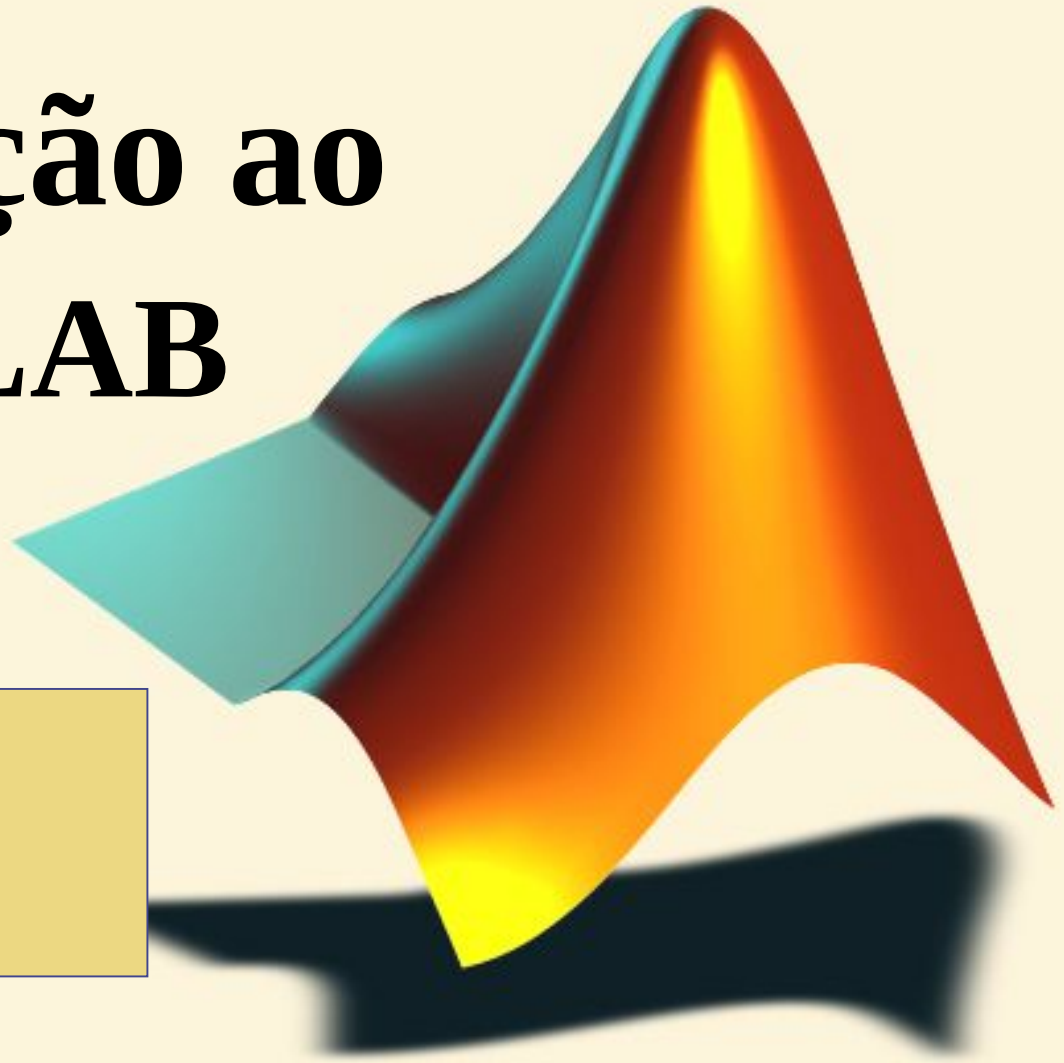
Introdução ao MATLAB



Prof. Diomar Cesar Lobão

Prof. José Flávio Feiteira

UFF – Volta Redonda, out 2009

Índice

1. Parte I

- O que é o Matlab?
- Componentes do MATLAB
- MATLAB Desktop
- Matrizes
  - Arrays Numéricos
  - Arrays de Caracteres “String”
- Matemática Elementar
  - Operadores Lógicos
  - Funções Matemáticas
  - Interpolação de Polinômios
- Importando e Exportando Dados

Índice

Continua

- Introdução aos Gráficos
  - Plotando gráficos 2D
  - Subplots
  - Plotando gráficos 3D
  - Gráficos especiais
- Editando M-files

2. Parte II

- Script e arquivos de Funções
- O Básico de um M-file
- Comandos de controle de Fluxo
- Programando um M-file

Índice

Continua

- Tipos de Dados
  - Arrays Multidimensional
  - Estruturas
  - Células de Arrays “Cell Arrays”
- Funções numéricas não-lineares
- Equações Diferenciais Ordinárias
- Handle Graphics
- Objetos Gráficos
- Interface Gráfica do Usuário (GUI)

O que é o MATLAB?

