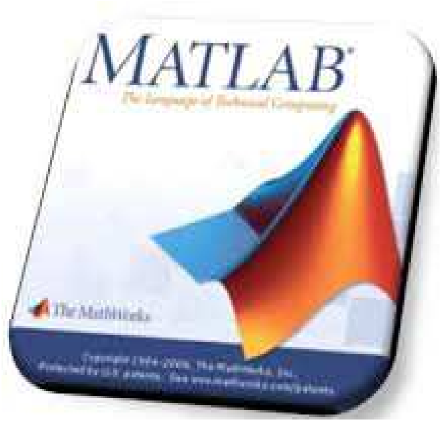
LABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULON
UMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABOR
ATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRI
COCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRI
ODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOM
MATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁ
LCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLA
BLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULO
NUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABO
RATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉR
ICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓR
IODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCO
MMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODEC
ÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATL
ABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCUL
ONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLAB
ORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉ
RICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓ
RIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOC
OMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODE
CÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMAT
LABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIO
DECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMM
ATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCU



LABORATÓRIO DE CÁLCULO NUMÉRICO COM MATLAB

Leila Zardo Puga

LABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULON
UMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABOR
ATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRI
COCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRI
ODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOM
MATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁ
LCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLA
BLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULO
NUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABO
RATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉR
ICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓR
IODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCO
MMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODEC
ÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATL
ABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCUL
ONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMALABORAT
ÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICO
COMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIOD
ECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMA
TLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLC
ULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABL
ABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONU
MÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORA
TÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRIC
OCOMMATLABLABORATÓRIODECÁLCULONUMEECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCLONUMÉR
ICOCOMMATLABLABORATÓRIODECÁOMMATLABLABORATÓRIODECÁLCULONUMÉRICOCOMMATLABLABORATÓRIODECÁLMNIODEC

Sumário

O presente texto, **Laboratório de Cálculo Numérico com Matlab**, pretende servir de material de apoio as aulas de Laboratório da disciplina Cálculo Numérico, do Curso de Engenharia do Centro Universitário da F.E.I.. Os conceitos teóricos não serão aqui abordados, mas podem ser encontrados no livro Cálculo Numérico, adotado como livro texto da disciplina, de nossa autoria.

Nesse sentido, os *Exercícios Resolvidos* resumem-se somente em apresentar as soluções através de 2 abordagens: uma teórica e a outra prática fazendo uso do *software* Matlab. Nas resoluções com Matlab procuramos explorar diversas funções bibliotecas inerentes ao *software* ou, mesmo ainda, codificando e implementando certas *function* no diretório de usuário.

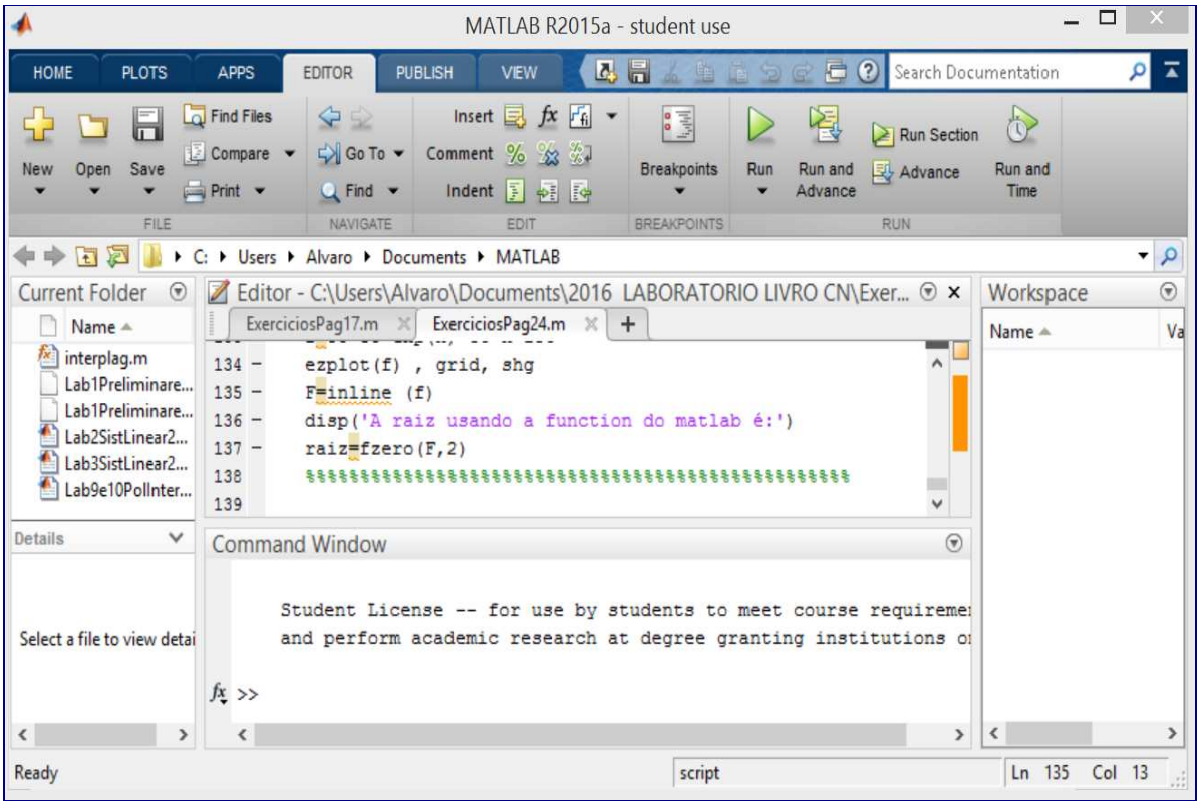
Há uma vasta bibliografia sobre o funcionamento/manipulação do Matlab, algumas das quais indicamos em nosso Curso disponível em http://moodle.fei.edu.br/moodle/ , que não serão aqui tratados. Abordamos somente algumas noções preliminares em Matlab que vamos usar nas aulas de Laboratório de Cálculo Numérico na F.E.I..

Esclarecemos ainda que este texto é uma redação preliminar. Aceitamos sugestões de colegas, professores e demais interessados no tema.

São Paulo, janeiro de 2016.

(I) Preliminares em Matlab

Nas aulas vamos usar a área de trabalho (figura abaixo) do Matlab configurada do seguinte modo:



- Deixar **somente** as 2 janelas centrais: ***Command Window*** e ***Editor***.
- Fechar as demais janelas (para fechar use o ícone no canto superior direito).
- Em aula você verá o que significa cada uma dessas janelas e como reabri-las.
- Podemos digitar em Command Window ou no Editor.
- **Só** Command Window fornece respostas dos comandos que digitamos.
- Ao usar o Editor temos que clicar em (Run) para obter as respostas em Command Window.

Para exemplificar, em *Command Window* , digite os comandos conforme apresentados nos **6 itens** abaixo. Anote suas observações.

1. Calculadora

2+3 ;
2+3
ans , 3*ans
y=2 , exp(y)
y=y+1
sin(pi/2)
ln=log(4)
logbase10=log10(4)
1/0 , 0/0
u=1:3 , norm(u)

2. Formatação

help format
format long
eps
format rat
pi
format compact
single(pi)
digits
format bank
format short

3. Comandos

help diary
clc
mkdir MA2311
disp('Qual é o seu nome?')
% isto é um comentário
save dados
clear
diary on
load dados
edit

4. Matrizes

vetlinha=[1 2 -2 4 1]
vetcoluna=[1; 2; -2; 4; 1]
vetlinha'
matriz=[1 1 3; 3 4 2; 1 5 1]
matriz '
matriz*matriz
matriz ∧2
matriz.. ∧2
inv(matriz)
det(matriz)
size(matriz)

5. Intervalos e Índices

matriz(2,3)
matriz(2,3)=7
matriz(1,:)
matriz(:,2)
matriz(2:end, 1:end-1)
matriz(:)
1:13
1:0.5:13
10:–0.5:1
sum(1:10)
max(matriz)

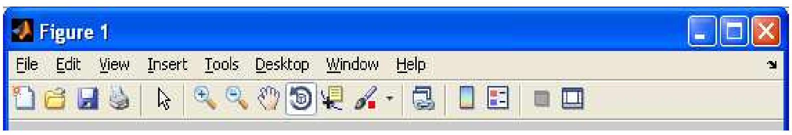
6. Gráficos 2D

t=0:pi/16:2*pi;
figure (1)
plot(cos(t),sin(t))
axis square , grid
title('graf da circunf')
xlabel('eixo x') , ylabel('eixo y')
hold on
plot(cos(3*t),sin(t),'m:')
hold off
figure (2)
subplot(211);
plot(t,cos(t),'ro'); title('cosseno');
subplot(212);
plot(t,sin(t),'g*'); title('seno');

O que é Script?

Abra um arquivo em Matlab (*New Script*), use o icone 

, e digite os comandos dados ao lado. Salve o arquivo com o nome **teste1.** Em *Command Window* é só digitar **teste1** e ver o que acontece! Procure manipular a **Figura 1** com alguns dos ***ícones*** do menu abaixo.




```
n=30;
m=30;
for i=1:m
for j=1:n
a(i,j)=sqrt(i+j);
end
end
b=[a  a'; a.^3  a'.^5];
mesh(b)
```

O que é **Function**?

Abra um arquivo a partir do Matlab (*New Function*)
Digite nesse arquivo os comandos dados ao lado. Salve o arquivo exatamente com o nome **escal**
Em *Command Window* é só digitar, por exemplo, os vetores a=[1 2 3] e b=[2 4 6] e depois escal(a,b).
Qual é o resultado?

```
function z=escal(a,b)
%A função escal retorna o vetor
%z com o resultado do produto
%escalar entre dois vetores a e
%b dados e de mesmo tamanho
if size(a) ~= size(b)
error('Erro: vetores não tem
mesmo tamanho');
end
z=sum(a.*b)
end
```

(II) Primeiros Exercícios

(a) Operações com Matrizes

Dadas as matrizes A , B e C, pede-se calcular:

$$A=\begin{bmatrix}-2 & 0 & 3\\ 1 & 4 & 7\\ -1 & 3 & 2\end{bmatrix},\quad B=\begin{bmatrix}0 & 1 & 4\\ 2 & 2 & 1\\ -1 & 1 & 1\end{bmatrix}\ \text{e}\ C=\begin{bmatrix}-1 & 2 & -5\end{bmatrix}$$

No *Editor* digite os comandos e após clique em  *(Run)* para obter as respostas em *Command Window.*

```
clear   % limpar/esvaziar as variáveis nomeadas
clc   % repaginar o cursor para o topo da janela
A=[-2 0 3 ; 1 4 7 ; -1 3 2]    % matriz A (tipo 3X3) dada no enunciado
B=[0 1 4 ; 2 2 1 ; -1 1 1]     % matriz B dada no enunciado
C=[-1 2 -5]  % matriz C (tipo 1X3) dada no enunciado
P=abs(C)    % valor absoluto dos elementos da matriz C
C*A    % multiplicação usual da matriz C pela matriz A
S=A+B   % S é a soma das matrizes A e B
R=A-C   % R é a subtração das matrizes A e C
M=A*B    % M é a multiplicação usual de matrizes
MP= A.*B    % M é a multiplicação pontual de matrizes
D=det(A)  % determinante da matriz A
Trans=C'  % transposta da matriz C
3*C     % multiplicação de um número real pela matriz C
inv(B)  % matriz inversa de B
P1=S(1,:)  % todos os elementos da primeira linha da matriz S
C2=R(:,2)   % todos os elementos da segunda coluna da matriz R
```

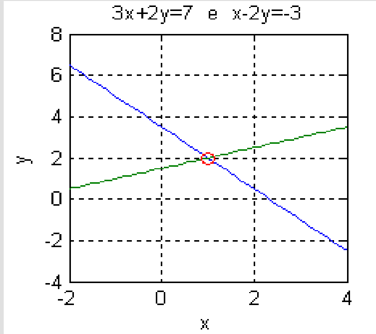
(b) Resolvendo um Sistema linear com a *function plot* do Matlab:

Determinar graficamente (no plano cartesiano) a solução do seguinte sistema linear $\begin{cases}3x+2y=7\\ x-2y=-3\end{cases}$.

Digitando no ***Editor***

```
x=-2:0.1:4 ; %domínio para o gráfico no eixo x
y=-4:0.1:8 ;  %intervalo visual para o eixo y
fx1='(7-3*x)/2' ;  %equação do sistema linear
fx2='(x+3)/2' ;    %equação do sistema linear
y1=subs(fx1,x) ;  %cálculo de f(x1)
y2=subs(fx2,x);   %cálculo de f(x2)
plot(x,y1,x,y2)  %gráfico das 2 retas y1 e y2
grid   %plano quadriculado/grade
% ginput localiza graficamente a intersecção
[X,Y]=ginput(1)   %intersecção de y1 e y2
```

Janela gráfica



Após a digitação, clique em (Run) para obter as respostas em *Command Window.*

(III) Como **enviar** e **salvar** o Exercício de Laboratório de CN

1º Forma de Salvar Arquivo (formato **.txt)**

- Somente quando a digitação é na janela *Command Window*
- Iniciar digitando **diary LabSeuNome** na janela *Command Window*
- No final da resolução dos exercícios você deverá digitar **diary off**
- Salvar o arquivo com a resolução dos exercícios como **LabSeuNome**
- Se houver erros, poderá corrigi-los digitando **Edit LabSeuNome**
- Nesse caso, clique em **Save as** e salve novamente o arquivo
- Enviar esse arquivo para o **Moodle** no link de **sua Turma**

2º Forma de Salvar Arquivo (formato **.pdf** ou **.doc)**

- Somente quando digita-se na janela *Editor*
- No Matlab clicar em **PUBLISH** e em **Edit P. Options** fazer:

(1) Output settings:

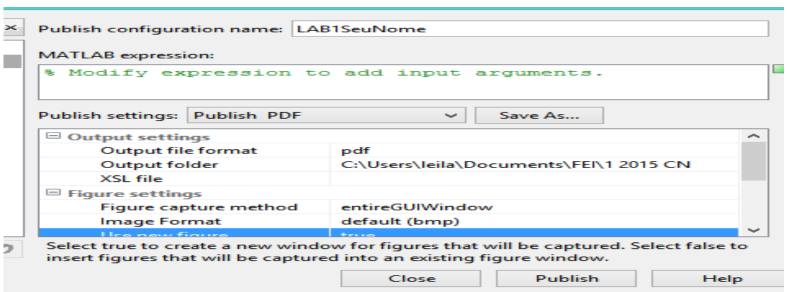
Escolher **PDF** (ou **DOC**) em **output file format**

Escolher seu **diretótio de aluno** em **output folder**

(2) Publish settings:

Escrever **Publish PDF (** ou **Publish DOC) e** Clicar em **SAVE AS**

(3) Clicar em **PUBLISH** (canto inferior direito)



