp("B = A + A");
B = A + A;
pause

// SOMA DE CONSTANTE E MATRIZ
disp("Somar 1 a todos os elementos de uma matriz:");
disp("C = B + 1");
C = B + 1;
pause

// MULTIPLICAÇÃO DE MATRIZES
disp("Multiplicação de matrizes:");
disp("A = [1 2 3;4 5 6;7 8 9]");
disp("C = [1 2 0;0 0 1;0 2 3]");
disp("D = A * C");
A = [1 2 3;4 5 6;7 8 9];
C = [1 2 0;0 0 1;0 2 3];
D = A*C;
pause

disp("Multiplicação elemento a elemento:");
disp("A = [1 0 0;0 2 3;5 0 4]");
disp("B = [2 0 0;0 2 2;0 0 3]");
disp("C = A. * B");
A = [1 0 0;0 2 3;5 0 4];
B = [2 0 0;0 2 2;0 0 3];
C = A. * B;
pause

// EXTRAÇÃO DE ELEMENTOS DA MATRIZ
disp("Extração da linha 2:");
disp("a = C(2,:)");
a = C(2,:);
disp("Extração da coluna 3:");
disp("b = C(:,3)");
b = C(:,3);
disp("Extração da última linha:");
disp("c = C($,:)");
c = C($,:);
pause

// TRAÇO DE MATRIZ ("TRACE")
disp("Traço de uma matriz:");
disp("A = [1 2 3; 4 5 6; 7 8 9]");
disp("t = trace(A)");
A = [1 2 3; 4 5 6; 7 8 9];
t = trace(A);
pause

// CARACTERÍSTICA DE UMA MATRIZ ("RANK")
disp("Rank (característica) de uma matriz:");
disp("r = rank(A)");
r = rank(A);
pause

```

```

// MATRIZ TRANSPOSTA
disp("Matriz transposta:");
disp("B = A'");
B = A';
pause

// MATRIZ INVERSA
disp("Inversa de matriz:");
disp("A = [0 1;-2 -3]");
disp("B = inv(A)");
disp("C = A * B");
A = [0 1;-2 -3];
B = inv(A);
C = A * B;
pause

// DETERMINANTE DE UMA MATRIZ
disp("Determinante de uma matriz:");
disp("d = det(A)");
d = det(A);
pause

disp("----- Polinômios -----");

// POLINÔMIOS
disp("Polinômio em x com raízes em 0 e -1:");
disp("v = [0 -1]");
disp("p1 = poly(v,'x')");
v = [0 -1];
p1 = poly(v,'x');
pause

disp("Polinômio em z com coeficientes 1 e 2 e 1:");
disp("p2 = poly([1 2 1], 'z', 'coeff')");
p2 = poly([1 2 1], 'z', 'coeff');

// FUNÇÃO RACIONAL
disp("Funções racionais:");
disp("p1: numerador:");
disp("p1 = poly(v,'s')");
p1 = poly(v,'s');
disp("p2: denominador:");
disp("p2 = poly([5 2 1], 's', 'coeff')");
p2 = poly([5 2 1], 's', 'coeff');
disp("Função racional:");
disp("f = p1/p2");
f = p1/p2;
pause

// EXTRAÇÃO DE COEFICIENTES
disp("Extração dos coeficientes:");
disp("a = coeff(p2)");
a = coeff(p2);
pause

// CÁLCULO DE RAÍZES
disp("Cálculo de raízes:");
disp("p = roots(p1)");
p=roots(p1);

```

pause

*// AUTOVETORES E AUTOVALORES*

```
disp("Autovalores e autovetores:");
disp("d -> matriz diagonal cujos elementos são os autovalores.");
disp("v -? matriz cujas colunas são os autovetores.");
disp("[v,d] = spec(A)");
[v,d]=spec(A)
pause
```

```
disp("----- Funções -----");
```

*// DEFINIÇÃO DE FUNÇÃO*

```
disp("Definição de uma função:");
disp("deff('[y]=teste(x)', 'if x<0 then y=-(x^2), else y=sin(x), end'");
deff('[y]=teste(x)', 'if x<0 then y=-(x^2), else y=sin(x), end');
disp("Uma vez definida a função, podemos calcular seu valor no ponto x=pi/2:");
disp("y=teste(0.5*pi)");
y=teste(0.5*pi);
pause
```

```
disp("No caso de x ser um vetor, a sintaxe seria:");
disp("deff('[y]=h(x)', 'n=length(x); for i=1:n, if x(i)<0 then y(i)=2, else y(i)=1+(x(i)-1)^2, end, end'");
deff('[y]=h(x)', 'n=length(x); for i=1:n, if x(i)<0 then y(i)=2, else y(i)=1+(x(i)-1)^2, end, end');
pause
```