- Software de alto desempenho
  - ***Computação***
  - ***Visualização***
  - ***Ambiente "Easy-to-use"***
- Linguagem de alto nível
  - ***Tipos de Dados***
  - ***Funções***
  - ***Controle de fluxo***
  - ***Input/output***
  - ***Gráficos***
  - ***Capacidade de programação Orientada a Objetos***

Componentes do MATLAB

- Ambiente de Desenvolvimento
- Linguagem de Programação
- Gráficos
- Toolboxes
- Aplicações de Interfaciamento de Programas

Toolboxes

- Coleções de funções p/ resolver problemas em variadas aplicações.
  - FFP – (Fast Fourier Processing) Toolbox
  - Imagem - Toolbox
  - Wavelet - Toolbox
  - Redes Neurais - Toolbox
  - Lógica Fuzzy - Toolbox
  - Controle - Toolbox
  - Comunicação – Toolbox
  - Signal Processing – Toolbox

Ferramentas MATLAB no Desktop

- Janela de Comando
- Janela de História de comandos
- Help Browser
- Janela do Workspace
- Editor – M - files (script)

Cálculos na linha de Comando

MATLAB como uma calculadora

```
» -5/(4.8+5.32)^2
ans =
   -0.0488
» (3+4i)*(3-4i)
ans =
    25
» cos(pi/2)
ans =
  6.1230e-017
» exp(acos(0.3))
ans =
    3.5470
```

Atribuição de Variáveis

```
» a = 2;
» b = 5;
» a^b
ans =
    32
» x = 5/2*pi;
» y = sin(x)
y =
     1
» z = asin(y)
z =
    1.5708
```

Ponto e vírgula inibe output

Resultado atribuído a "ans" se nome não é especificado

() parenteses para input de função

Nota sobre Workspace:
Números são armazenados em "double-precision" formato

Funções Gerais

- whos: Lista as variáveis em uso
- clear all:Limpa as variáveis e funções da memória
- Close all: Fecha todas as figures
- cd: muda de diretório
- dir: Lista os arquivos do diretório
- echo: Ecoa os comando de um M-file
- format: Estabelece o formato numérico

Conseguindo Ajuda

- *Comando help* (**>>help**)
- *Comando lookfor* (**>>lookfor**)
- Help Browser (**>>doc**)
- *Comando helpwin* (**>>helpwin**)
- Documentos sobre Matlab
  - "Matlabroot\help\pdf_doc\"
- Link para The MathWorks

Matrizes

- Entrando e Gerando Matrizes
- Subscripts (Vetores, matrizes)
- Expansão Escalar
- Concatenação
- Apagando Linhas e Colunas
- Extração de Arrays
- Multiplicação de Matrizes

Entrando Arrays Numéricos

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Separador de Linha (;)

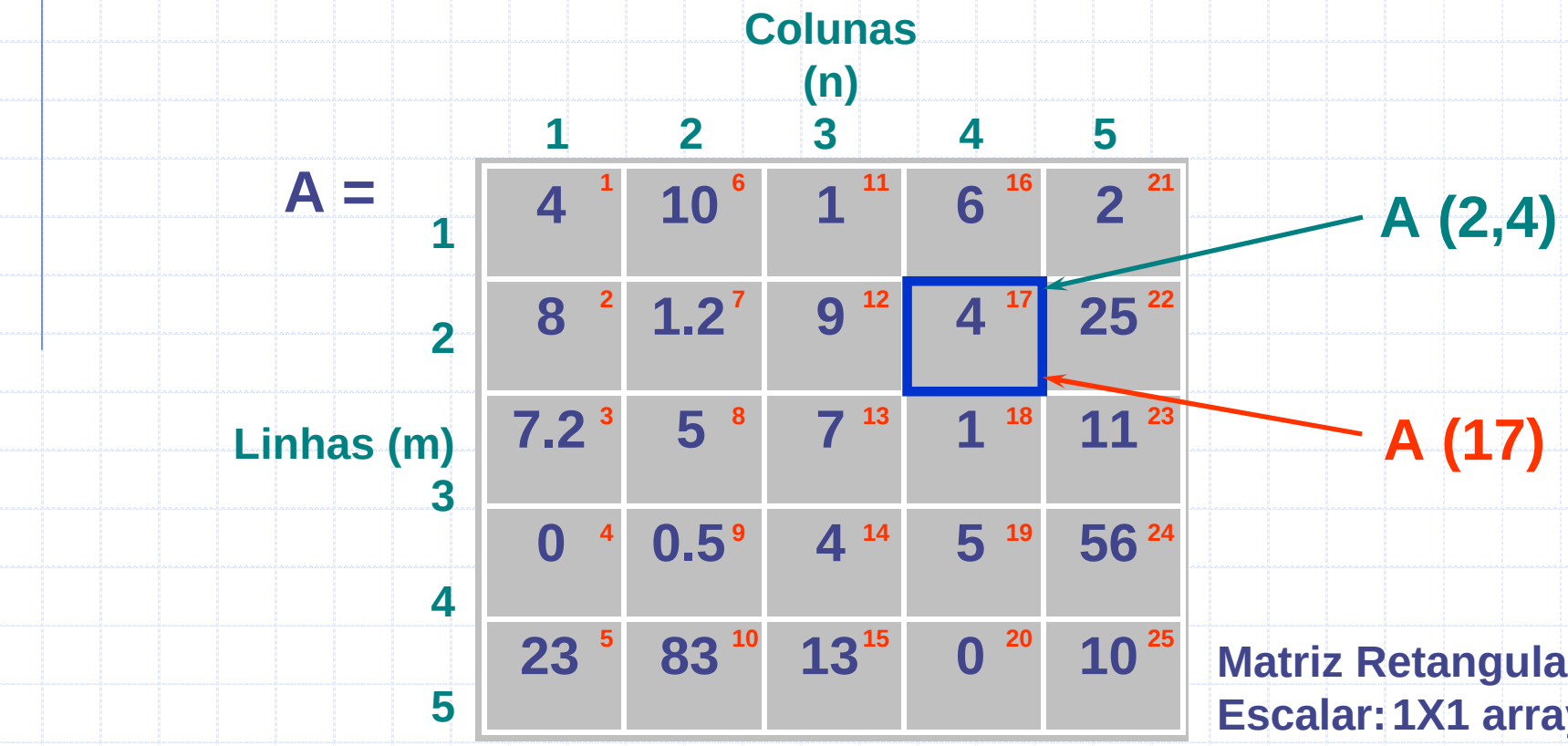
Coluna / virgula (,)

```
» a=[1 2;3 4]
a =
     1     2
     3     4
» b=[-2.8, sqrt(-7), (3+5+6)*3/4]
b =
   -2.8000   0 + 2.6458i   10.5000
» b(2,5) = 23
b =
   -2.8000  0 + 2.6458i  10.5000   0         0
         0             0         0   0  23.0000
```

Uso das chaves [ ]

- Qualquer expressão em MATLAB pode ser entrada como elemento de matriz
- Matrizes devem ser retangular.