- Enviar esse arquivo para o **Moodle** no link de **sua Turma**

Tabela com algumas funções/comandos principais	
Função/Comando	**O que faz**
ans	exibir a última resposta (de expressão não "nomeada")
%	fazer comentários (sem executar o comando)
;	matlab executa o comando mas não exibe o resultado na tela
disp	exibir no display Command Window
clear	limpar/apagar variável ou função da memória
clc	limpar janela e reposicionar cursor no topo da tela
format long	formato em ponto fixo com 15 dígitos
format bank	formato bancário em ponto fixo 2 dígitos
format rat	formato racional (fração entre números inteiros)
format short	formato em ponto fixo de 4 dígitos
format compact	formato compacto que elimina linha extra
syms X	assumir que a variável X é simbólica
log(x)	logaritmo natural de x (logaritmo na base e)
log10(x)	logaritmo na base 10
exp(x)	função exponencial e^x
cos(t)	função cosseno de t para t em radianos
sin(z)	função seno de z para z em radianos
sqrt(y)	raiz quadrada de y
pi	número pi=3.1416
exp(1)	número e=2.71828
figure	criar uma janela gráfica
t=1:0.5:6	intervalo de extremos inferior 1 e superior 6, com elementos variando de passo 0.5
m*m	multiplicação matricial da matriz m
m.*m	multiplicação pontual dos elementos da matriz m
m.^2	potenciação pontual dos elementos da matriz m
det(m)	determinante da matriz m
rref(C)	diagonalizar (escalonar inferior e superiormente) a matriz C
inv(m)	inverter uma matriz quadrada (com determinante não nulo)

Tabela com algumas funções/comandos principais	
Função/Comando	**O que faz**
digits(5)	cálculos efetuados com 5 dígitos
plot(t , gt)	esboçar/plotar o gráfico da função gt
grid	plano quadriculado (linhas grades)
plot(x, y, '*r')	plotar o gráfico de y em função de x na cor vermelha usando para grafar a curva o símbolo *
ezplot(fx)	fazer gráfico da função fx dada
diff(f)	derivada de f (de ordem 1)
subs(f,x,y)	substituir em f as ocorrências de x por y
fzero(f , v)	raiz/zero da função f nas vizinhanças de v
size(x)	tamanho/número de elementos de x
ones(n)	matriz nXn com elementos iguais a "um"
vpa(C)	variável de precisão aritmética p/ evaluar C
sum(Y)	somar os elementos de Y
abs(z)	valor absoluto/módulo de z
polyfit(X,Y,n)	determina os coeficientes de um polinômio de grau n (potências em ordem decrescente)
polyfit(X,Y,1)	ajuste polinomial por uma reta
interplag	function interpolação de Lagrange
solve	solução simbólica de equação algébrica
length(x)	comprimento do vetor x
prod(D)	produto dos elementos de D
factorial(n)	produto de todos os inteiros de 1 até n
simplify(X)	simplificar os elementos de X
input	solicitar/esperar pela entrada de dados via teclado
fprintf('O valor é: %8.5f\n' , s)	exibir o valor de s na formatação dada 8.5
fprintf('\n')	pular uma linha no display
int(f, a , b)	integral da função f nos extremos de a até b
f=@(x)(.....)	@ é um operador especial para designar função
axis([xmin,xmax,ymin,ymax])	escalas para os eixos x e y num gráfico
eval(X)	evaluar/calcular a expressão string X

1. SISTEMAS LINEARES

1.1 Método da Triangularização de Gauss

Exercícios: Resolver pelo Método da Triangularização de Gauss os seguintes sistemas lineares:

(1*) $\begin{cases} x - y - z = 4 \\ 2x + y + z = 0{,}5 \\ x + y - 2z = -1{,}5 \end{cases}$ **(2)** $\begin{cases} 3x - 2y - 5z = 2 \\ x - y + z = 3 \\ 4x - 3y - 4z = 5 \end{cases}$ **(3)** $\begin{cases} 2x - 2y + z = 1 \\ 2x - y + 5z = -3 \\ 4x - 3y + 6z = 4 \end{cases}$ **(4*)** $\begin{cases} 2x - 2y + 2z - 2t = 4 \\ x - y + z + t = 4 \\ 3x + y - 5z + 3t = 4 \\ 4x - y + 3z - t = 10 \end{cases}$

Resolução (1):

Usando o dispositivo prático (visto em aula de teoria) e escalonando o sistema, temos:

x	y	z	indep.
1	–1	–1	4
2	1	1	0,5
1	1	–2	–1,5
	3	3	–7,5
	2	–1	–5,5
		–9	–1,5

Da última linha do dispositivo, calculamos o valor de z, –9z = –1,5 e, portanto, z=$\frac{1}{6}$. Substituindo o valor de z na penúltima equação, calculamos y; ou seja: 2.y–$\frac{1}{6}$= –5,5. Assim, y=$\frac{-8}{3}$. Na primeira equação calculamos x; x–$\frac{1}{6}+\frac{8}{3}$=4. Daí, x=$\frac{3}{2}$. A solução é $\left(\frac{3}{2}, \frac{-8}{3}, \frac{1}{6}\right)$ e o sistema linear é possível e determinado.

Há muitas formas/procedimentos para resolver um sistema linear no Matlab.

Vamos aqui usar os seguintes (funções bibliotecas/internas do Matlab):

function inv ; *function rref* ; *function * e *function solve*

Para cada um dos Exercícios propostos acima usamos um desses procedimentos.

Para resolver com a *function inv* devemos, inicialmente, **notar que**:

"Um sistema linear pode ser escrito na forma matricial A.X=B".

Sendo A a matriz dos coeficientes das incógnitas, B a matriz dos termos independentes do sistema linear e X a matriz das incógnitas.

Assim, admitindo-se que A possui matriz inversa A^{-1}, temos que: $A^{-1}.A.X=A^{-1}.B \Rightarrow I.X= A^{-1}.B \Rightarrow X= A^{-1}.B$

Portanto, para determinar a solução do sistema, a matriz X, devemos calcular o produto matricial $A^{-1}.B$.

É o 1º Procedimento, a seguir:

1º Procedimento: Função **inv(A)**

Para o Exercício (1), digitando no Matlab em *Command Window*, obtemos:

```
>> % A é a matriz dos coeficientes das incógnitas do sistema linear
>>A=[ 1  -1  -1 ;  2  1  1  ; 1  1  -2 ]
A =
          1.00         -1.00         -1.00
          2.00          1.00          1.00
          1.00          1.00         -2.00
>> % B é a matriz dos termos independentes do sistema linear
>>B=[ 4 ; 0.5 ; -1.5 ]
B =
          4.00
          0.50
         -1.50
>> d=det(A)  % matriz A com determinante d não nulo é invertível
d =
         -9.00
>> S=inv(A)*B    % S é a matriz com a solução do sistema linear
S =
          1.50
         -2.67
          0.17
>> % valores x , y e z das incógnitas do sistema linear
 x=S(1)  , y=S(2)  , z=S(3)
x =
          1.50
y =
         -2.67
z =
          0.17
>> disp('O sistema linear é possível e determinado')
O sistema linear é possível e determinado
```

Resolução (2):

Usando o dispositivo prático e escalonando o sistema, temos:

x	y	z	indep.
3	–2	–5	2
1	–1	1	3
4	–3	–4	5
	–1	8	7
	–1	8	7
		0	0

Da última linha do dispositivo temos: $0.z = 0$ e, portanto, z está indeterminado. Isso significa que $\forall z \in \Re$ é solução do sistema. Esse sistema linear é possível e indeterminado.

Para resolvê-lo no Matlab vamos usar a *function rref*, que procura "diagonalizar uma matriz" ; isto é, decompor uma matriz dada em matriz identidade I e matriz solução. Neste caso, a matriz R não está diagonalizada (diagonalização incompleta) pois, por exemplo, na última linha de R há somente valores nulos.

2º Procedimento: Função **rref(MC)**

Digitando no Matlab, em *Command Window,* obtemos:

```
>> % MC é a matriz completa associada ao sistema linear
>>  MC=[3 -2 -5 2; 1 -1 1 3 ; 4 -3 -4 5]
MC =
     3    -2    -5     2
     1    -1     1     3
     4    -3    -4     5
>> % a função rref  faz a diagonalização da matriz MC
>>  R=rref(MC)
R =
     1     0    -7    -4
     0     1    -8    -7
     0     0     0     0
>>  disp('O sistema linear é possível e indeterminado')
O sistema linear é possível e indeterminado
```

Resolução (3):

Usando o dispositivo prático e escalonando o sistema, temos:

x	y	z	indep.
2	–2	1	1
2	–1	5	–3
4	–3	6	4
	2	8	–8
	2	8	4
		0	24

Da última linha do dispositivo temos: $0.z = 25$ e, portanto, z não existe em $\Re$. Esse sistema linear é impossível.

Resolvendo no Matlab com a *function rref,* temos:

```
>> % matriz completa associada ao sistema linear
>> MC=[2 -2 1 1 ; 2 -1 5 -3 ; 4 -3 6 4]
MC =
     2    -2     1     1
     2    -1     5    -3
     4    -3     6     4
```

```
>> % a função rref  busca fazer a “diagonalização” da matriz MC
>>  R=rref(MC)
R =
         1         0        4.5         0
         0         1        4           0
         0         0        0           1
>> disp('Note que há 3 valores nulos e um não-nulo na última linha de R')
Note que há 3 valores nulos e um valor não-nulo na última linha de R
>>  disp('O sistema linear é impossível')
O sistema linear é impossível
```

Resolução (4):

Usando o dispositivo prático e escalonando o sistema, temos:

x	y	z	t	indep.
2	–2	2	–2	4
1	–1	1	1	4
3	1	–5	3	4
4	–1	3	–1	10
pivô nulo	← 0	0	4	4
	8	–16	12	–4
	6	–2	6	4
	8	–16	12	–4
permutando as equações	6	–2	6	4
	0	0	4	4
		80	–24	56
			4	4

Da última linha temos que 4.t=4 e, portanto, t=1. Na penúltima linha do dispositivo calculamos o valor de z, obtendo: $80z=24t+56 \Rightarrow 80z=80 \Rightarrow z=1$. Da equação $6y-2z+6t=4 \Rightarrow y=0$. Na primeira equação do sistema linear calculamos x: $2x-2y+2z-2t=4 \Rightarrow x=2$. O sistema linear é possível e determinado.

Resolvendo no Matlab com a *function* \ :

3º Procedimento: Função \

A *function* \ é a mesma que a *function inv*, somente mudando a sintaxe de digitação no Matlab. Portanto, para resolver um sistema linear com a *function* \ temos que verificar se a matriz A, dos coeficientes das incógnitas, é invertível. Caso contrário, devemos usar outro procedimento.

Digitando no Matlab, em *Command Window:*

```
>> % A é a matriz dos coeficientes das incógnitas do sistema
>>  A=[ 2 -2 2 -2 ; 1 -1 1 1 ; 3 1 -5 3 ; 4 -1 3 -1 ]
A =
     2    -2     2    -2
     1    -1     1     1
     3     1    -5     3
     4    -1     3    -1
>>  % B é a matriz dos termos independentes do sistema linear
>> B=[ 4 ; 4 ; 4 ; 10 ]
B =
     4
     4
     4
    10
>> d=det(A)    % matriz A com determinante não nulo é invertível
d =
   80.0000
>> S=A\B     % S é a matriz com a solução do sistema linear
S =
    2.0000
    0.0000
    1.0000
    1.0000
>> % valores das incógnitas x , y , z e t do sistema linear
>> x=S(1)   ,   y=S(2)   ,   z=S(3)   ,   t=S(4)
x =
    2
y =
    0
z =
    1
t =
    1
```

4º Procedimento: Função *solve*

Para exemplificar esse procedimento vamos resolver no Matlab o Exercício 1, cuja solução é $\left(\frac{3}{2}, \frac{-8}{3}, \frac{1}{6}\right)$.

```
>>  syms  x y z
 S = solve(x-y-z==4 , 2*x+y+z==0.5 , x+y-2*z==-1.5)
S =
    x: [1x1 sym]
    y: [1x1 sym]
    z: [1x1 sym]
>> %Para ver a solução basta chamar a estrutura S
S = [S.x  S.y  S.z]
S =
[ 3/2, -8/3, 1/6]
```

Outro exemplo, utilizando a *function solve* do Matlab, é o Exercício 2, um sistema linear possível e indeterminado. Neste caso, temos:

```
>>   syms  x y z
 S = solve(3*x-2*y-5*z==2 , x-y+z==3 , 4*x-3*y-4*z==5)
Warning: The solutions are parametrized by the symbols:
z
S =
    x: [1x1 sym]
    y: [1x1 sym]
    z: [1x1 sym]
>>  S = [S.x  S.y  S.z]
S =
[ 7*z - 4, 8*z - 7, z]
```

Algumas Observações:

1. Como vimos, o determinante da matriz A dos coeficientes das incógnitas do Exercício 1 é não nulo. E os determinantes nos Exercícios 2 , 3 e 4? É possível resolvê-los com a *function inv* ?

2. Note a estrutura da matriz R *("diagonalização incompleta")* nos Exercícios 2 e 3. Como é a matriz diagonalizada do Exercício 1?

3. Nos Exercícios classificamos os sistemas, pois resolvemos a mão em aula antes de digitar no Matlab. Mas se não tivéssemos feito isso? Como classificar um sistema usando codificação? Elaborar uma *function* que faça isso.

1.2 Métodos Iterativos de Gauss-Seidel e de Jacobi

Exercícios: Resolver pelo Método Iterativo de Gauss-Seidel os seguintes sistemas lineares:

(5) $\begin{cases} \quad\quad y - 5z = 8 \\ x + 10y \quad\quad = 20 \\ 10x \quad\quad + 3z = 5 \end{cases}$, com precisão inferior a 0,05

(6) $\begin{cases} 5x_1 + x_2 - x_3 = 2 \\ 2x_1 - x_2 + 8x_3 = 4 \\ x_1 - 10x_2 + 4x_3 = 5 \end{cases}$, de modo que $\left|x_i^{j+1} - x_i^j\right| \leq 0{,}02$, para i=1, 2, 3 e j=0, 1, 2, ... r,

Resolução (5):

Inicialmente, como visto na aula de teoria, devemos ordenar as equações de modo que os maiores elementos fiquem na diagonal principal. Então:

$$\begin{cases} 10x \quad + 3z = 5 \\ x + 10y \quad = 20 \\ \quad y - 5z = 8 \end{cases} \Rightarrow \begin{cases} x_{k+1} = \quad\quad -0{,}3z_k + 0{,}5 \\ y_{k+1} = -0{,}1x_{k+1} \quad\quad + 2{,}0 \\ z_{k+1} = \quad +0{,}2y_{k+1} \quad - 1{,}6 \end{cases}$$

valores iniciais ⇓	1ª iteração ⇓		
$x_0 = 0{,}5$	$x_1 = 1{,}0$	$x_2 = 0{,}9$	$x_3 = 0{,}9$
$y_0 = 2{,}0$	$y_1 = 1{,}9$	$y_2 = 1{,}9$	$y_3 = 1{,}9$
$z_0 = -1{,}6$	$z_1 = -1{,}2$	$z_2 = -1{,}2$	$z_3 = -1{,}2$

Note que na 3ª iteração a solução do sistema linear converge para $x = 0{,}9$, $y = 1{,}9$ e $z = -1{,}2$.