```
disp("----- Plotagem -----");
```

*// PLOTANDO A FUNÇÃO*

```
disp("Plotar a função entre -4 e 4:");
disp("Criando um vetor com os valores variando de -4 a 4, com passo de 0.5:");
disp("x=-4:0.5:4;");
x=-4:0.5:4;
disp("Calculando a função:");
disp("y=h(x);");
y=h(x);
disp("Plotando o resultado:");
disp("plot2d(x,y)");
plot2d(x,y);
pause
```

*// PARÂMETROS DE PLOTAGEM*

```
disp("Criando uma nova janela grafica:");
disp("set('current_figure', 1)");
disp("Plotando o resultado com asteriscos:");
disp("plot2d(x,y,-3)");
set("current_figure", 1);
plot2d(x,y,-3);
pause
disp("Criando uma nova janela grafica:");
disp("set('current_figure', 2)");
disp("Aumentando o tamanho dos asteriscos:");
disp("xset('mark size', 4)");
disp("plot2d(x,y,-3)");
set("current_figure", 2);
xset("mark size", 4);
plot2d(x,y,-3);
```

## 1.2 SAÍDAS

Figura 1: Teste de Plotagem – Gráfico de Linha

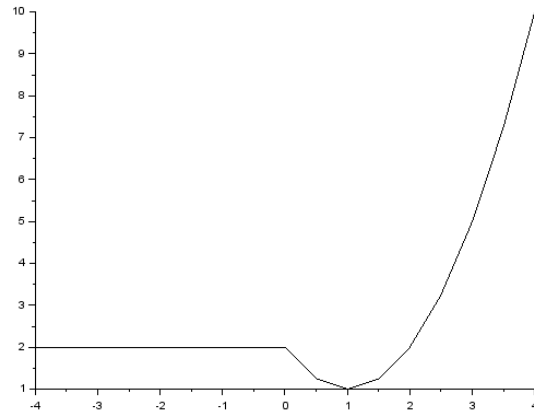


Figura 2: Teste de Plotagem – Dispersão 1

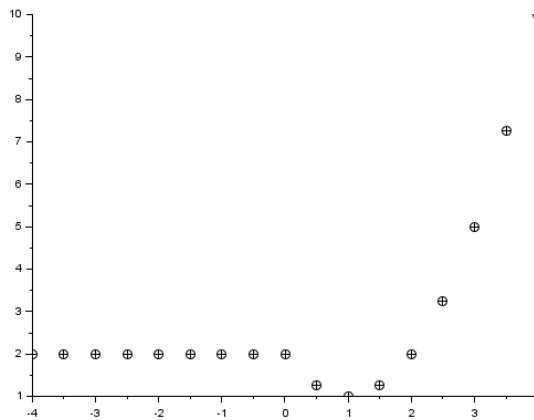
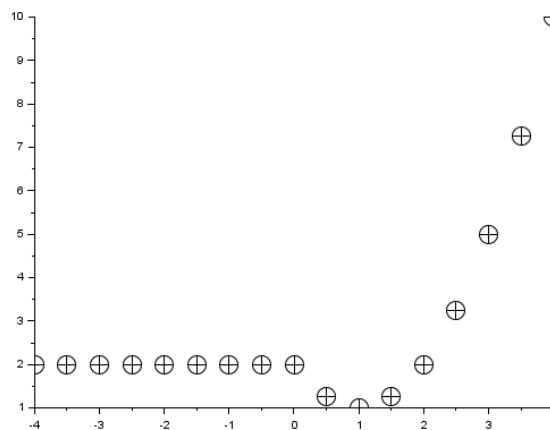


Figura 3: Teste de Plotagem – Dispersão 2





## 2. MACROS

---

Como solicitado, os comandos apresentados no enunciado foram repetidos e executados no SciLab.

### 2.1 PRIMEIRO TESTE – TESTE.SCI

#### 2.1.1 Instruções – Código

```
function [y] = teste(x)
    y = x + x^2 + sin(x*2*%pi);
endfunction
```

#### 2.1.2 Saída

Executando o arquivo com o valor  $\pi/2$ , obtém-se:

```
Execution done.