Matrizes em MATLAB



Matriz Retangular:
Escalar: 1X1 array
Vetor: mX1 array
1Xn array
Matriz: m-by-n array

Entrando Arrays Numérico

Expansão escalar

Criando sequências:
operador (:)

Função p/ criar matrizes aleatória.

```
» w=[1 2;3 4] + 5
w =
     6     7
     8     9
» x = 1:5
x =
     1     2     3     4     5
» y = 2:-0.5:0
y =
  2.0000   1.5000   1.0000   0.5000   0
» z = rand(2,4)
z =
  0.9501   0.6068   0.8913   0.4565
  0.2311   0.4860   0.7621   0.0185
```

Concatenação Arrays Numéricos

Uso de [ ] p/ combinar arrays como "elementos" de matriz

<u>Separador Linha:</u>
Ponto - virgula (;)

<u>Separador Coluna:</u>
espaço / virgula (,)

```
» a=[1 2;3 4]
a =
     1     2
     3     4
» cat_a=[a, 2*a; 3*a, 4*a; 5*a, 6*a]
cat_a =
     1     2     2     4
     3     4     6     8
     3     6     4     8
     9    12    12    16
     5    10     6    12
    15    20    18    24
```

Uso de chaves [ ]

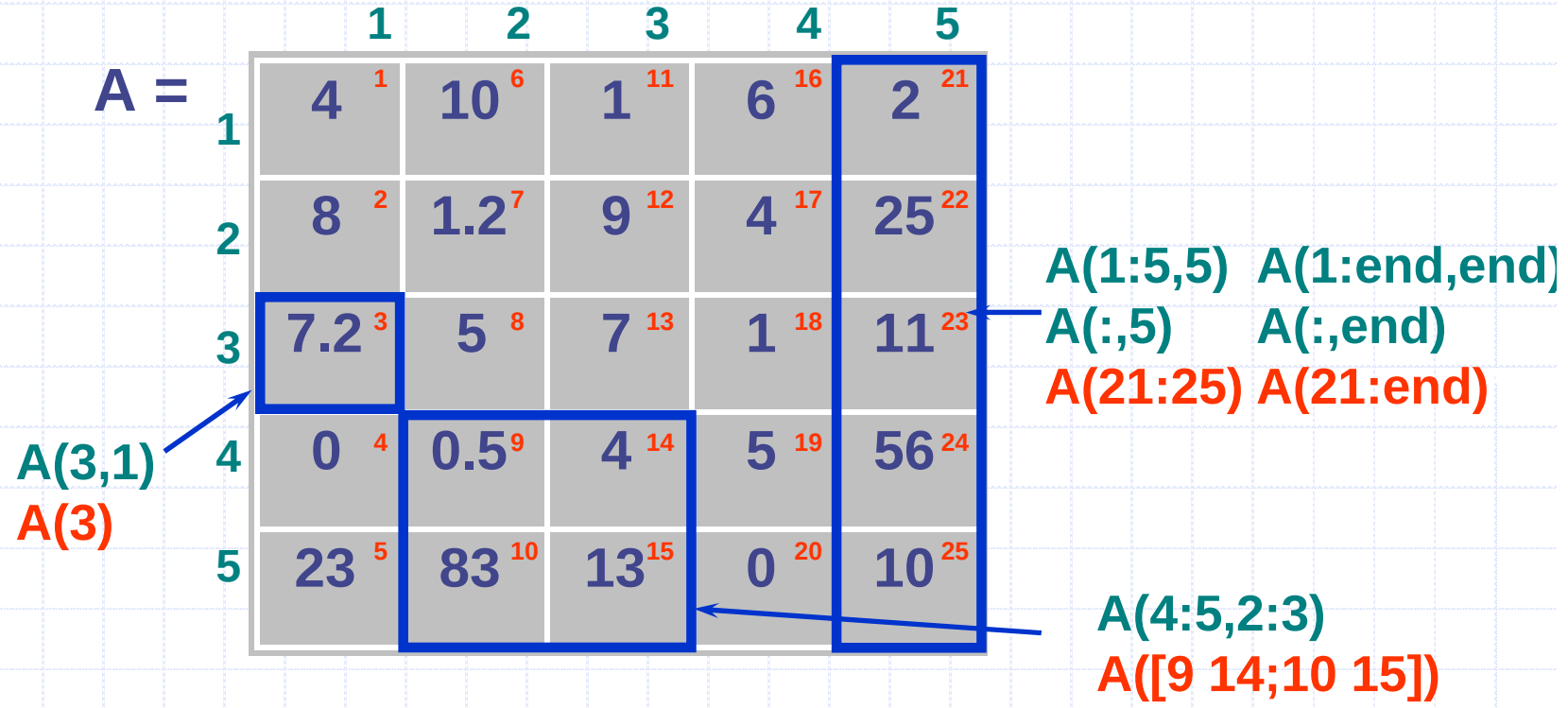
4*a

Nota: A matriz resultante deve ser retangular.