Para resolver no Matlab um sistema linear pelo processo iterativo de Gauss-Seidel, precisamos de um programa, uma *function*, que faça isso. Elaboramos a *function gausseidel*, que deverá ser implementada no diretório.

```
function gausseidel(A,VI,prec)
%Profa. Leila Z. Puga
%A função gausseidel determina a solução de um sistema linear SL pelo
%método iterativo de Gauss-Seidel.
%Admite-se que SL seja convergente e que os maiores valores estão na
%diagonal principal da matriz A associada ao SL.
%Para usar essa function devemos digitar os seguintes dados de entrada:
% A é a matriz completa associada ao sistema linear SL
% VI é a matriz com os valores iniciais
% prec é a precisão escolhida
% Após digitar gausseidel(A,VI,prec)
[L,C]=size(A);
n=L; m=C;
k=2;
x=VI;
x(:,2)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>prec
    k=k+1;
    if cont==1
        k=k-1;
        cont=0;
    end
    for i=1:n
        x(i,k)=A(i,m);
        for j=1:i-1
                x(i,k)=x(i,k)-A(i,j)*x(j,k);
        end
          for j=i+1:n
                x(i,k)=x(i,k)-A(i,j)*x(j,k-1);
          end
        x(i,k)=x(i,k)/A(i,i);
    end
end
gausseidel=x;
GaussSeidel= [1:k ; x]
end
```

A resolução do Exercício (5), em *Command Window*, usando essa *function* é a seguinte:

```
>> A=[10 0 3 5; 1 10 0 20; 0 1 -5 8]   % matriz completa associada ao SL
A =
        10.00            0          3.00          5.00
         1.00        10.00             0         20.00
            0         1.00         -5.00          8.00
>> VI=[0.5 ; 2 ; -1.6]  % matriz com os valores iniciais
VI =
         0.50
         2.00
        -1.60
>> prec=0.05  % precisão dada no enunciado
prec =
         0.05
>> gausseidel(A,VI,prec)  % function implementada no diretório
gausseidel =

    0.5000    0.9800    0.8659    0.8652
    2.0000    1.9020    1.9134    1.9135
   -1.6000   -1.2196   -1.2173   -1.2173
```

Resolução (6):

Ordenando o sistema: $\begin{cases} 5x_1 + x_2 - x_3 = 2 \\ 2x_1 - x_2 + 8x_3 = 4 \\ x_1 - 10x_2 + 4x_3 = 5 \end{cases} \Rightarrow \begin{cases} 5x_1 + x_2 - x_3 = 2 \\ x_1 - 10x_2 + 4x_3 = 5 \\ 2x_1 - x_2 + 8x_3 = 4 \end{cases}$

$$\begin{cases} x_1^{k+1} = -0{,}2\,x_2^k + 0{,}2x_3^k + 0{,}4 \\ x_2^{k+1} = 0{,}1x_1^{k+1} + 0{,}4x_3^k - 0{,}5 \\ x_3^{k+1} = -0{,}25\,x_1^{k+1} + 0{,}125x_2^{k+1} + 0{,}5 \end{cases}$$

$x_1^0 = 0{,}4$	$x_1^1 = 0{,}6$	$x_1^2 = 0{,}51$	$x_1^3 = 0{,}53$
$x_2^0 = -0{,}5$	$x_2^1 = -0{,}24$	$x_2^2 = -0{,}32$	$x_2^3 = -0{,}31$
$x_3^0 = 0{,}5$	$x_3^1 = 0{,}32$	$x_3^2 = 0{,}33$	$x_3^3 = 0{,}33$

A condição dada no enunciado está satisfeita, pois $\left|x_1^3 - x_1^2\right| = 0{,}02$; $\left|x_2^3 - x_2^2\right| = 0{,}01$ e $\left|x_3^3 - x_3^2\right| = 0$.

A solução do sistema linear é $x_1=0{,}53$; $x_2=-0{,}31$ e $x_3=0{,}33$.

No Matlab, a resolução usando a *function gausseidel*, é a seguinte:

```
>> A=[5 1 -1 2 ; 1 -10 4 5 ; 2 -1 8 4]   % matriz completa associada ao SL
A =
     5     1    -1     2
     1   -10     4     5
     2    -1     8     4
>> VI=[0.4 ; -0.5 ; 0.5]   % matriz com os valores iniciais
VI =
    0.40
   -0.50
    0.50
>> prec=0.02  % precisão desejada
prec =
    0.0200
>> gausseidel(A,VI,prec)  %function implementada no diretório
ans =
          0.40          0.60          0.51          0.53          0.53
         -0.50         -0.24         -0.32         -0.31         -0.31
          0.50          0.32          0.33          0.33          0.33
```

Exercício (7): Verifique, inicialmente, pelo *Critério de Sassenfeld* que o sistema linear é convergente para **m=–1** e determine a solução de SL

$$\begin{cases} -mx - 0{,}5y + 0{,}1z = -1{,}2 \\ 0{,}2x + y - 0{,}3z = 0{,}5 \\ 0{,}3x - 0{,}1y + mz = 1{,}5 \end{cases}$$

, usando o método iterativo de *Jacobi.*

Resolução:

Pelo Critério de Sassenfeld, se **m=–1** o sistema linear é convergente. Para verificar que converge, calculamos os valores de β (b1 e b2 e b3) e, em *Command Window* , digitamos:

```
>>A=[1 -0.5 0.1 -1.2;0.2 1 -0.3 0.5;0.3 -0.1 -1 1.5]  %matriz completa associada SL
>>b1=abs(A(1,2)/A(1,1))+abs(A(1,3)/A(1,1))   %critério de Sassenfeld
>>b2=abs(A(2,1)/A(2,2))*b1+abs(A(2,3)/A(2,2))   %idem
>>b3=abs(A(3,1)/A(3,3))*b1+abs(A(3,2)/A(3,3))*b2   %idem
```

As respostas obtidas são b1=0.60 , b2 =0.42 e b3 =0.22 e, portanto, SL pode ser resolvido por um método iterativo. Vamos resolver esse sistema pelo método iterativo usando o arquivo *function jacobi* (esse arquivo/função deverá estar implementado no diretório) para determinar a solução.

Nota: O método iterativo de Jacobí é muito similar ao método iterativo de Gauss-Seidel.

A diferença que há entre eles é que os novos valores calculados em cada iteração não são empregados nos cálculos subsequentes das outras variáveis no método de Jacobi. Em geral, isso acarreta em um número maior de iterações. Como no método iterativo de Gauss-Seidel a cada novo valor obtido nas iterações ser empregado nos cálculos posteriores, a convergência (geralmente) é “mais rápida” (menos iterações”).

```
function jacobi(A,VI,prec)

%Profa. Leila Z. Puga
%Essa function determina pelo método iterativo de Jacobi a solução de SL
%Admite-se que SL esteja ordenado e é convergente
%A é a matriz completa associada ao SL
%VI é a matriz com o chute/valores iniciais
%prec é a precisão desejada

[L,C]=size(A);
n=L;
m=C;
k=2;
x=VI;
x(:,2)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>prec
    k=k+1;
    if cont==1
        k=k-1;
        cont=0;
    end
    for i=1:n
        x(i,k)=A(i,m);
        for j=1:n
            if j~=i
                x(i,k)=x(i,k)-A(i,j)*x(j,k-1);
            end
        end
        x(i,k)=x(i,k)/A(i,i);
    end
end
jacobi=x
end
```

Resolvendo o Exercício (7) no *Editor* digitamos:

```
A=[1 -0.5 0.1 -1.2;0.2 1 -0.3 0.5;0.3 -0.1 -1 1.5]  %matriz associada a SL
format bank   %formatação com 2 casas decimais
% matriz A , chute inicial e precisão 0.001 são:
jacobi(A , [-1.2 ; 0.5 ; -1.5] , 0.001)
```

Em *Command Window* temos:

```
A =
    1.0000   -0.5000    0.1000   -1.2000
    0.2000    1.0000   -0.3000    0.5000
    0.3000   -0.1000   -1.0000    1.5000
jacobi =
    -1.20    -0.80    -0.86    -0.98    -0.95    -0.94    -0.95    -0.95
     0.50     0.29     0.09     0.14     0.17     0.15     0.15     0.15
    -1.50    -1.91    -1.77    -1.77    -1.81    -1.80    -1.80    -1.80
```

1.3 Exercícios: Resolução a mão (use os conceitos teóricos vistos em aula) e também no Matlab.

(I) Resolver os sistemas lineares pelos métodos da triangularização de Gauss e iterativo de Gauss-Seidel (com prec=0.01), se possível. Efetuar mudança de variável, se necessário. Usar duas casas decimais.

(a) $\begin{cases} x - 2y + 3\,\text{sen}\,z = 2{,}5 \\ x - 2y + \text{sen}\,z = 0{,}5 \\ x + y - 2\,\text{sen}\,z = 2{,}0 \end{cases}$ **(b)** $\begin{cases} 2x + y + 4z = 1 \\ 4x + y + 2z = 1 \\ x + 4y + 2z = 1 \end{cases}$

Respostas: **(a)** x =2.50 , y=1.50 e $z=\dfrac{\pi}{2}+k\pi\ ,k\in \mathbb{Z}$, **(b)** x=y=z=0,14

(II)

A gasolina é uma mistura de elementos químicos chamados hidrocarbonetos. A combustão acontece quando ela reage com o gás oxigênio resultando em gás carbônico e água.
A reação de combustão da gasolina pode ser dada pela equação

$$x.C_8H_{18} + y.O_2 \rightarrow z.CO_2 + t.H_2O$$

Sabendo-se que z=16, determine quantas moléculas são liberadas de água.

Resposta: x=2, y=25 e t=18 (moléculas de água)

(III) Permutando-se a ordem das equações do sistema linear abaixo, mostre que o Critério da Soma por Linhas não garante a convergência do sistema, mas que o Critério de Sassenfeld está satisfeito. Portanto, o sistema é convergente e pode ser resolvido por um método iterativo. Resolva por Jacobí e Gauss-Seidel (com prec=0.01) .

$$\begin{cases} 3x + 3y - 5z = 2 \\ 10x + 3y + 2z = -20 \\ 2x + 5y - 3z = 10 \end{cases}$$

Resposta: x=–2.81 , y=2.92 e z= –0.33

Algumas Observações:

(1) Elaborar uma *function* em Matlab para verificar se os maiores valores, de cada equação de um sistema linear SL, estão na diagonal principal da matriz A associada ao SL.

(2) Elaborar um programa em Matlab para verificar os 2 critérios de convergência: Sassenfel e Soma por Linhas.

2. ZEROS DE EQUAÇÕES

2.1 Métodos da Dicotomia e Newton-Raphson

Exercícios (8*): Determinar as raízes (zeros) de cada equação, levando em conta os 3 itens seguintes:

(a) x.ln(x)–3=0 , com precisão $\varepsilon \leq 0{,}05$; **(b)** $e^{x-1} - \sqrt{x} = 0$, com precisão $\varepsilon \leq 0{,}005$

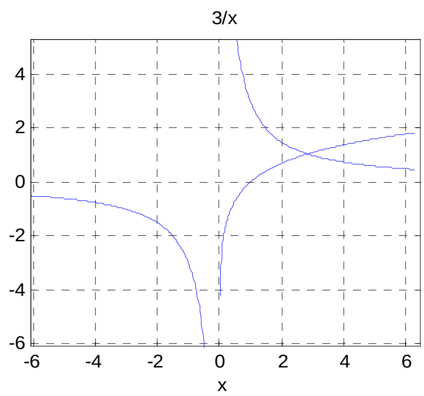
(I) Localizar graficamente todas as raízes, determinando seus intervalos

(II) Calcular a maior raiz com 3 c.d., pelos métodos da dicotomia e de Newton-Raphson

(III) Usar a *function fzero* do Matlab para calcular a raiz.

Resolução (8a):

A equação x.ln(x)–3=0 pode ser escrita na forma $\underbrace{\ln(x)}_{F} = \underbrace{\frac{3}{x}}_{G}$.



Pelos gráficos de F e G vemos que há só uma intersecção e, portanto, uma única raiz $\alpha \in]2 , 4[$.

Determinando um intervalo de amplitude unitária, pelo Teorema de Bolzano, temos:

$$f(x) = x.\ln(x) - 3$$

$$\left.\begin{array}{l} f(2) = -1.6137 < 0 \\ f(3) = 0.2958 > 0 \end{array}\right\rangle \Rightarrow \alpha \in]2 , 3[$$

Vamos calcular, inicialmente, a raiz α com precisão $\varepsilon \leq 0{,}05$ pelo método da dicotomia.

Seja x_0=2,5 e calculando f(2,5) < 0. Como f(2,5) < 0 e f(3) > 0 então a raiz está em] 2,5 ; 3 [.

Consideramos, agora, o ponto médio desse intervalo x_1=2,75. Como f(2,75)< 0 e f(3) > 0 então a raiz está no intervalo] 2,75 ; 3 [. Continuamos, com esse procedimento, até que $|x_n - \alpha| \leq 2\varepsilon$ (erro menor ou igual ao dobro da precisão ε). A tabela abaixo resume os cálculos obtidos:

			Intervalo da raiz	Ponto médio	$\mid x_n - \alpha \mid \leq 0{,}1$
F(2) < 0		F(3) > 0	]2 ; 3 [	$x_0 = 2{,}5$	0,5
F(2) < 0	F(2,5) < 0	F(3) > 0	]2,5 ; 3 [	$x_1 = 2{,}75$	0,25
F(2,5) < 0	F(2,75) < 0	F(3) > 0	]2,75 ; 3 [	$x_2 = 2{,}875$	0,125
F(2,75) < 0	F(2,875) > 0	F(3) > 0	]2,75; 2,875 [	$x_3 = 2{,}8125$	0,0625
				$x_4 = 2{,}84375$	

Note que no cálculo $| x_3 - \alpha | = 0{,}0625 \leq 2\varepsilon$ é menor que o dobro da precisão desejada.

Isso significa que basta considerar o ponto médio deste último intervalo e temos a raiz com a precisão pedida, $\alpha \cong 2{,}84375$.

Pelo método de Newton-Raphson, calculando a raiz α com precisão $\varepsilon \leq 0{,}05$, temos:

$f(x)=x.\ln(x)-3$ e $f'(x)=\ln(x)+1$;

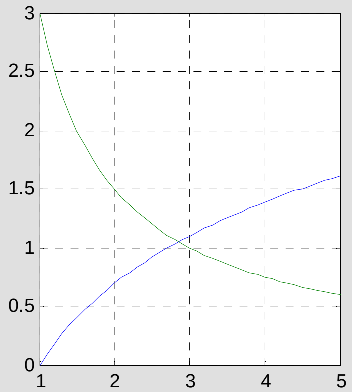
$$x_{k+1} = x_k - \frac{x_k.\ln(x_k) - 3}{\ln(x_k) + 1}$$

Como a raiz $\alpha \in]\,2\,,\,4\,[$ então consideremos $x_0=2$.

Assim, nas iterações, temos: $x_1=2.5$, $x_2=2.870$, $x_3=2.857$ e $x_4=2.857$. A raiz é $\alpha=2{,}857$.