--> teste(0.5*%pi)
ans =

    3.6078962
```

### 2.2 SEGUNDO TESTE – TESTE.SCE

#### 2.2.1 Instruções – Código

```
def '([y] = test0(x)', 'y = x + x^2 + sin(x*2*%pi)')
def '([y] = test1(x)', 'y = -x + x^2 + x^3')
def '([y] = test2(x)', 'y = sqrt(x)')
```

```
x = -2:0.5:3;
```

```
a = 1;
```

```
b = 0;
```

```
t1 = (a == 1);
```

```
t2 = (b > 0.5);
```

```
if and ([t1 t2]) then
    y = test0(x);
elseif or ([t1 t2]) then
    y = test1(x);
else
    y = test2(x);
end
```

```
plot2d(x,y,-3)
```

```
set("current_figure",1)  
xset('mark size',2)  
plot2d(x,y,-3)
```

```
set("current_figure",2)  
xset('mark size',4)  
plot2d(x,y,-3)
```

```
set("current_figure",3)  
xset('mark size',5)  
plot2d(x,y,-3)
```

## 2.2.2 Saídas

A partir das definições de t1 e t2, nota-se que o primeiro é *True* enquanto o segundo é *False*. Aplicando as estruturas condicionais presentes no código nas expressões *booleanas* t1 e t2, chega-se que y se associa ao test1. Assim, obtém-se os seguintes gráficos:

Figura 4: Janela Gráfica 0 – Dispersão

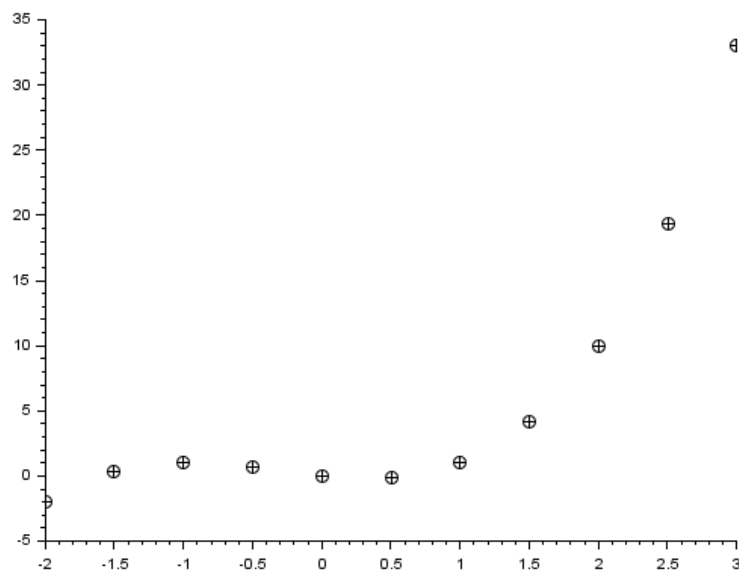


Figura 5: Janela Gráfica 1 – Dispersão, Tamanho 2

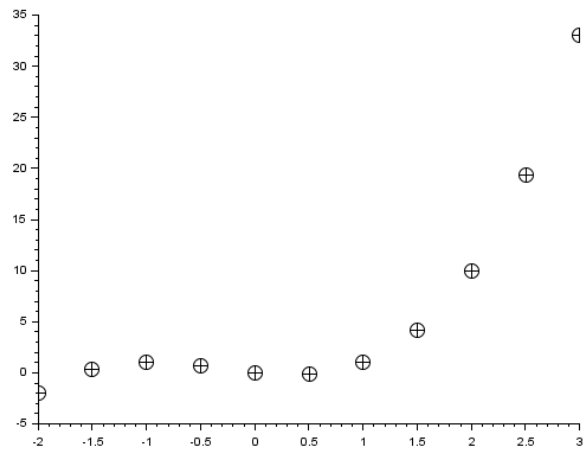


Figura 6: Janela Gráfica 2 – Dispersão, Tamanho 4

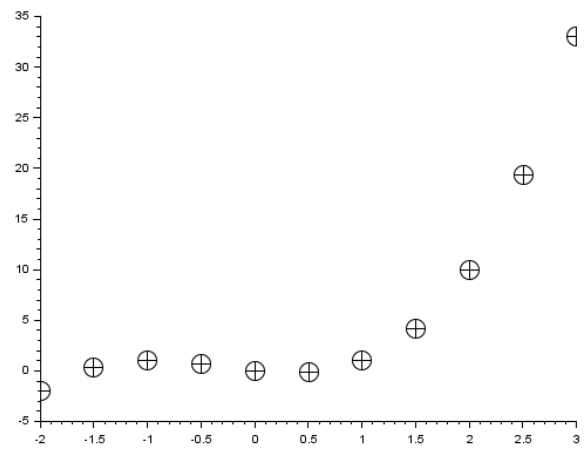


Figura 7: Janela Gráfica 3 – Dispersão, Tamanho 5