Apagando Linhas e Colunas

```
» A=[1 5 9;4 3 2.5; 0.1 10 3i+1]
A =
   1.0000            5.0000            9.0000
   4.0000            3.0000            2.5000
   0.1000           10.0000            1.0000+3.0000i
» A(:,2)=[]
A =
   1.0000            9.0000
   4.0000            2.5000
   0.1000            1.0000 + 3.0000i
» A(2,2)=[]
???  Matriz indexada vazia não é permitida!!!.
```

Array Subscript (I,J)



Multiplicação de Matrizes

```
» a = [1 2 3 4; 5 6 7 8];
» b = ones(4,3);
» c = a*b
c =
    10    10    10
    26    26    26
```

[2x4]

[4x3]

[2x4]*[4x3] → [2x3]

a(2° Linha).b(3° coluna)

Multiplicação de elemento a elemento de um array

```
» a = [1 2 3 4; 5 6 7 8]; -> (2X4)
» b = [1:4; 1:4]; -> (2X4)
» c = a.*b -> Operador (.)
c =
     1     4     9    16
     5    12    21    32
```

c(2,4) = a(2,4)*b(2,4)

Funções de Manipulação de Matrizes

- *zeros*: Cria um array de “0”
- *ones*: Cria um array de “1”
- *eye*: Cria matriz identidade
- *rand*: Cria uma distribuição uniforme de números randômicos
- *diag*: Cria matrizes Diagonal e diagonal de uma matriz
- *size*: Retorna as dimensão de um array
- *fliplr*: Gira uma matriz da esquerda p/ direita
- *flipud*: Gira uma matriz cima e p/ baixa
- *repmat*: Cria e Re-mapeia uma matriz

Funções de Manipulação de Matrizes

- *transpose (')*: Matriz Transposta
- *rot90*: rotaciona a matriz de 90 graus
- *tril*: Parte inferior da matriz triangular
- *triu*: Parte superior de uma matriz triangular
- *cross*: Produto vetorial
- *dot*: Produto escalar
- *det*: Determinante de uma matriz
- *inv*: Inversa de uma Matriz
- *eig*: Autovalores e Autovetores
- *rank*: Rank de uma matriz (Fornece uma estimativa do número de linhas ou colunas linearmente independentes de uma matriz).

Arrays de Caracteres (Strings)

- Cria usando (')

```
» str  = 'Oh facil,'
str =
Oh facil,
» str2 = 'MATLAB e bom!'
str2 =
MATLAB e bom!
```

- Cada caracteres é um elemento separado
  - (16 bits por caracter)

- Indexação similar ao array numérico

Concatenação de Caractere

Usando operador [ ] :
Cada linha de ter mesmo comprimento

Separador linha: (;)

Separador coluna:
espaço / virgula (,)

```
» str ='Hi there,';
» str1='Everyone!';
» new_str=[str, ' ', str1]
new_str =
Hi there, Everyone!
» str2 = 'Isn''t MATLAB great?';
» new_str2=[new_str; str2]
new_str2 =
Hi there, Everyone!
Isn't MATLAB great?
```

1x9 vetor

1x19 vetor

2x19 matriz

Para strings de diferente tamanho:

- STRVCAT
- Char

(ambas funcionam igualmente)

```
» new_str3 = strvcat(str, str2)
new_str3 =
Hi there,
Isn't MATLAB great?
```

2x19 matriz

Trabalhando com Arrays de String

- Comparação de String
  - strcmp: compara todo o strings
  - strncmp: compara os primeiros ‘N’ caracteres
  - findstr: acha substring dentro um string maior
- Comparação entre arrays numérico & string:
  - num2str: converte array de numérico p/ string
  - str2num: converte array de string p/ numérico

Matemática Elementar

- Operadores Lógicos
- Funções Matemáticas
- Polinômios e Interpolação

Operadores Lógicos

= =	equal to
>	Maior que
<	Menor que
>=	Maior ou Igual
<=	Menor ou igual
~	não(negativa)
&	e
|	ou
isfinite(), etc. . . .	
all(), any()	
find	

```
» Mass = [-2 10 NaN 30 -11 Inf 31];
» each_pos = Mass>=0
each_pos =
     0     1     0     1     0     1     1
» all_pos = all(Mass>=0)
all_pos =
     0
» all_pos = any(Mass>=0)
all_pos =
     1
» pos_fin = (Mass>=0)&(isfinite(Mass))
pos_fin =
     0     1     0     1     0     0     1
```

Nota:

- => 1 => TRUE
- => 0 => FALSE

Funções Matemáticas Elementares

- *abs*, *sign*: Valor Absoluto e Função Sinal
- *sin*, *cos*, *asin*, *acos*…: Trigonométricas
- *exp*, *log*, *log10*: Exponential, Logaritmo Natural e Comum (base 10)
- *ceil*, *floor*: Aproxima na direção do infinito
- *fix*: Aproxima na direção de zero

Funções Matemáticas Elementares

- *round*: Aproxima na direção do próximo inteiro
- *gcd*: Maior divisor comum
- *lcm*: Mínimo múltiplo comum
- *sqrt*: Raíz quadrada
- *real*, *imag*: Real e Imaginária
- *rem*: Resto depois da divisão

Funções Matemáticas Elementares

- *max*, *min*: Máximo e Mínimo de arrays
- *mean*, *median*: Média e Mediana de arrays
- *std*, *var*: Desvio padrão e Variância
- *sort*: Ordenação elementos em ordem ascendente
- *sum*, *prod*: Somatório & Produto de Elementos
- *trapz*: Integração numérica Trapezoidal
- *cumsum*, *cumprod*: Soma acumulativa e produto
- *diff*, *gradient*: Diferenças e Gradiente numérico