Para a resolução no Matlab, digitamos no *Editor:*

```
% Localização gráfica das raízes
%x é o domínio/intervalo ]1,5[ de visualização
x=1:0.1:5   ;
% Decompondo f em y1 e y2, Note o sinal . em y2
y1=log(x)   ;
y2=3./x    ;
% gráfico das funções y1 e y2
plot(x,y1,x,y2)  %function plot para fazer os gráficos de y1 e y2 em função de x
grid    % grade/plano quadriculado
```



Digitando no *Editor* e obtendo as respostas em *Command Window:*

Editor

```
% Separação das raízes
% passo unitário no intervalo [1,4]
x=1:1:4 ;
y=x.*log(x)-3 ;   % equação dada no enunciado
% verificar a troca de sinal pelo Teorema Bolzano
trocasinal=[x' ,  y']
```

Command Window

```
trocasinal =
    1.0000   -3.0000
    2.0000   -1.6137
    3.0000    0.2958
    4.0000    2.5452
```

Para calcular a raiz pelo método da dicotomia/bissecção precisamos, inicialmente, implementar um programa, a *function dicot* em Matlab:

```
function dicot(f,a,b,prec)

%Profa Leila Zardo Puga
%Esta função dicot calcula a raiz da função f pela dicotomia ou bissecção
%Admite-se que f tenha uma só raiz no intervalo [a,b].
%A precisão desejada é dada pelo usuário em prec.
%A função f deverá estar implementada ou ser simbólica ou inline ou @

fa=f(a);  %valor numérico de f no extremo a do intervalo
fb=f(b);   %valor numérico de f no extremo b do intervalo
it=0;   %início da contagem das iterações
raiz=(a+b)/2;   %ponto médio
while abs(b-a)>2*prec %precisão desejada
    fraiz=f(raiz);
    if fa*fraiz>0  %a raiz está a direita de y
        a=raiz;
    else
        b=raiz;
    end
raiz=(a+b)/2;
it=it+1;
end
disp('O número it de iterações é:'), it
disp('A raiz aproximada da equação é:'),raiz
end
```

Em *Command Window*, as respostas são as seguintes:

```
>> syms  x
F=input('Entre com a expressão da função dada: f(x)=  ');
f=inline(F);
Entre com a expressão da função dada: f(x)= x*log(x)-3
>> a=2 , b=3 , prec=0.05
a =
    2
b =
    3
prec =
   0.0500
>> dicot(f,a,b,prec)
O número it de iterações é:
it =
    4
A raiz aproximada da equação é:
raiz =
   2.8438
```

Para calcular a raiz α pelo método de Newton-Raphson precisamos, inicialmente, implementar a seguinte *function NR* em Matlab:

```
%Profa. Leila Zardo Puga
%A function NR calcula a raiz de uma equação pelo método de Newton-Raphson
%Devem ser dados, f (inline ou syms ou @), um valor inicial vi que está no
intervalo que contém a raiz e também dar a precisão desejada prec.

function NR(f,vi,prec)
syms x
F=input('Entre com a expressão da função dada: f(x)= ');
f=inline(F);
deriv=diff(F);  %determinando a derivada de f
df=inline(deriv);   %definindo a derivada como função
prec=input('Entre com a precisão desejada: prec= ');
vi=input('Entre com o valor inicial: x0= ');
k=1;   %contador do número de iterações
x(k)=vi;  % valor inicial
x(k+1)=x(k)-f(x(k))/df(x(k));  %fórmula de Newton-Raphson
while abs(x(k+1)-x(k))> prec
    k=k+1;
    x(k+1)=x(k)-f(x(k))/df(x(k));
end
r=eval(x);
fprintf('Foram feitas k iterações, k=%2.0f\n',k);
disp('Os valores obtidos em cada iteração são:  ')
raiz=r' ;  %mostra os valores de cada iteração
raiz= [1:k+1 ; r]'
end
```

Digitando em *Command Window,* obtemos:

```
>> Entre com a expressão da função dada: f(x)= x*log(x)-3
Entre com a precisão desejada: prec= 0.05
Entre com o valor inicial: x0= 2
Foram feitas k iterações, k= 3
Os valores obtidos em cada iteração são:
raiz =
    1.0000    2.0000
    2.0000    2.9531
    3.0000    2.8581
    4.0000    2.8574
```

Para o item (c), usando a *function fzero* do Matlab, para calcular a raiz, temos:

Editor

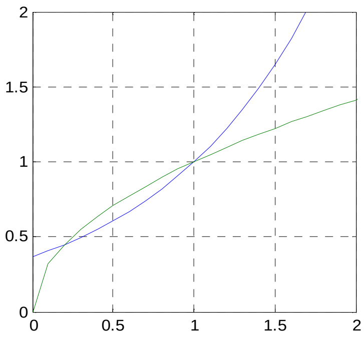
```
syms  x
F=input('Entre com a expressão da
função dada: f(x)= ');
f=inline(F);
disp('A raiz usando a function fzero
do matlab é:')
raiz=fzero(f,2)
```

Command Window

```
Entre  com  a  expressão  da
função dada: f(x)= x*log(x)-3
A  raiz  usando  a  function
fzero do matlab é:
raiz =
   2.857390783514366
```

Resolução (8b)

A equação $e^{x-1} - \sqrt{x} = 0$, com precisão $\varepsilon \leq 0{,}005$, pode ser escrita na forma $\underbrace{e^{x-1}}_{F} = \underbrace{\sqrt{x}}_{G}$.



Pelos gráficos de F e G vemos que há 2 raízes. Determinando um intervalo, para cada uma delas, pelo Teorema de Bolzano temos:

$$f(x) = e^{x-1} - \sqrt{x}$$

$$\left.\begin{array}{l} f(1) = 0 \\ f(0) = 0.3679 \quad > 0 \\ f(0.5) \;=\; -0.1006 \quad < 0 \end{array}\right\rangle \Rightarrow \quad \alpha = 1 \;;\; \beta \in \,]\,0\,,\,0.5\,[$$

Vamos calcular a raiz β com precisão $\varepsilon \leq 0{,}005$ pelo método da dicotomia.

No Matlab em *Command Window* temos:

```
syms  x
F=input('Entre com a expressão de f(x)=  ');
f=inline(F);
Entre com a expressão de f(x)= exp(x-1)-sqrt(x)
>> a=0   ,  b=0.5   ,   prec=0.005
a =
     0
b =
    0.5000
prec =
    0.0050
>> dicot(f,a,b,prec)
O número it de iterações é:
it =
     6
A raiz aproximada da equação é:
raiz =
    0.2070
```

Calculando no Matlab a raiz β com precisão $\varepsilon \leq 0{,}005$ pelo método de Newton-Raphson usando a *function NR:*

```
function NR(f,vi,prec)
syms x
F=input('Entre com a expressão da função f(x)= ');
f=inline(F);
```

```
deriv=diff(F);
df=inline(deriv);
prec=input('Entre com a precisão desejada: prec= ');
vi=input('Entre com o valor inicial: x0= ');
k=1;
x(k)=vi;   % valor inicial
x(k+1)=x(k)-f(x(k))/df(x(k));
while abs(x(k+1)-fx(k)>prec
    k=k+1;
    x(k+1)=x(k)-f(x(k))/df(x(k));
end
r=eval(x);
fprintf('Foram feitas k iterações, k=%2.0f\n',k);
disp('Os valores obtidos em cada iteração são:  ')
raiz=r'  %mostra os valores de cada iteração
end
```

Em *Command Window* temos as seguintes respostas:

```
>> NR
Entre com a expressão da função f(x)= exp(x-1)-sqrt(x)
Entre com a precisão desejada: prec= 0.005
Entre com o valor inicial: x0= 0.1
Foram feitas k iterações, k= 3
Os valores obtidos em cada iteração são:
raiz =
    0.1000
    0.1769
    0.2015
    0.2032
```

Para o item (c), usando a *function fzero* do Matlab para calcular as 2 raízes, temos:

Editor

```
syms  x
F=input('Entre com a expressão
da função dada: f(x)= ');
f=inline(F);
disp('A raiz usando a function
do matlab é:')
raiz1=fzero(f,0.5)
raiz2=fzero(f,1)
```

Command Window

```
>> syms  x
F=input('Entre com a expressão da
função dada: f(x)= ');
Entre com a expressão da função
dada: f(x)= exp(x-1)-sqrt(x)
A raiz usando a function do
matlab é:
>> raiz1=fzero(f,0.5)
raiz1 =
   0.203187869979980
>> raiz2=fzero(f,1)
raiz2 =
     1
```

2.2 Exercícios: Resolver no Matlab

(I) Fazer o gráfico e calcular a *maior raiz* α de cada equação por um dos 3 métodos vistos.

(a) $(x+1)^2 - \frac{1}{x^2} = 0$ (α=0,618) **(b)** $ln(x+4) - e^x = 0$ (α=0,392) **(c)** $x^2 + \ln(x^2) = 4$ (α=0,753)

(d) $x.\ln(x+1) = 5$ (α=3,383) **(e)** $sen(x) + 2x + 1 = 0$ (α=-0,335) **(f)** $(x+1)^2 - e^{x-1} = 0$ (α=-0.536)

(g) $(x-1)^2 = \frac{1}{x}$ (α=1,755) **(h)** $x^2 - \sqrt{3x} = 0$ (α=1,442) **(i)** $x^3 - (x+1)^2 = 0$ (α=2,148)

(II) **(P1 2015)** Uma moeda é atirada para o alto com trajetória dada por $f(x)=2x+x^2+3x^3-x^4$. A que distância, no chão, do ponto ponto onde foi arremessada a moeda começa a cair? Resolver pelo método de Newton-Raphson. Duas casas decimais. *Resposta:* A distância é f(2,53)=19,07 .

(III) **(P1 2014)** A equação de Kepler, usada para determinar órbitas de satelite é dada por M=x–E.sen(x). Dado que M=0,5 e E=0,2 resolva a equação utilizando o método de Newton-Raphson com duas casas decimais. *Resposta:* A raíz é r=0,62.

(IV) **(P1 2013)** Dada a função $y = x^4 - 3x^3 - 5x^2 - 4x - 1$ calcular **a** tal que y(a) < y(x) para todo o x no domínio da função y. Em qual intervalo está esse valor de a? Escrever a fórmula de Newton-Raphson adaptada ao problema. Duas casas decimais de precisão. *Resposta:* α=3,15 .

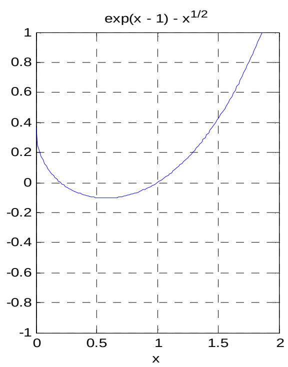
(V) **(P1 2012)** Um trem sai da cidade A para a cidade B com a seguinte equação de movimento $x_1(t)=30-80.e^{-t}$. No mesmo horário sai um trem de B para a cidade A com equação do movimento dada por $x_2(t)=80.t$. Após quanto tempo esses dois trens se cruzam, sabendo-se que a distância entre A e B é 190km? Resolver pelo método de Newton-Raphson. Duas casas decimais. *Resposta:* Os dois trens se cruzam após t= 2,12

Algumas Observações:

Ao lado temos o gráfico do Exercício (8b), $f(x) = e^{x-1} - \sqrt{x}$.

É interessante notar que no uso da fórmula iterativa de Newton-Raphson o valor inicial x_0=0,5 é, praticamente, quando a derivada f '(x)=0 e daí a reta tangente é horizontal. Isso significa, que os cálculos poderão ser "problemáticos"; ou seja, na fórmula iterativa podemos ter valor nulo (ou próximos) em denominador.

Por isso, usamos nos cálculos um outro valor inicial x_0=0,1.



3. Ajuste de Curvas - Método dos Mínimos Quadrados (MMQ)

3.1 Ajuste Polinomial

Exercício 9*: Num estudo sobre a retificação de um trecho de rio foram coletados os dados fornecidos na tabela abaixo e julgou-se conveniente que o traçado fosse ajustado por um polinômio do 2º grau. **(a)** Usando o MMQ, determine qual é a equação dessa curva. **(b)** Usando essa curva, calcule o valor do polinômio em 2,49. **(c)** Qual é o resíduo total? Cálculos com 2 casas decimais.

x	1	2	3	4
y	1,8	2,9	3,5	2,6

Resolução: O ajuste pelo polinômio de grau 2 é $y \approx P_2(x) = a + bx + cx^2$, sendo as funções g_0, g_1 e g_2 dadas por

$g_0(x) = 1$, $g_1(x) = x$ e $g_2(x) = x^2$.

Vamos usar o seguinte dispositivo:

	x_i	**a** $g_0(x_i)=1$	**b** $g_1(x_i)=x$	**c** $g_2(x_i)=x^2$	$f(x_i)$
calculando g_0, g_1 e g_2 nos pontos tabelados	1	1	1	1	1,8
	2	1	2	4	2,9
	3	1	3	9	3,5
	4	1	4	16	2,6
coeficientes do sistema normal		4	10	30	10,8
		10	30	100	28,5
		30	100	354	86,5
triangularização de Gauss			20	100	6
			100	516	22
				320	-160

Da última linha do dispositivo temos que 320c=–160 e, portanto, c=–0,50 .

Na equação: 20b+100c=6 $\Rightarrow$ b=2,80 . Da equação 4a+10b+30c=10,8 temos que a=–0.55 .

A equação da curva ajuste é $f(x)=-0,55+2,80x-0,50x^2$. O valor do polinômio em 2,49 vale aproximadamente f(2,49)=3,32. Para calcular o resíduo total, construímos a seguinte tabela:

x	y	f(x)	$\varepsilon_i = \lvert y - f(x) \rvert$
1	1,8	1,75	0,05
2	2,9	3,05	0,15
3	3,5	3,35	0,15
4	2,6	2,65	0,05

O resíduo total é $\sum_{i=1}^{4} \Sigma(\varepsilon_i)^2 = 0,05$.

A resolução desse exercício no Matlab, no *Editor*, é a seguinte:

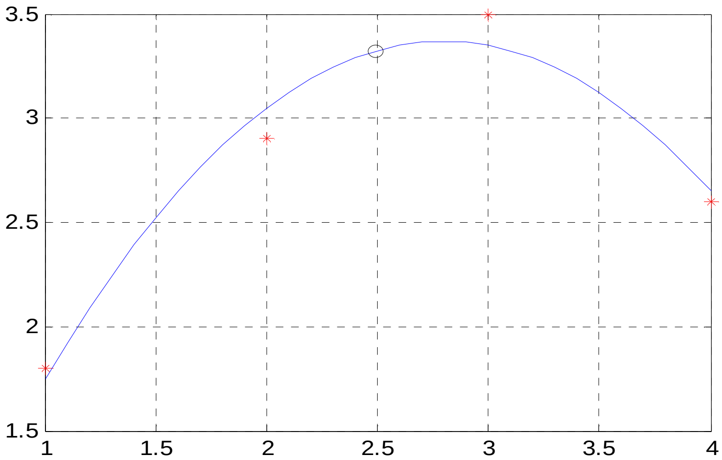
```
x=[1; 2; 3 ;4]    %dados da tabela disponibilizados em coluna
y=[1.8 ; 2.9 ; 3.5 ; 2.6]    %dados tabelados de y no enunciado
g=[x.^0    x.^1    x.^2 ] ;  %funções g para o ajuste desejado
SN= g'*[g , y]   %obter o sistema normal SN
c=rref(SN)   %resolver o SN, obtendo coeficientes do polinômio de grau 2

disp('fz é a expressão algébrica do polinômio:')
syms  z    %variável simbólica z
fz=c(1,4)+c(2,4)*z+c(3,4)*z^2  %polinômio de grau 2 na variável z
fz=vpa(fz,2)    %variável de precisão aritmética de 2 dígitos

Y=c(1,4)+c(2,4).*x+c(3,4).*x.^2; %valores do polinômio de grau 2 em x
tab=[x ; y ; Y ; y-Y]' %tabela de resíduos
res=sum((y-Y).^2)   %resíduo total

P= 2.49   ,  %dado do enunciado
fP=c(1,4)+c(2,4)*P+c(3,4)*P^2  %valor do polinômio em P

fx=min(x):0.1:max(x);  %domínio de visualização gráfica
fy=c(1,4)+c(2,4).*fx+c(3,4).*fx.^2; %valores do polinômio em fx
%gráfico com x e y grafados em vermelho com o símbolo *, curva fy na cor
azul e par ordenado (P,fP) grafados com o símbolo o na cor preta.
plot(x,y,'*r',fx,fy,'b' , P , fP, 'ko');
grid   %plano quadriculado , grade
```

Em *Command Window*, obtemos:

```
x =
          1.00
          2.00
          3.00
          4.00
y =
          1.80
          2.90
          3.50
          2.60
SN =
          4.00        10.00         30.00         10.80
         10.00        30.00        100.00         28.50
         30.00       100.00        354.00         86.50
c =
          1.00            0             0          -0.55
             0         1.00             0           2.80
             0            0          1.00          -0.50

fz é a expressão algébrica do polinômio:
fz =
- z^2/2 + (14*z)/5 - 77405618595439/140737488355328
```

```
fz =
- 0.5*z^2 + 2.8*z - 0.55

tab =
          1.00          1.80          1.75          0.05
          2.00          2.90          3.05         -0.15
          3.00          3.50          3.35          0.15
          4.00          2.60          2.65         -0.05
res =
          0.05
```

O gráfico da equação da curva $f(x)=-0,55+2,80x-0,50x^2$ é dado abaixo.



3.2 Ajuste Trigonométrico – Análise Harmônica

Exercício 10: Dada a tabela abaixo fazer a análise harmônica, se possível.

x	1	2	3	4
f(x)	1,5	2,6	1,3	3,2

Resolução:

Para fazer uma análise harmônica é necessário que $x \in [1, 4]$ percorra todo o círculo trigonométrico com passo constante. Portanto, neste caso, devemos ter passo $h = \frac{2\pi}{4} = \frac{\pi}{2}$ que corresponde a seguinte atribuição:

t	0	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$
x	1	2	3	4
f(x)	1,5	2,6	1,3	3,2

A mudança de variável t=ax+b é dada por: $\begin{cases} 0 = a + b \\ \frac{\pi}{2} = 2a + b \end{cases} \Rightarrow \begin{cases} a = \frac{\pi}{2} \\ b = \frac{-\pi}{2} \end{cases} \Rightarrow t = \frac{\pi}{2}(x-1)$

A curva de ajuste é $y = a_0 + a_1.\cos t + b_1.\operatorname{sen} t + a_2.\cos 2t$, para $t = \frac{\pi}{2}(x-1)$.

	a_0 1	a_1 cos(t)	b_1 sen(t)	a_2 cos(2t)	F(t)
	1	1	0	1	1,5
	1	0	1	–1	2,6
	1	–1	0	1	1,3
	1	0	–1	–1	3,2
coeficientes do sistema normal	4	0	0	0	8,60
	0	2	0	0	0,20
	0	0	2	0	–0,60
	0	0	0	4	–3,00

O sistema já está escalonado e sua solução é: $4a_0 = 8{,}60 \Rightarrow a_0 = 2{,}15$;

$2a_1 = 0{,}20 \Rightarrow a_1 = 0{,}10$; $2b_1 = -0{,}60 \Rightarrow b_1 = -0{,}30$ e $4a_2 = -3{,}00 \Rightarrow a_2 = -0{,}75$

A curva de ajuste é $y = 2{,}15 + 0{,}10.\cos t - 0{,}30.\operatorname{sen} t - 0{,}75.\cos 2t$, com $t = \frac{\pi}{2}(x-1)$.

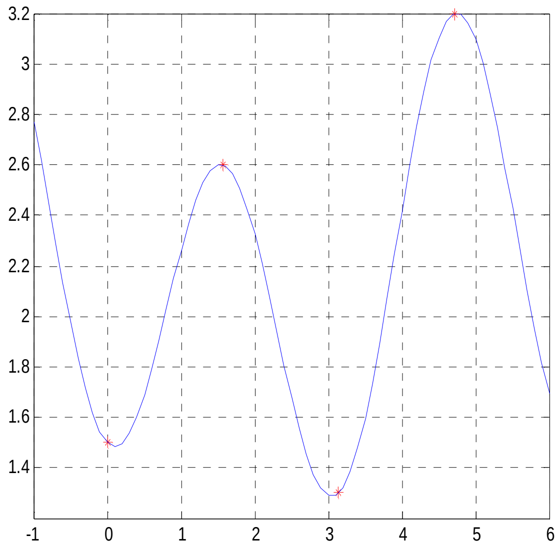
A resolução desse exercício no Matlab, janela *Editor,* segue abaixo:

```
% Análise Harmônica
x=[1  2  3  4]   %dados tabelados
t=(x-1)*(pi/2) ,  %mudança de variável
y=[1.5  2.6  1.3  3.2]   %dados da tabela
n=size(t) ;    %número de elementos em t
g0=ones(n) ;  g1=cos(t) ;  g2=sin(t) ;  g3=cos(2*t) ;  %funções de ajuste
g=[g0 ; g1 ; g2 ; g3]'   %dispositivo para SN
G=g'*g ,  b=g'*y' ,  c=inv(G)*b   %resolvendo o SN
syms z    %variável simbólica
fz=c(1)+c(2).*cos(z)+c(3).*sin(z) +c(4).*cos(2*z)  %curva de ajuste
disp('para z=(x-1)*(pi/2)')  %mudança de variável feita
disp('O ajuste por análise harmônica é:')
ah=vpa(fz,3)  %variável de precisão aritmética para fz com 3 dígitos
%Gráfico
m=-1:0.1:6;  %domínio de visualização gráfica (eixo horizontal)
fm=c(1)+c(2).*cos(m)+c(3).*sin(m) +c(4).*cos(2*m); %cálculos de fz em m
plot( t , y , 'r*', m , fm , 'b' ) ,  grid     %gráfico e grade
```

Em *Command Window* temos as respostas:

```
x =
     1     2     3     4
t =
         0    1.5708    3.1416    4.7124
y =
    1.5000    2.6000    1.3000    3.2000
g =
    1.0000    1.0000         0    1.0000
    1.0000    0.0000    1.0000   -1.0000
    1.0000   -1.0000    0.0000    1.0000
    1.0000   -0.0000   -1.0000   -1.0000
G =
    4.0000   -0.0000    0.0000         0
   -0.0000    2.0000    0.0000    0.0000
    0.0000    0.0000    2.0000    0.0000
     0        0.0000    0.0000    4.0000
b =
    8.6000
    0.2000
   -0.6000
   -3.0000
c =
    2.1500
    0.1000
   -0.3000
   -0.7500
fz =
cos(z)/10 - (3*cos(2*z))/4 - (3*sin(z))/10 + 43/20
para z=(x-1)*(pi/2)
O ajuste por análise harmônica é:
ah =
0.1*cos(z) - 0.75*cos(2.0*z) - 0.3*sin(z) + 2.15
```

O gráfico da curva de ajuste $y = 2{,}15 + 0{,}10.\cos t - 0{,}30.\text{sen}t - 0{,}75.\cos 2t$ é



3.3 Ajuste por outras famílias

Exercício 11: A tabela abaixo fornece a distância y entre um objeto e a origem de um sistema referencial sendo x o ângulo formado pelo objeto-referencial. Sabendo-se que $y = \dfrac{a}{1 + b..sen(x)}$, pede-se: **(a)** Determinar os valores de a e b. **(b)** Construir a Tabela de Resíduos, calculando o resíduo total.

x (radianos)	$\pi/2$	$3\pi/4$	$9\pi/8$
y (metros)	0,713	0,782	1,221

Resolução:

Linearizando a curva da família: $y = \dfrac{a}{1 + b.sen(x)} \Rightarrow \left\{ \underbrace{\dfrac{1}{y}}_{Y} = \underbrace{\dfrac{b}{a}}_{A} \underbrace{sen\ (x)}_{X} + \underbrace{\dfrac{1}{a}}_{B} \right.$

x	**B** 1	**A** X=sen(x)	$Y=\dfrac{1}{y}$
$\pi/2$	1	1	1,403
$3\pi/4$	1	0,707	1,279
$9\pi/8$	1	-0.383	0,819
	3	1,324	3,501
	1,324	1,647	1,994
		3,188	1,347

O sistema normal é $\begin{cases} 3B + 1{,}324A = 3{,}501 \\ 1{,}324B + 1{,}647A = 1{,}994 \end{cases}$ e resolvendo-o, temos: A=0,422 e B=0.983

A reta é Y=0,422 X+0.983 e, dai, a=1,02 e b=0,43 . A família é $f(x) = \dfrac{1.02}{1 + 0.43..sen(x)}$

Construindo a Tabela de Resíduos:

x	y	f(x)	ε_i=| y-f(x) |
$\pi/2$	0,713	0,713287	0,000287
$3\pi/4$	0,782	0,781957	0,000043
$9\pi/8$	1,221	1,219864	0,001136

O resíduo total é $\sum_{i=1}^{3} \Sigma(\varepsilon_i)^2 = 0{,}000001375 < 0{,}000002$.

No *Editor* do Matlab, digitamos:

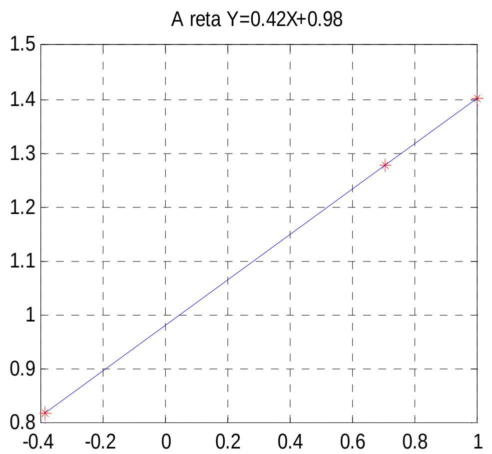
```
clear , clc , format short   %limpar, repaginar e formato com 4 c.decimais
disp('Ajuste de Curvas não-lineares nos parâmetros')
x=[pi/2 , 3*pi/4 , 9*pi/8]   %dados da tabela
y= [0.713 , 0.782 , 1.221]  %dados tabelados
disp('A linearização é:')  ,  X= sin(x)  ,  Y=1./y
L=polyfit(X,Y,1)  %linearização usando a function polyfit
disp('O vetor linha L fornece os coeficientes da reta Y=AX+B')
A=vpa(L(1),2) , B=vpa(L(2),2),  %usando 2 dígitos em vpa
disp('A equação da reta é')
syms Z  %variável simbólica Z
reta=poly2sym([A,B],Z)   %function poly2sym escreve a equação da reta Y
r=vpa(reta, 3)  %variável de precisão aritmética 3 dígitos significativos
disp('Os valores para a e b são:')
a= 1./B   ,  b=A*a  , %dados de mudança na linearização
disp('O ajuste da família é dada pela equação:')
'1.02/(1+0.43*sin(x))'
Fam=1.02./(1+b.*sin(x));
format short  ,  %formato com 4 casas decimais
res=y-Fam ;  T=[x;y;Fam;res]';  %res é o erro em cada ponto
disp('Tabela de Resíduos')  ,  Tab=eval(T)
disp ('Resíduo Total')  ,  RT=vpa(sum(res.^2),8)
disp('O gráfico da reta')
figure (1)   %nomeando o gráfico
R=subs(reta, Z,X);
plot(X,Y,'r*',X,R,'-b') ,  grid
title('A reta Y=0.42X+0.98') %colocando título no gráfico
disp('O gráfico da família')
figure (2)   %nomeando o gráfico
t= 0:0.01:5 ;  %domínio de visualização gráfica no eixo horizontal
Ft=1.02./(1+b.*sin(t));
plot(x,y,'or',t,Ft,'-b') , grid   %gráfico e plano quadriculado
title('A família 1.02/(1+0.43*sin(x))')  %colocando título no gráfico
```

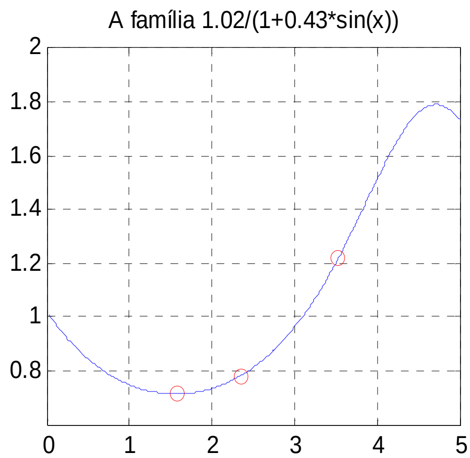
As respostas em *Command Window:*

```
Ajuste de Curvas não-lineares nos parâmetros
x =
    1.5708    2.3562    3.5343
y =
    0.7130    0.7820    1.2210
A linearização é:
X =
    1.0000    0.7071   -0.3827
Y =
    1.4025    1.2788    0.8190
L =
    0.4220    0.9805
O vetor linha L fornece os coeficientes da reta Y=AX+B
A =
0.42
```

```
B =
0.98
A equação da reta é
reta =
0.4219864742*Z + 0.9804696674
r =
0.422*Z + 0.98
Os valores para a e b são:
a =
1.019919364
b =
0.430392176
O ajuste da família é dada pela equação:
ans =
1.02/(1+0.43*sin(x))
Tabela de Resíduos
Tab =
    1.5708    0.7130    0.7131   -0.0001
    2.3562    0.7820    0.7820   -0.0000
    3.5343    1.2210    1.2211   -0.0001
Resíduo Total
RT =
0.000000023746661
```

Os gráficos da reta e da família:





Exercício 12: Sabe-se que a lei que rege o crescimento de uma população é dada por $y = a.b^t$, onde t é o tempo. Numa cidade, foram coletados os dados da tabela . Pede-se fazer uma previsão da população para 2016.

t (ano)	2000	2004	2008	2012
P(milhares de habitantes)	2,00	2,20	2,42	2,66

Resolução:

Linearizando: $P = a.b^t \Rightarrow \left\{ \underbrace{\ln(P)}_{Y} = \underbrace{\ln(b)}_{A}\, \underbrace{t}_{X} + \underbrace{\ln(a)}_{B} \right.$

Para facilitar os cálculos vamos fazer a mudança de variável: x=0,25t-500

Fazendo o ajuste pela reta Y=A+BX , que nesse caso, $g_0(X)=1$ e $g_1(X)=X$, que no dispositivo prático temos:

1	X	Y	
1	0	0,693	
1	1	0,788	
1	2	0,884	
1	3	0,978	
4	6	3,343	
6	14	5,490	$\Rightarrow$ 6B+14A=5,490 $\Rightarrow$ B=0,693
	20	1,902	$\Rightarrow$ 20A=1,902 $\Rightarrow$ A=0,095

A reta é Y=0,095X+0,693.

Como A=ln(b) temos b=1,1 e, ainda, sendo B=ln(a) obtemos a=2. A família é então $y = 2.(1{,}1)^x$

Uma previsão para t=2016 é quando x=4 e, portanto, $y=2.(1{,}1)^4$= 2,928 (milhares de habitantes).