Polinômios e Interpolação

- Polinômios
  - Representação
  - Raízes (**>> roots**)
  - Cálculo (**>> polyval**)
  - Derivadas (**>> polyder**)
  - Ajuste de Curva (**>> polyfit**)
  - Expansão em Funções Parciais (**residue**)
- Interpolação
  - 1D (**interp1**)
  - 2D (**interp2**)

Exemplo

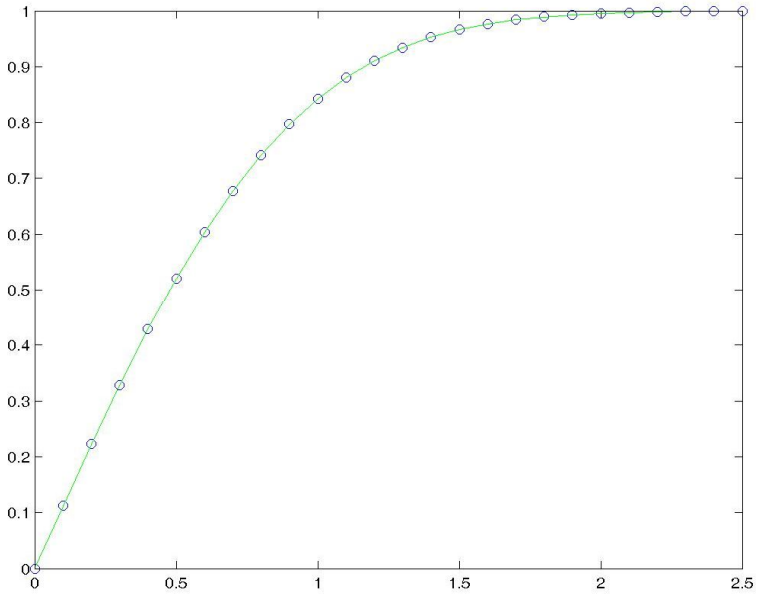
```
polysam=[1 0 0 8];
roots(polysam)
ans =
  -2.0000
   1.0000 + 1.7321i
   1.0000 - 1.7321i
Polyval(polysam,[0 1 2.5 4 6.5])
ans =
    8.0000    9.0000   23.6250   72.0000  282.6250
polyder(polysam)
ans =
     3     0     0
[r p k]=residue(polysam,[1 2 1])
r =  3     7
p = -1    -1
k =  1    -2
```

[R,P,K] = RESIDUE(B,A) finds the residues, poles and direct term of a partial fraction expansion of the ratio of two polynomials B(s)/A(s). If there are no multiple roots,

$$\frac{B(s)}{A(s)} = \frac{R(1)}{s - P(1)} + \frac{R(2)}{s - P(2)} + \ldots + \frac{R(n)}{s - P(n)} + K(s)$$

Exemplo

```
x = [0: 0.1: 2.5];
y = erf(x);
p = polyfit(x,y,6)
yy = polyval(p,x);
 %
figure(1);
plot(x,y,'bo'), hold on;
plot(x,yy,'g-');
p =
  0.0084   -0.0983    0.4217
 -0.7435   0.1471    1.1064
  0.0004
```




```
interp1(x,y,[0.45 0.95 2.2 3.0])
ans =
    0.4744    0.8198    0.9981       NaN
```

Não existe valor de x=3.0!

Importando e Exportando Dados

- Usando comando "*Save*" e "*Load* "

```
save fname
save fname x y z
save fname -ascii
save fname -mat
```

```
load fname
load fname x y z
load fname -ascii
load fname -mat
```

Input/Output p/ arquivo Texto

- Ler dado formatado, reusando o formato *string N* vezes.

```
»[A1…An]=textread(filename,format,N)
```

- Importa e Exporta dado **Numérico** com arquivos delimitados ASCII

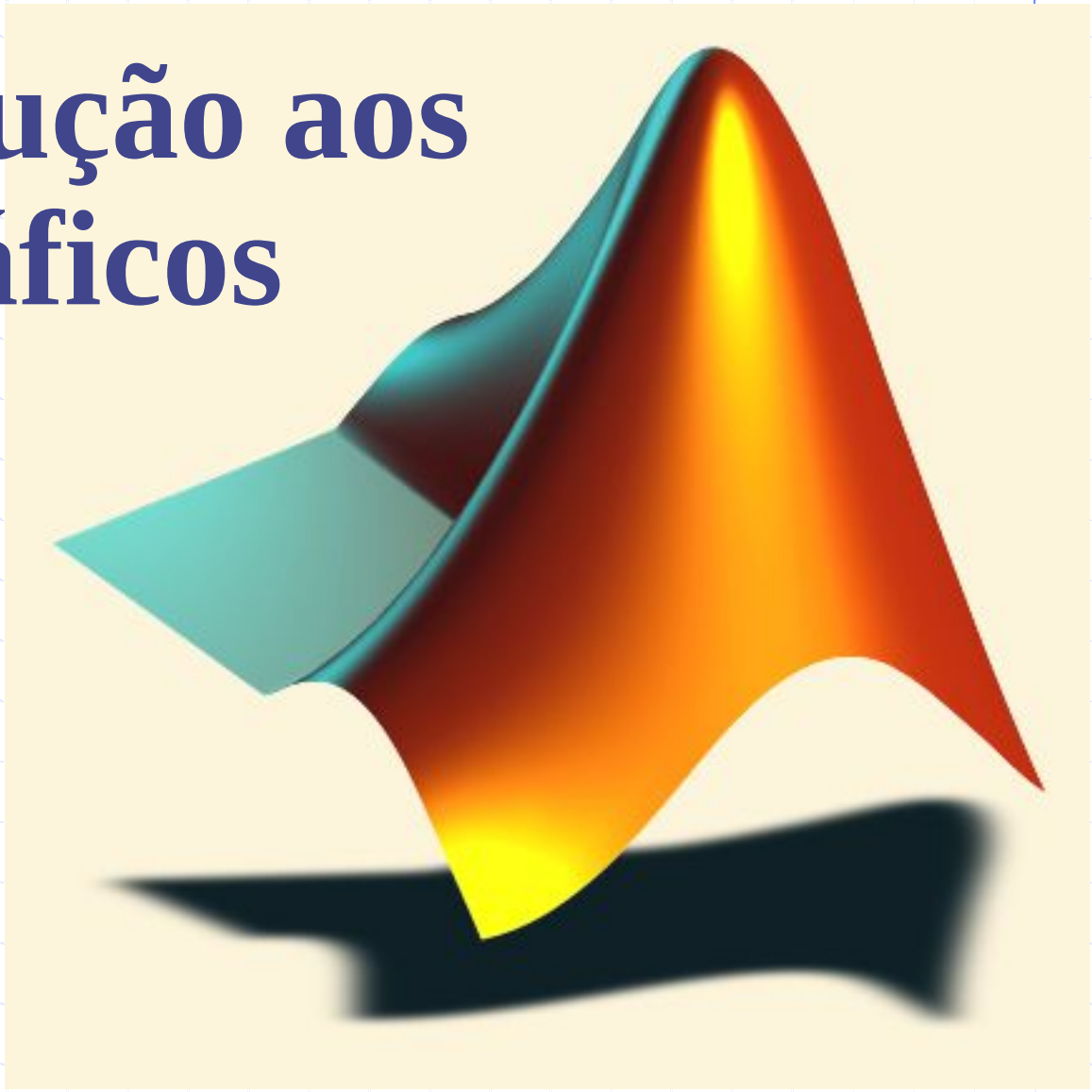
```
» M = dlmread(filename,delimiter,range)
```

Input/Output arquivo Binário

- ***fopen***: Abre um arquivo p/ input/output
- ***fclose***: Fecha um ou mais arquivos abertos
- ***fread***: Ler dado binário de um arquivo
- ***fwrite***: Escreva dado binário em arquivo
- ***fseek***: Manipula posição num arquivo

```
» fid= fopen('mydata.bin', 'wb');
» fwrite (fid,eye(5) , 'int32');
» fclose (fid);
» fid= fopen('mydata.bin', 'rb');
» M= fread(fid, [5 5], 'int32')
» fclose (fid);
```

Introdução aos Gráficos



Gráficos

- Plotagem Básica

  plot, title, xlabel, grid, legend, hold, axis

- Editando Plot

  Editar Propriedades

- Plotagem usando: *Malha e Superfície*

  meshgrid, mesh, surf, colorbar, patch, hidden

- Handle

Plotagem 2-D

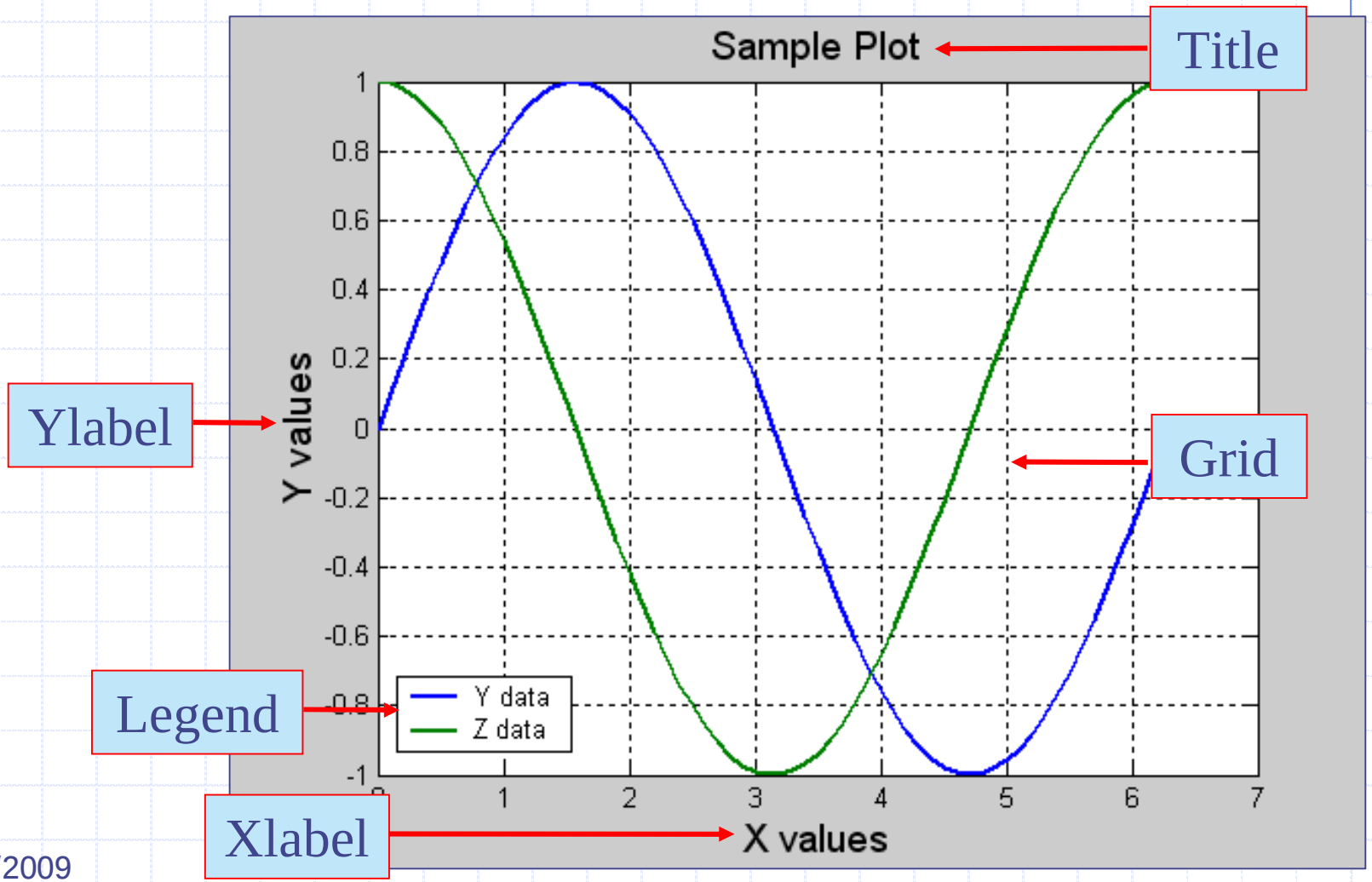
Sintaxe:

```
plot(x1, y1, 'clm1', x2, y2, 'clm2', ...)
```