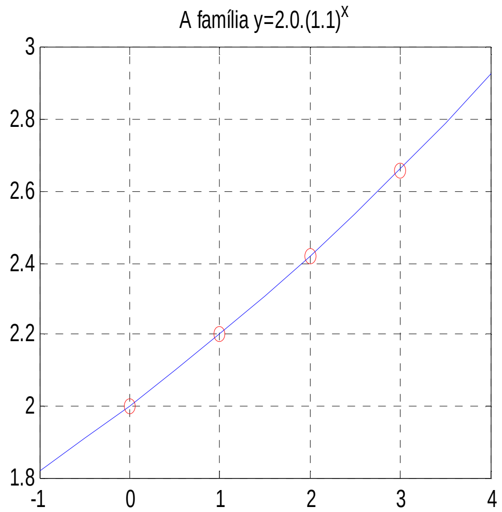
Digitando no *Editor*, a resolução no Matlab é:

```
disp('Ajuste de Curvas não-lineares nos parâmetros')
t=[2000 , 2004 , 2008 , 2012]  ,   P= [2.0 , 2.2 , 2.42 , 2.66]  ,
x= 0.25*t-500 % mudança de variável
disp('A linearização é:')  ,  X=x , Y=log(P),
L=polyfit(X,Y,1)  %  ajuste por um polinômio de grau 1
disp('O vetor linha L fornece os coeficientes da reta Y=AX+B')
A=vpa(L(1),2) , B=vpa(L(2),2),  % vpa com 2 digitos significativos
disp('Os valores para a e b são')  , a= exp(B) , b=exp(A)
disp('O ajuste pela família é') ,  'y=2.0.(1.1)^x'
disp(' Uma previsão para 2016 é:')
syms  z   %z é variável simbólica
Ajuste=a*(b.^z);
X=subs(Ajuste,z,4)  % note que se t=2016 então x=4
disp(' Fazendo o gráfico temos:')
u=-1:0.5:4 ;  %domínio de visualização para o gráfico
Fam=a*(b.^u) ;   % ajuste pela família
plot(x,P,'or',u,Fam) %gráfico que o par (x,P) é grafado em vermelho com o
title('A família y=2.0.(1.1)^x')
grid , shg  %grade e a function shg traz a tela para a frente
```

No Matlab em *Command Window,* as respostas são:

```
Ajuste de Curvas não-lineares nos parâmetros
t =
        2000        2004        2008        2012
P =
    2.0000    2.2000    2.4200    2.6600
x =
     0     1     2     3
A linearização é:
X =
     0     1     2     3
Y =
    0.6931    0.7885    0.8838    0.9783
L =
    0.0951    0.6933
O vetor linha L fornece os coeficientes da reta Y=AX+B
A =
0.095
B =
0.69
Os valores para a e b são
a =
2.00030066152441946689913423070 78
b =
1.09975200088633197067348692783 2
O ajuste pela família é
ans =
y=2.0.(1.1)^x
 Uma previsão para 2016 é:
X =
2.9260000000296
```

O gráfico da familia:



3.4 Exercícios: Resolver usando o Matlab

(1) Ajustar os dados da tabela por uma curva da família $y = a_0 + a_1.\text{sen}(2x) + a_2.\cos(3x)$. Fazer o gráfico. Cálculos com três casas decimais.

x	0	$\pi/5$	$\pi/2$	$7\pi/6$
f(x)	0	0,8	0,5	1

Resposta: F=0.401*sen(2.0*x) - 0.411*cos(3.0*x) + 0.464

(2) Ajustar os dados da tabela por $y = a.b^x$. Calcule um valor aproximado de f(3,55). Usar 3 casas decimais.

x	1	2	3	4
f(x)	0,67	1,00	1,20	1,53

Fazer o gráfico.

Respostas: $y = 0{,}542.(1{,}305)^x$.
f(3.55)=1.392

(3) A tabela abaixo mostra a queda da pressão P de uma caldeira em função da temperatura t. Sabendo-se que a família $P = \dfrac{t}{k + m.t}$ é adequada para representar o fato, fazer uma previsão para P se t=200ºC. Fazer o gráfico.

t (Cº)	500	400	300
P	50	25	15

Resposta: k=35 e m=–0,05 ; $P = \dfrac{t}{35 - 0{,}05\,t}$, se t=200ºC então P=7,89.

(4) Verificar graficamente qual das seguintes famílias melhor ajusta os dados da tabela abaixo:

Família 1: $y = a.e^{bx}$, *Família 2:* $y = \dfrac{x}{a + bx}$, *Família 3:* $y = a.x^b$, *Família 4:* $y = a.b^x$

x	1	2	3	4	5
y	0,5	2,0	5,0	10,0	18,0

Resposta: Pelo gráfico, a Família 3 tem mais pontos alinhados e, portanto, é a que melhor ajusta os dados tabelados.

(5) Os dados da tabela abaixo foram coletados durante a modelagem da carroceria do novo fusca, constatando que podem ser ajustados por um polinômio de grau 7. Determine graficamente essa curva .

x	0.98	1.18	1.4	1.78	2.23	2.95	3.7	4.4	5.18	5.9	6.9	7.5	8.7	9.4	10.5	11.9	12.5	13.2
y	1.5	2.15	2.5	2.9	3.22	3.3	3.4	3.8	4.2	4.65	4.9	4.95	4.98	4.9	4.75	4	3.31	1.4

4. Interpolação Polinomial

4.1 Polinômio de Lagrange

Exercício 13*: A tabela abaixo refere-se à potência P em função do número r de rotações do motor de um carro. Determine: **(a)** A expressão algébrica do polinômio de Lagrange; **(b)** Se r=2 *rpm* qual é a potência P do motor? **(c)** Se a potência é P=18 hp qual é o número de rotações r? **(d)** Delimitar o erro de truncamento em **r** admitindo-se que $f(z)= e^{0,1z}$ *(dados fictícios)* .

r *(100 **x** rpm)*	1	6	13
P *(100 **x** hp)*	5,4	10,4	25,8

Resolução:

O polinômio interpolador de Lagrange para três pontos tabelados é de grau ≤ 2 . Então:

$$P_2(x) = f(x_0).L_0^2(x) + f(x_1).L_1^2(x) + f(x_2).L_2^2(x), \text{ sendo:}$$

$$L_0^2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} = \frac{(x-6)(x-13)}{(1-6)(1-13)} = \frac{x^2-19x+78}{60}$$

$$L_1^2(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = \frac{(x-1)(x-13)}{(6-1)(6-13)} = \frac{x^2-14x+13}{-35}$$

$$L_2^2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{(x-1)(x-6)}{(13-1)(13-6)} = \frac{x^2-7x+6}{84}$$

Assim, $f(x) \approx P_2(x) = (5,4)\,\dfrac{x^2-19x+78}{60} + (10,4)\,\dfrac{x^2-14x+13}{-35} + (25,8)\,\dfrac{x^2-7x+6}{84}$.

A expressão algébrica do polinômio de Lagrange é $P_2(x) = 0,1x^2+0,3x+5$.

Se r=2 *rpm* então a potência P do motor é $P_2(2)$ = P=6hp.

Se a potência é P=18 hp então o número de rotações r é calculado por: $18=0,1x^2+0,3x+5 \Rightarrow x=10$

Para delimitar o erro de truncamento em **r,** admitindo-se que $f(z)= e^{0,1z}$ *(dados fictícios)*, temos:

O polinômio tem grau 2 e, portanto, o erro de truncamento é:

$$R_3(x) = |E_{tr}| < \frac{|(x-x_0)(x-x_1)(x-x_2)|}{(2+1)!}.\max\left|f^{(2+1)}(c)\right| \quad , \quad x_0 < c < x_2 .$$

Sendo $f(z)= e^{0,1z}$, a derivada de ordem 3 de f é $f'''(z) = (0,1)^3 e^{0,1z}$

$$|E_{tr}| < \frac{|(2-1)(2-6)(2-13)|}{(2+1)!}\max\left|(0,1)^3 e^{0,1c}\right| \quad , \ 1<c<13$$

$$|E_{tr}| = \frac{(1).(-4).(-11)}{6}(0.003669) = \frac{0,161436}{6} = 0,026906 \quad \Rightarrow \ |E_{tr}| < 0,03 .$$

Para resolver no Matlab precisamos programar a seguinte *function interplag*

```
%Profa. Leila Zardo Puga
%A function interplag determina o Polinômio Interpolador de Lagrange. Pode
%ser usada de 3 formas diferentes: para interpolar um valor, para vetor no
%caso de fazer gráfico e para obter expressão algébrica no caso simbólico.
%Devem ser dados
%x : vetor das abscissas (dados tabelados)
%y : vetor das ordenadas (dados tabelados)
%Z : valor ou vetor a ser interpolado ou simbólico Z

function L = interplag(x,y,Z)
n = length(x) ;
L= 0 ;
for k = 1:n;
P = 1;
for j=[1:k-1,k+1:n]
P = P.*((Z-x(j))./(x(k)-x(j)));
end
L = L+ P*y(k);
end
```

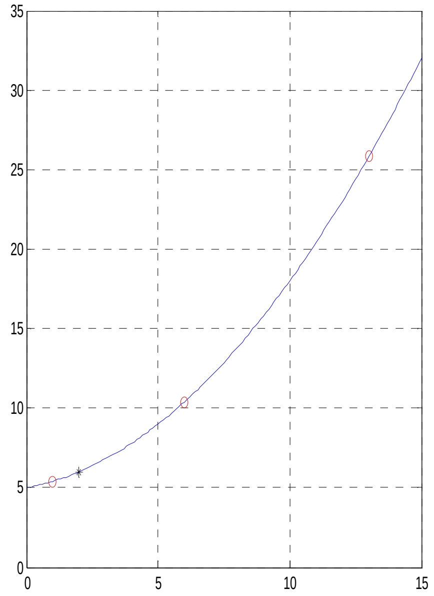
Digitando no *Editor* do Matlab o Exercício 13 temos:

```
x=[1 6 13]  %vetor das abscissas
y=[5.4  10.4  25.8]  %vetor das ordenadas
disp('A expressão algébrica do polinômio de Lagrange é:')
syms z  %variável simbólica z
LG = interplag(x,y,z);  %usando a function interplag para Z simbólico
PL=vpa(simplify(LG) ,3)    %PL é o polinomio de Lagrange

disp('Se r=2rpm a potência P do motor é:')
r= 2
P= interplag(x,y,r) %usando a function interplag para um valor r
disp('Se P=18 o numero de rotações r do motor é:')
rot=solve(18==PL)  %function solve para resolver equação

disp('O erro de truncamento em d é calculado por:')
syms t , %variável simbólica t
f=exp(0.1*t)  %função dada no enunciado do exercício
d3f=diff(f,3)  %diff determina a derivada, neste caso, de ordem 3
S=(subs(d3f,x));  %calcula o valor da 3ªderivada em x
M=max(eval(abs(S)))    % M é maior valor da derivada
et=abs(prod(r-x))*M/factorial(3) %fórmula para o erro de truncamento

disp('Gráfico do Polinômio de Lagrange')
u=0 : 0.1 : 15 ;   %domínio de visualização para o gráfico
w= interplag(x,y,u);  % function interplag em que w avalia o polinômio em u
plot(u, w, x, y, 'ro', r, P, 'k*')  ,  grid    %gráfico e grade
disp('Note que o polinômio passa pelos 3 pontos da tabela')
```

As respostas em *Command Window:*

```
x =
          1.00          6.00          13.00
y =
          5.40          10.40          25.80
A expressão algébrica do polinômio de Lagrange é:
PL =
0.1*z^2 + 0.3*z + 5.0
Se r=2rpm a potência P do motor é:
P =
          6.00
Se P=18 o numero de rotações r do motor é:
rot =
 -12.99999999
 9.9999999999
O erro de truncamento em d é calculado por:
f =
exp(t/10)
d3f =
exp(t/10)/1000
M =
   0.003669296667619

et =
   0.026908175562541
Gráfico do Polinômio de Lagrange
Note que o polinômio passa pelos 3 pontos da tabela
```

Gráfico do polinômio interpolador de Lagrange:



Exercício 14: Para percorrer 21 Km numa maratona um velocista gastou 30 minutos, admitindo-se velocidade constante. Após t=8 minutos, do início da corrida, havia percorrido d=7,36 Km. Pede-se: **(a)** Que distância o velocista percorreu nos primeiros t=20 minutos? **(b)** Quantos minutos gastou para chegar à metade do caminho?

Tempo **t** (min)	0	8	30
Distância **d** (km)	0	7,36	21

Considere todos os pontos da tabela.

Use 3 casas decimais (3 c.d.)

Resolução:

Para uma tabela com 3 pontos, o polinômio de grau 2 é $P_2(t)=a_0+a_1t+a_2t^2$.

Impondo que nos pontos tabelados os valores de P e d coincidam (interpol. por Lagrange): $P_2(t_0)=d(t_0)$ e assim

$$\begin{cases} a_0 + a_1t_0 + a_2t_0^2 = d(t_0) \\ a_0 + a_1t_1 + a_2t_1^2 = d(t_1) \\ a_0 + a_1t_2 + a_2t_2^2 = d(t_2) \end{cases}$$

Esse sistema linear pode ser escrito na forma matricial:

$$\underbrace{\begin{bmatrix} 1 & t_0 & t_0^2 \\ 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \end{bmatrix}}_{\text{Vandermont}} . \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} d(t_0) \\ d(t_1) \\ d(t_2) \end{bmatrix} \Rightarrow \underbrace{\begin{bmatrix} 1 & 0 & 0^2 \\ 1 & 8 & 8^2 \\ 1 & 30 & 30^2 \end{bmatrix}}_{\text{Vandermont}} . \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 7{,}36 \\ 21 \end{bmatrix} \Rightarrow \begin{cases} 1a_0 = 0 \\ 1a_0 + 8a_1 + 64a_2 = 7{,}36 \\ 1a_0 + 30a_1 + 900a_2 = 21 \end{cases}$$

Pelo dispositivo prático, resolvendo esse sistema linear, temos:

a_0	a_1	a_2	indep	
1	0	0	0	$\Rightarrow$ $a_0=0$
1	8	64	7,36	
1	30	900	21	
	8	64	7,36	$\Rightarrow$ $a_1=1$
	30	900	21	
		5280	–52,8	$\Rightarrow$ $a_2=-0{,}01$

O polinômio de grau 2 é $P_2(t) = t-0{,}01t^2$.

A distância que o velocista percorreu para t=20 minutos é de $(20)-(0{,}01)(20)^2 = 16$ Km .

Para chegar à metade do caminho ele gastou: $P_2(t) = 10{,}5 = t-0{,}01t^2 \Rightarrow 0{,}01t^2 - t + 10{,}5 = 0 \Rightarrow t = 11{,}92$ min

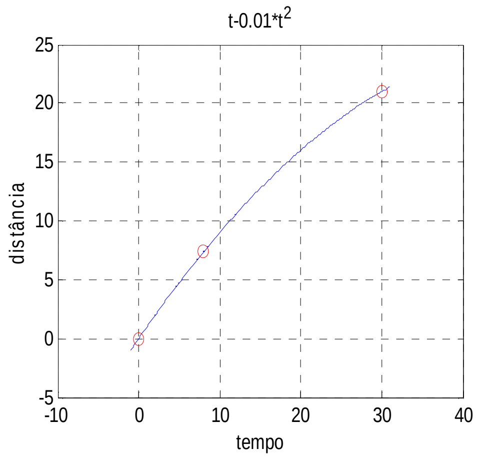
No *Editor* do Matlab, temos:

```
x=[0 , 8 , 30]  ,  y=[0 , 7.36 , 21]     %dados tabelados
V = vander(x)   %function vander para a matriz de Vandermonde
c=inv(V)*y'  %resolvendo o sistema linear
syms z   % z é variável simbólica
P=c(1)*z^2+c(2)*z+c(3)   %expressão do polinômio
PL=vpa(P,3) %variável de precisão aritmética com 3 dígitos significativos
Dist=subs(P,z,20)   %function subs que calcula o valor de P para z=20
Tempo=solve(PL==10.5)  %function solve do Matlab para resolver equações
```

As respostas em *Command Window:*

```
x =
     0     8    30
y =
     0    7.3600   21.0000
V =
     0       0      1
    64       8      1
   900      30      1
c =
   -0.0100
    1.0000
         0
P =
- z^2/100 + z
PL =
- 0.01*z^2 + z
Dist =
16
Tempo =
 11.92113447
 88.07886552
```

O gráfico do polinômio interpolador de Lagrange é:



4.2 Exercícios: Resolver no Matlab

(1) Se $f(x) = x^3+bx+c$, com a, b$\in \Re$, deseja-se obter uma aproximação de f(0,98) através do polinômio interpolador de Lagrange. Pode-se escolher uma tabela em que os valores de x têm passo h=0,5 e os valores de f(x) são arredondados para 4 casas decimais, como tabela 1, ou então utilizar a tabela 2 para f em que os valores de x têm passo h=0,05, mas os valores de f(x) são arredondados para 3 casas decimais. Qual é a melhor escolha?

Tabela 1	
x	f(x)
0,5	–
1,0	–

Tabela 2	
x	f(x)
0,95	–
1,0	–

Resposta: A Tabela 2 é melhor, pois fornece o menor erro de truncamento, sendo: $E_{tr1} < 0{,}0288$ e $E_{tr2} < 0{,}0018$.

(2) Usar o polinômio interpolador e todos os pontos da tabela para transformá-la numa tabela com passo unitário.

x	0	2	5
f(x)	2,8	3,1	5,6

Resposta: $P_2(x)=0{,}137x^2-0{,}123x+2{,}8$.

Tabela com passo unitário

x	0	1	2	3	4	5
f(x)	2,8	2,81	3,1	3,66	4,49	5,6

(3)

Cosmoland é um parque temático na Baía de Yokohama, dentro da região metropolitana de Tóquio, Japão, com a incrível montanha-russa mergulhadora *Vanish.* São 744 metros de comprimento e 35 metros de altura com uma passagem por um túnel sub-aquático que faz você sentir que está mergulhando em uma piscina.

Quando foi projetada, pensou-se em uma curva tal como pode ser vista na foto ao lado, cujas coordenadas (x,y) em escala reduzida, é dada na tabela abaixo. Por interpolação de Lagrange determine essa curva no plano cartesiano.

x	6,4	6,5	6,6	6,8	7,3	7,5	7,7	7,9	8
y	12,0	16	19,4	22,0	20,1	17,1	13,5	10,0	8,2

Resposta: Ver o gráfico .

5. Integração Numérica

5.1 Fórmula do Trapézio

Exercício 15:

A estrutura do *Hotel Burj Al Arab* em Dubay foi inspirada na vela de um barco. Uma parte do revestimento externo é de tecido similar a seda que preserva a luminosidade e temperatura (*parte branca na foto*). *Laurinha* em sua viagem a Dubay ficou curiosa para saber o custo e a quantidade de tecido utilizada nesse revestimento. Decidiu então tirar uma foto e obteve as medidas *(escala reduzida)* dadas na tabela. Admitindo-se que o valor do tecido por u.m.² é Ξ$ 1,47 mil , qual foi o custo do tecido empregado? Use somente a fórmula do Trapézio e 2 c.d. Delimitar o erro de truncamento supondo-se que $f(z) = \log(1 - 2.z)$. *(dados fictícios)*

x *(u.m.)*	10	15	20	25	30	35	40
y *(100 u.m.)*	89	76	71	68	65	52	43

Resolução:

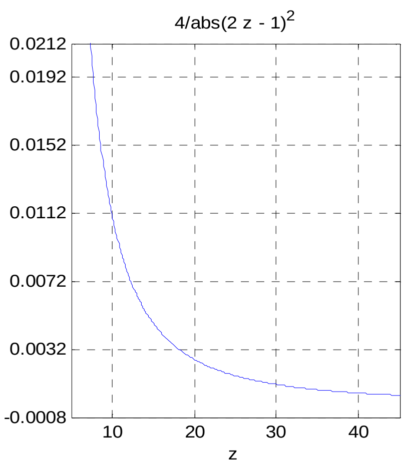
Devemos usar 6 vezes a fórmula do trapézio e, portanto, temos:

$$I = \frac{5}{2}.(89 + 2(76 + 71 + 68 + 65 + 52) + 43) = 1990$$

Assim, o custo do tecido empregado é de 1,47.(1990)=2925,5 mil Ξ$.

Para delimitar o erro de truncamento, temos que calcular as derivadas seguintes:

$$f(z) = \log(1 - 2.z) \;\Rightarrow\; f'(z) = \frac{-2}{(1-2.z)} \;\Rightarrow\; f''(z) = \frac{-4}{(1-2.z)^2}$$



Ao lado temos o gráfico de | f '' (z) |.

Pelo gráfico, podemos constatar que | f ''(z) | assume maior valor em z=10.

De fato:

$$f''(10) = \left| \frac{-4}{(1-2.10)^2} \right| = 0{,}01108033$$

$$f''(40) = \left| \frac{-4}{(1-2.40)^2} \right| = 0{,}00064092$$

Portanto, devemos considerar M=0,01108033

$$E_{tr} = n*(h\hat{}3/12)*M = 6.\frac{5^3}{12}0.01108033 = 0{,}692520625 < 0{,}7$$

A resolução no Matlab, com o *Editor*, é dada abaixo:

```
%Cálculo da integral pela Fórmula do Trapézio
disp('Os dados tabelados para t e V são:')
t=[10 , 15 , 20 , 25 , 30 , 35 , 40]
V=[89 , 76 , 71 , 68 , 65 , 52 , 43]
disp('Calculo da integral, a cada 2 pontos, pela Fórmula do Trapézio:')
h=5 ;  %passo h entre os pontos tabelados
T=(h/2)*(V(1)+2*(V(2)+V(3)+V(4)+V(5)+V(6))+V(7)) ;  %fórmula
disp('O custo do tecido empregado foi de:')
Ar=T  ;   Custo=1.47*Ar

disp('Cálculo do erro de truncamento pela Fórmula do Trapézio:')
syms  z  , f=log(1-2*z) , d2f=diff(f,2)   % derivada 2ª da função f
disp('Pelo gráfico de d2f vemos que o maior valor M é em x=10 ')
D2f=abs(d2f)  % valor absoluto da derivada 2ª da função f
ezplot(D2f , [5 , 45])  ,  grid    %function ezplot para fazer gráfico

disp('O maior valor M é:')
M=abs(subs(d2f , 10)) %calcular em módulo o valor de d2f em z=10

disp('Calculando os 6 erros de truncamento para fórmula do Trapézio:')
a=10 ;  b=40 ;   n=abs(b-a)/h ;
ErroTrunc=eval(n*(h^3/12)*M)  %fórmula do erro de truncamento para trapézio
```

As respostas obtidas em *Command Window* são as seguintes:

```
Os dados tabelados para t e V são:
t =
    10    15    20    25    30    35    40
V =
    89    76    71    68    65    52    43
Calculo da integral, a cada 2 pontos, pela fórm. do trapézio:
O custo do tecido empregado foi de:
Custo =
       2925.3
Cálculo do erro de truncamento pela Fórmula do Trapézio:
f =
log(1 - 2*z)
d2f =
-4/(2*z - 1)^2
Pelo gráfico de d2f vemos que o maior valor M é em x=10
D2f =
4/abs(2*z - 1)^2
O maior valor M é:
M =
4/361
Calculando os 6 erros de truncamento para fórmula do Trapézio:
ErroTrunc =
      0.69252
```

Usando a *function trapz* do Matlab, digitamos no *Editor* e obtemos as respostas em *Command Window:*

Editor	**Command Window**
%Cálculo usando a function trapz do Matlab disp('Os dados tabelados para t e V são:') t=[10 , 15 , 20 , 25 , 30 , 35 , 40] V=[89 , 76 , 71 , 68 , 65 , 52 , 43] I=trapz(t,V)	Os dados tabelados para t e V são: t = 10 15 20 25 30 35 40 V = 89 76 71 68 65 52 43 I = 1990

5.2 Fórmula de Simpson

Exercício 16

A engenheira *Carolina* quer proteger a piscina das folhagens das plantas de sua casa de campo em Atibaia. Pensou em comprar uma rede de proteção e como não sabe quantos m² tem sua piscina decidiu então fazer uma medição obtendo os dados registrados na tabela abaixo. Admitindo-se que o valor é $4,60 por m², quanto custará essa rede para a *Carolina*? Use só a fórmula de Simpson e 2 c.d. Delimitar o erro de truncamento sabendo-se que $f(t) = e^{(0.5-0.1.t)}$. *(dados fictícios)*.

x	0	4	8	12	16	20	24
y	21	25	23	19	15	21	16

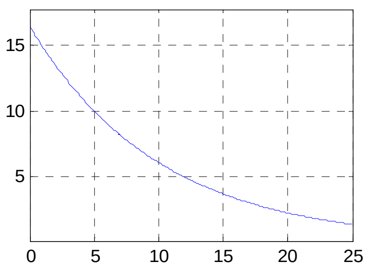
Resolução:

Devemos usar somente a fórmula de Simpson (3 vezes) e, portanto, temos:

$$I = \frac{4}{3}.\left(21 + 4(25 + 19 + 21) + 2(23 + 15) + 16\right) = 497{,}33$$

Assim, o custo da rede é de 4,60.(1990)=$2287,73

Para delimitar o erro de truncamento, usamos a 4ª derivada de f que é $f^{(IV)}(t) = (-0.1)^4.e^{0.5-0.1t}$



Ao lado temos o gráfico de $|f^{IV}(t)|$.

Pelo gráfico podemos ver que $|f^{IV}(t)|$ assume maior valor em t=0.

De fato:

$f^{IV}(0) = (-0.1)^4.e^{0.5-0.1(0)}$ = 0,000164872

$f^{IV}(24) = (-0.1)^4.e^{0.5-0.1(24)}$ = 0,000014957

Portanto, devemos considerar M=0,000164872

E_{tr} = n*(h^5/90)*M = $3.\frac{4^5}{90}$0,00016487 2 = 0,00562 < 0,006

No *Editor* do Matlab temos:

```
disp('Os dados tabelados para x e y são:')
x=[0 , 4 , 8 , 12 , 16 , 20 , 24]
y=[21 , 25 , 23 , 19 , 15 , 21 , 16]
disp('Calculo da integral, a cada 3 pontos, pela Fórmula de Simpson:')
h=4 ,  T=(h/3)*(y(1)+4*(y(2)+y(4)+y(6))+2*(y(3)+y(5))+y(7))
disp('O custo da rede de proteção é de')
Ar=T  ;  C=4.6*Ar  ,  Custo=vpa(C,6)
disp('Cálculo do erro de truncamento pela Fórmula de Simpson:')
syms  t   ,   f=exp(0.5-0.1*t)
disp('A 4ª derivada da função é:')
d4f=diff(f , 4),  D4f=abs(d4f)
ezplot(D4f , [0 , 25]) , grid
disp('Pelo gráfico, o maior valor M é em t=0 ')  ,  M=subs(d4f , 0)
disp('Calculo dos 3 erros de truncamento para fórmula de Simpson:')
a=0 ; b=24 ; n=abs(b-a)/(2*h) ;
disp('O erro de truncamento é menor ou igual a:')
Erro=n*(h^5/90)*M
```

Em *Command Window* temos as seguintes respostas:

```
Os dados tabelados para x e y são:
x =
     0     4     8    12    16    20    24
y =
    21    25    23    19    15    21    16
Calculo da integral, a cada 3 pontos, pela fórmula de Simpson:
h =
     4
T =
  497.33
O custo da rede de proteção é de
C =
   2287.7
Custo =
2287.73
Cálculo do erro de truncamento pela Fórmula de Simpson:
f =
exp(1/2 - t/10)
A 4ª derivada da função é:
d4f =
exp(1/2 - t/10)/10000
D4f =
exp(1/2 - real(t)/10)/10000
Pelo gráfico, o maior valor M é em t=0
M =
0.00016487
Calculo dos 3 erros de truncamento para fórmula de Simpson:
O erro de truncamento é menor ou igual a:
Erro =
0.00562763527
```

Exercício 17*: Calcular um valor aproximado de $\int_1^2 \frac{1}{x}dx$ pela fórmula de Simpson com uma delimitação do erro de truncamento $E_{tr} \leq 0{,}0005$, de modo que a tabela de f no intervalo de integração tenha o menor número de pontos. Usar 3 casas decimais.

Resolução:

Vamos admitir que temos **n** integrais de Simpson, $n \in \mathbb{N}^*$ e, portanto, 2n intervalos cada um de amplitude h. Como a amplitude do intervalo todo (extremos da integral) é 2–1=1, temos 2.n.h=1 e, assim, $h = \frac{1}{2.n}$. Então:

$$E_{tr} < n.\left|\frac{h^5}{90}.\left(D^4\left(\frac{1}{x}\right)\right)\right| \quad \Rightarrow \quad E_{tr} < n.\frac{\left(\frac{1}{2.n}\right)^5}{90}.\frac{24}{x^5} = \frac{24}{32 \times 90 \times n^4 \times 1^5} < 0{,}0005$$. Para limitação superior, x=1.

Assim, $\frac{1}{n^4} < 0{,}06 \quad \Rightarrow \quad n^4 > 16{,}67 \quad \Rightarrow \quad n > 2{,}02$ e como $n \in \mathbb{N}^* \quad \Rightarrow \quad n = 3$

Se n=3 então temos 6 intervalos e, portanto, 7 pontos. Como $h = \frac{1}{2.n} = \frac{1}{6}$, a tabela é a seguinte:

x	1	7/6	8/6	9/6	10/6	11/6	2
$f(x) = \frac{1}{x}$	1	0,857	0,75	0,667	0,6	0,545	0,5

Um valor aproximado de $I = \frac{1/6}{3}.\left(1 + 4(0{,}857 + 0{,}667 + 0{,}545) + 2(0{,}75 + 0{,}6) + 0.5\right) = 0{,}693$

No Matlab *Editor* a resolução é:

```
a= 1 ,  b= 2     %extremos inferior e superior de integração
Etr= 0.0005    %erro de truncamento dado no enunciado
syms  x  %x é uma variável simbólica
fx= 1/x   %função f dada no enunciado
df=diff(fx, 4)    %derivada de ordem 4 da f
m=a:0.005:b ;  %extremos de variação para achar o maior valor df
R= eval(subs(df,x,m)) ; %substituindo e calculando na derivada df
M=max(abs(R))  %M é o maior valor em modulo no intervalo de m
k=ceil((abs(b-a)^5*M/(32*90*Etr))^0.25) %function ceil para achar o maior
fprintf('O número k de vezes em que Simpson é usada é:%3.0f\n',k);
%Cálculo da integral pela fórmula de Simpson
n=2*k     % n é o número de partições do intervalo [a , b]
h=abs((b-a)/n)    %passo entre os pontos tabelados
t=a:h:b  %extremos de variação
y=eval(subs(fx,t))
s=y(1)+y(n+1) ;   %primeiro e último pontos da tabela
u=2:2:n ;    % elementos na posição par
v=3:2:n-1;  % elementos na posição impar
```

```
s=s+4*sum(y(u))+2*sum(y(v));
s= (h/3)*s     % fórmula de Simpson
fprintf('Valor aproximado da integral é: %8.5f+/-%7.5f\n',s,Etr)
fprintf('\n' )
disp('Usando a  function int  para calcular o valor exato:')
syms  x ,   %variável simbólica x
I=int(fx, a , b) %function int do Matlab que calcula a integral definida
Vexato=eval(I)
```

Em *Command Window* temos as seguintes respostas:

```
a =
     1
b =
     2
Etr =
   0.0005
fx =
1/x
df =
24/x^5
M =
    24
O número k de vezes em que Simpson é usada é:  3
n =
     6
h =
    0.1667
t =
    1.0000    1.1667    1.3333    1.5000    1.6667    1.8333    2.0000
y =
    1.0000    0.8571    0.7500    0.6667    0.6000    0.5455    0.5000
Valor aproximado da integral é:  0.69317+/-0.00050

Usando a  function int do Matlab  para calcular o valor exato:
I =
log(2)
Vexato =
    0.69315
```

5.3 Exercícios: Resolver no Matlab

(1) O Exercício 15 só com a fórmula de Simpson e o Exercício 16 só com a fórmula do Trapézio.