Exemplo:

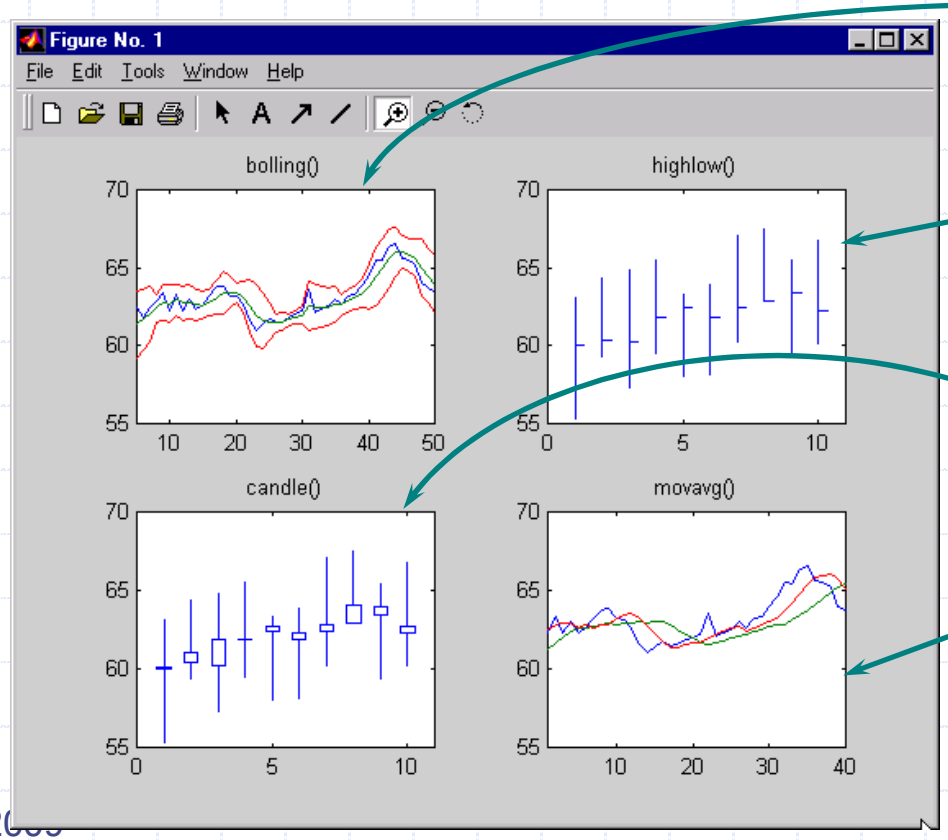
```
x=[0:0.1:2*pi];
y=sin(x);
z=cos(x);
plot(x,y,x,z,'linewidth',2)
title('Sample Plot','fontsize',14);
xlabel('X values','fontsize',14);
ylabel('Y values','fontsize',14);
legend('Y data','Z data')
grid on
```