Resposta: Exerc.15: Custo=2910,6mil e Et < 0.076734. *Exerc.16:* Custo= 2235.6mil e Et < 0.5275.

(2) Exercício 17 pela fórmula do Trapézio.   *Resposta:* I= 0.69316

6. Equações Diferenciais

6.1 Método de Euler

Exercício 18: Resolver usando a fórmula de Euler a equação diferencial (EDO) $y' = x.y$, com precisão de quatro casas decimais (4 c.d.) de modo que $y(0) = 1$, h=0,1 e $x \in [\,0, 1\,]$.

Resolução:

A fórmula de Euler, adaptada para esse Exercício, é $y(x + 0{,}1) = y(x) + 0{,}1.x.y(x)$.

Atribuindo-se valores para x, nessa fórmula e com y(0)=1, obtemos a seguinte tabela de valores para y(x) :

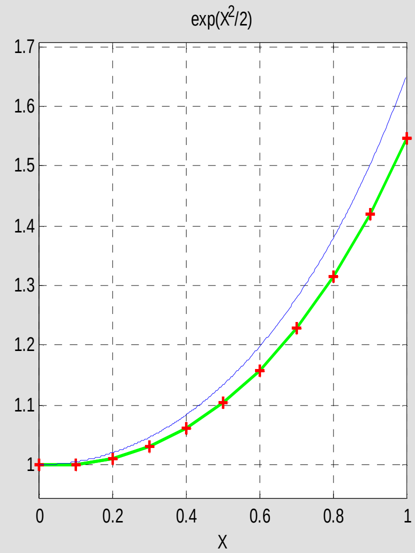
x	y(x)
0	1,0000
0,1	1,0000
0,2	1,0100
0,3	1,0302
0,4	1,0611
0,5	1,1035
0,6	1,1587
0,7	1,2282
0,8	1,3142
0,9	1,4193
1,0	1,5470

No Matlab, digitando no *Editor:*

```
h=0.1 ; a=0  ;  b=1  ;  %passo h e extremos do intervalo
x=a:h:b  ;
n=(b-a)/h ; y=1;  %número n de pontos tabelados e valor inicial y(0)=1
f=@(x,y) x*y  %EDO dada por y'=f(x,y)=x.y
for i=1:1:n
   K1(i)= f(x(i),y(i)) ;   %fórmula de Euler
   y(i+1)=y(i)+h*K1(i) ;  %fórmula de Euler
end
disp('A tabela-solução é dada por:')
disp( '     x       y       K1')
disp([x' y' [K1(1:n)';0]])   %tabela-solução da EDO
plot(x,y,'g',x,y,'r+','linewidth',1.5) ; %gráfico
hold on  %aguardar o próximo gráfico no mesmo plano cartesiano
disp('Usando a Função dsolve do MatLab')
syms X Y ;  %X e Y são variáveis simbólicas
solY=dsolve('DY=X*Y', 'Y(0)=1 ', 'X')  %function dsolve do Matlab
disp('Se x=1 então y(1) vale:')
y1=vpa(subs(solY,X,1),5)
m=0:1 ;
ezplot(solY , m)  , %gráfico da solução exata com a function ezplot
grid
hold off %”soltar” gráfico
```

As respostas, em *Command Window:*

```
f =
    @(x,y)x*y
A tabela-solução é dada por:
      x           y        K1
      0        1.0000      0
    0.1000     1.0000    0.1000
    0.2000     1.0100    0.2020
    0.3000     1.0302    0.3091
    0.4000     1.0611    0.4244
    0.5000     1.1036    0.5518
    0.6000     1.1587    0.6952
    0.7000     1.2283    0.8598
    0.8000     1.3142    1.0514
    0.9000     1.4194    1.2774
    1.0000     1.5471       0

Usando a Função dsolve do MatLab
solY =
exp(X^2/2)
Se x=1 então y(1) vale:
y1 =
1.6487
```



6.2 Método de Euler-Modificado

Exercício 19*: Resolver o Exercício anterior, pelo método de Euler-Modificado. Comparar com os resultados obtidos pelo método de Euler e, também com a solução exata.

Resolução:

As fórmulas de Euler-Modificado, adaptadas para este exercício, são: $\left\{\begin{array}{l} K_1 = x.y \\ K_2 = (x + 0{,}05).(y + 0{,}05.K_1) \\ y(x + 0{,}1) = y(x) + 0{,}1.K_2 \end{array}\right.$.

Atribuindo-se valores para x, calculam-se K_1 , K_2 e a solução y(x) que são dados na tabela:

x	y(x)	K_1	K_2
0	1	0	0,05
0,1	1,005	0,1005	0,151504
0,2	1,02015	0,20403	0,257588
0,3	1,045909	0,313773	0,371559
0,4	1,083065	0,433226	0,497127
0,5	1,132778	0,566389	0,638603
0,6	1,196638	0,717983	0,801149
0,7	1,276753	0,893727	0,99108
0,8	1,375861	1,100689	1,216261
0,9	1,497487	1,347738	1,48663
1,0	1,64615	—	—

Para x=1 pelo método de Euler-Modificado temos y(1)=1,6462 e esse resultado é bem melhor que o anterior do método de Euler, levando-se em conta que a solução exata é y(1)=1,6487 (ver Exercício 18 no Matlab)

No Matlab, a resolução no *Editor* é:

```
h=0.1 ; a=0  ;  b=1  ;  %passo h e extremos a e b do intervalo para x
x=a:h:b  ;
n=(b-a)/h ; y=1 ;  %número n de pontos tabelados e valor inicial y(0)=1
f=@(x,y) x*y    %EDO dada por y'=f(x,y)=x.y
for i=1:1:n
   K1(i)= f(x(i),y(i)) ;   %fórmula de Euler
   K2(i)= f(x(i)+h/2,y(i)+h/2) ;  %fórmula de Euler
   y(i+1)=y(i)+h*K2(i);  %fórmula de Euler
end
disp('A tabela-solução é dada por:')
disp( '       x        y         K1         K2')
disp([x' y' [K1(1:n)';0]  [K2(1:n)';0]])  %tabela-solução de y'=f(x,y)=xy
plot(x,y,x,y,'r+') , grid  %gráfico e grade
```

Em *Command Window* temos as respostas:

```
f =
    @(x,y)x*y
A tabela-solução é dada por:
      x          y         K1         K2
      0       1.0000      0       0.0525
    0.1000    1.0053    0.1005    0.1583
    0.2000    1.0211    0.2042    0.2678
    0.3000    1.0479    0.3144    0.3842
    0.4000    1.0863    0.4345    0.5113
    0.5000    1.1374    0.5687    0.6531
    0.6000    1.2027    0.7216    0.8143
    0.7000    1.2841    0.8989    1.0006
    0.8000    1.3842    1.1074    1.2191
    0.9000    1.5061    1.3555    1.4783
    1.0000    1.6539       0          0
```

6.3 Método de Runge-Kutta de 4ª ordem

Exercício 20: Resolver o Exercício 19, com h=0,1 e $x \in [0, 1]$, pelo método de Runge-Kutta de 4ª ordem.

Resolução:

As fórmulas de Runge-Kutta de 4ª ordem, adaptadas para este exercício, são as seguintes:

$$K_1 = x.y$$
$$K_2 = (x + 0{,}05).(y + 0{,}05.K_1)$$
$$K_3 = (x + 0{,}05).(y + 0{,}05.K_2)$$
$$K_4 = (x + 0{,}1).(y + 0{,}1.K_3)$$
$$y(x + 0{,}1) = y(x) + \frac{0{,}1}{6}.(K_1 + 2.K_2 + 2.K_3 + K_4)$$

Atribuindo-se valores, a solução y(x) é dada na tabela:

x	**y**	**K_1**	**K_2**	**K_3**	**K_4**
0	1.0000	0	0.0500	0.0501	0.1005
0.1000	1.0050	0.1005	0.1515	0.1519	0.2040
0.2000	1.0202	0.2040	0.2576	0.2583	0.3138
0.3000	1.0460	0.3138	0.3716	0.3726	0.4333
0.4000	1.0833	0.4333	0.4972	0.4987	0.5666
0.5000	1.1331	0.5666	0.6388	0.6408	0.7183
0.6000	1.1972	0.7183	0.8015	0.8042	0.8943
0.7000	1.2776	0.8943	0.9918	0.9954	1.1017
0.8000	1.3771	1.1017	1.2174	1.2223	1.3494
0.9000	1.4993	1.3494	1.4884	1.4950	1.6488
1.0000	1.6487	0	0	0	0

A resolução no *Editor* do Matlab:

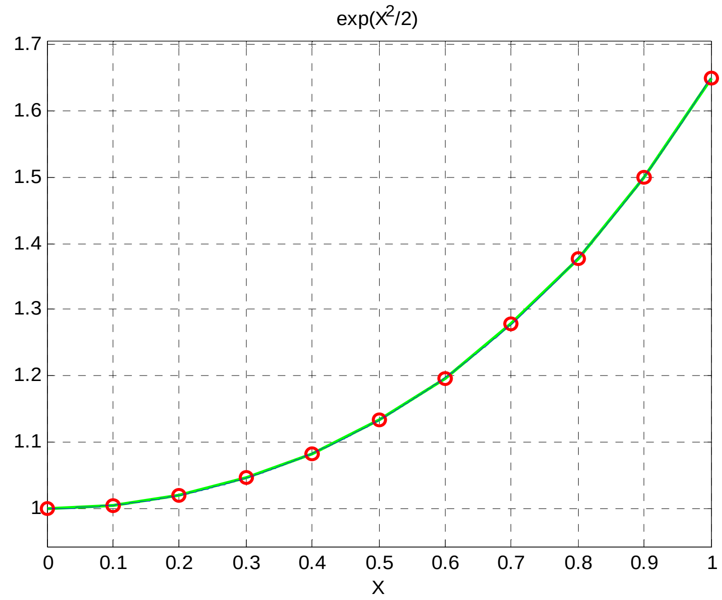
```
format compact
h=0.1 ; x=0:h:1 ; y=1 ;  % passo h=0,1 , x em [0 , 1] e y(0)=1 valor inicial
n=(1-0)/h ;  %  número n de partições do intervalo
F=@(x,y)x*y  %EDO dada no enunciado

for j=1:n
   K1(j)=F(x(j) , y(j));   %fórmulas de Runge-Kutta 4ªordem
   K2(j)= F(x(j)+h/2 , y(j)+(h/2)*K1(j)) ;
   K3(j)= F(x(j)+h/2 , y(j)+(h/2)*K2(j)) ;
   K4(j)= F(x(j)+h , y(j)+h*K3(j)) ;
   y(j+1)=y(j)+(h/6)*(K1(j)+2*(K2(j)+K3(j))+K4(j));
end

disp('A solução é:') ,
disp( '   x     y    K1   K2   K3  K4')
disp([x' ,y',[K1(1:n)';0] ,[K2(1:n)';0],[K3(1:n)';0] ,[K4(1:n)';0]])

disp('Usando a Função dsolve do MatLab')
syms X Y ;  %variáveis simbólicas
solY=dsolve('DY=X*Y', 'Y(0)=1 ', 'X')
```

```
%Gráfico
m=0:1  ;
plot(x,y,'g',x,y,'ro','linewidth',1.5) ;
hold on
ezplot(solY , m)  ,  grid
hold off
disp('Comparando a solução exata com a solução aproximada temos:')
format long

z=[0:0.1:1]';
SE=subs(solY,z);
disp( ' x     aproximada  exata')
disp(eval([x' , y',SE]))
```

As respostas em *Command Window:*

```
F =
    @(x,y)x*y
A solução é:
      x         y         K1         K2        K3         K4
      0       1.0000      0       0.0500    0.0501    0.1005
    0.1000    1.0050    0.1005    0.1515    0.1519    0.2040
    0.2000    1.0202    0.2040    0.2576    0.2583    0.3138
    0.3000    1.0460    0.3138    0.3716    0.3726    0.4333
    0.4000    1.0833    0.4333    0.4972    0.4987    0.5666
    0.5000    1.1331    0.5666    0.6388    0.6408    0.7183
    0.6000    1.1972    0.7183    0.8015    0.8042    0.8943
    0.7000    1.2776    0.8943    0.9918    0.9954    1.1017
    0.8000    1.3771    1.1017    1.2174    1.2223    1.3494
    0.9000    1.4993    1.3494    1.4884    1.4950    1.6488
    1.0000    1.6487         0         0         0         0

Usando a Função dsolve do MatLab
solY =
exp(X^2/2)

Comparando a solução exata com a solução aproximada temos:
         x                aproximada            exata
         0             1.000000000000000   1.000000000000000
   0.100000000000000   1.005012520833333   1.005012520859401
   0.200000000000000   1.020201339758369   1.020201340026756
   0.300000000000000   1.046027858885970   1.046027859908717
   0.400000000000000   1.083287064820495   1.083287067674959
   0.500000000000000   1.133148446117537   1.133148453066826
   0.600000000000000   1.197217347412631   1.197217363121810
   0.700000000000000   1.277621279469110   1.277621313204887
   0.800000000000000   1.377127694901942   1.377127764335957
   0.900000000000000   1.499302362448324   1.499302500056767
   1.000000000000000   1.648721007053397   1.648721270700128
```

O Gráfico:



6.4 Exercícios: Resolver a equação diferencial EDO pelos métodos de Euler, Euler-modificado, Runge-Kutta de 4ª ordem e também pela function dsolve do MatLab, nos seguintes casos:

(1) EDO : $y' = x$, com y(0)=1 , $x \in [0\ ,\ 2]$ e h=0,5

(2) EDO : $y' = y$, com y(0)=1 , $x \in [0\ ,\ 5]$ e h=1

(3) EDO : $y' = x^2 y$, com y(0)=2 , $x \in [0\ ,\ 1]$ e h=0,2

(4) EDO : $y' = \dfrac{x^2}{y}$, com y(0)=2 , $x \in [0\ ,\ 1]$ e h=0,2