Exemplo (Sample) de Plot



Subplots: Mais de um Gráfico por Janela

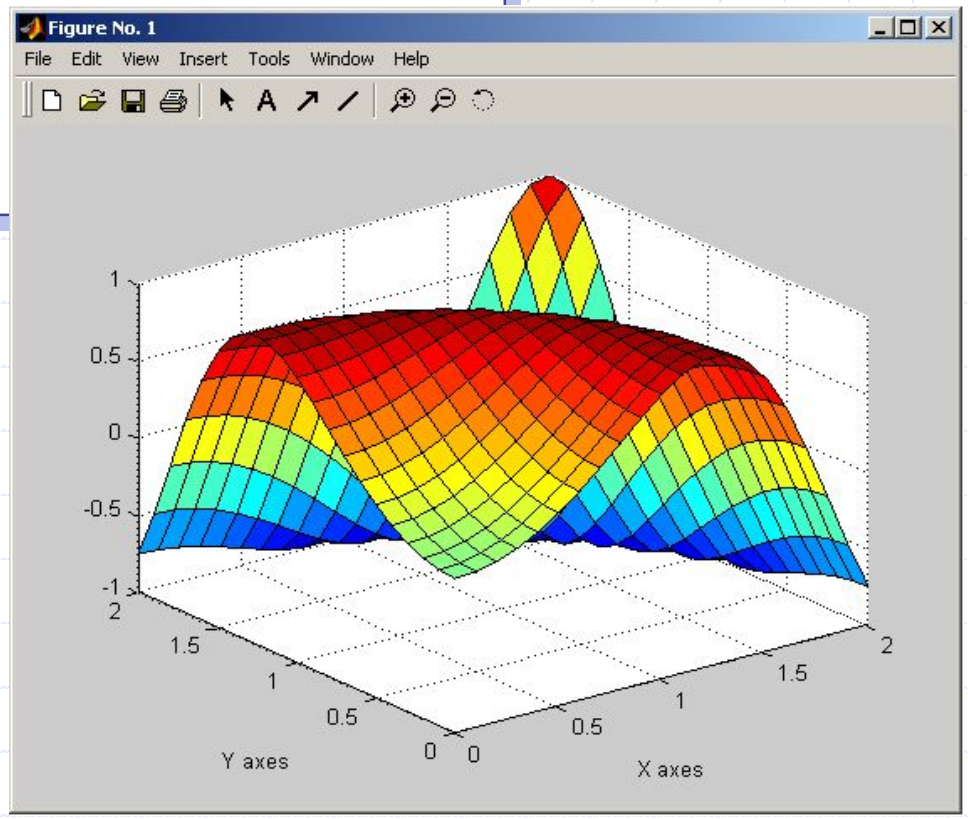
Sintaxe: `subplot(rows,cols,index)`




```
»subplot(2,2,1);
» …

»subplot(2,2,2)
» ...

»subplot(2,2,3)
» ...

»subplot(2,2,4)
» ...
```

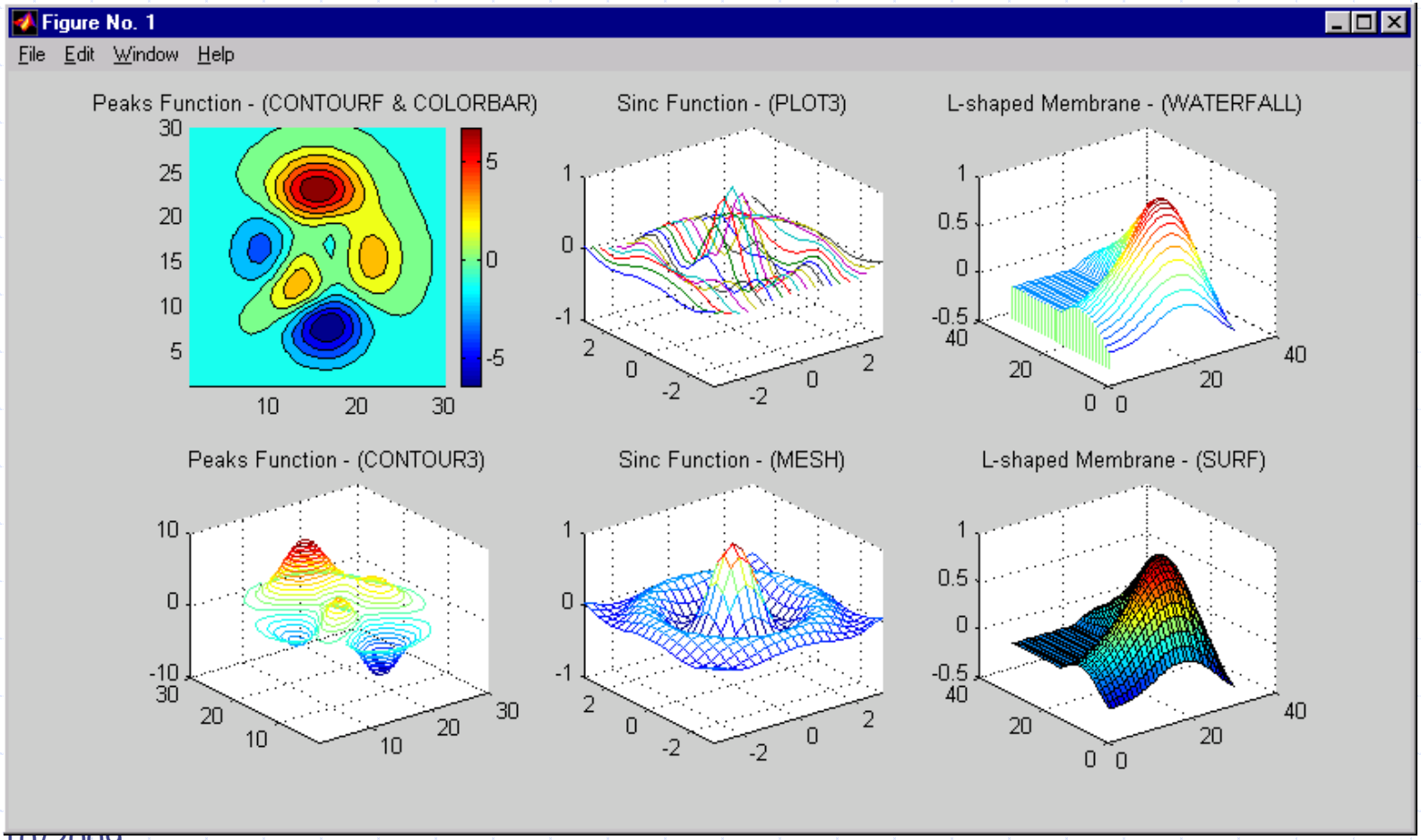
Exemplo: Plotando Superfície

```
x = 0:0.1:2;
y = 0:0.1:2;
[xx, yy] = meshgrid(x,y);
zz=sin(xx.^2+yy.^2);
surf(xx,yy,zz)
xlabel('X axes')
ylabel('Y axes')
```



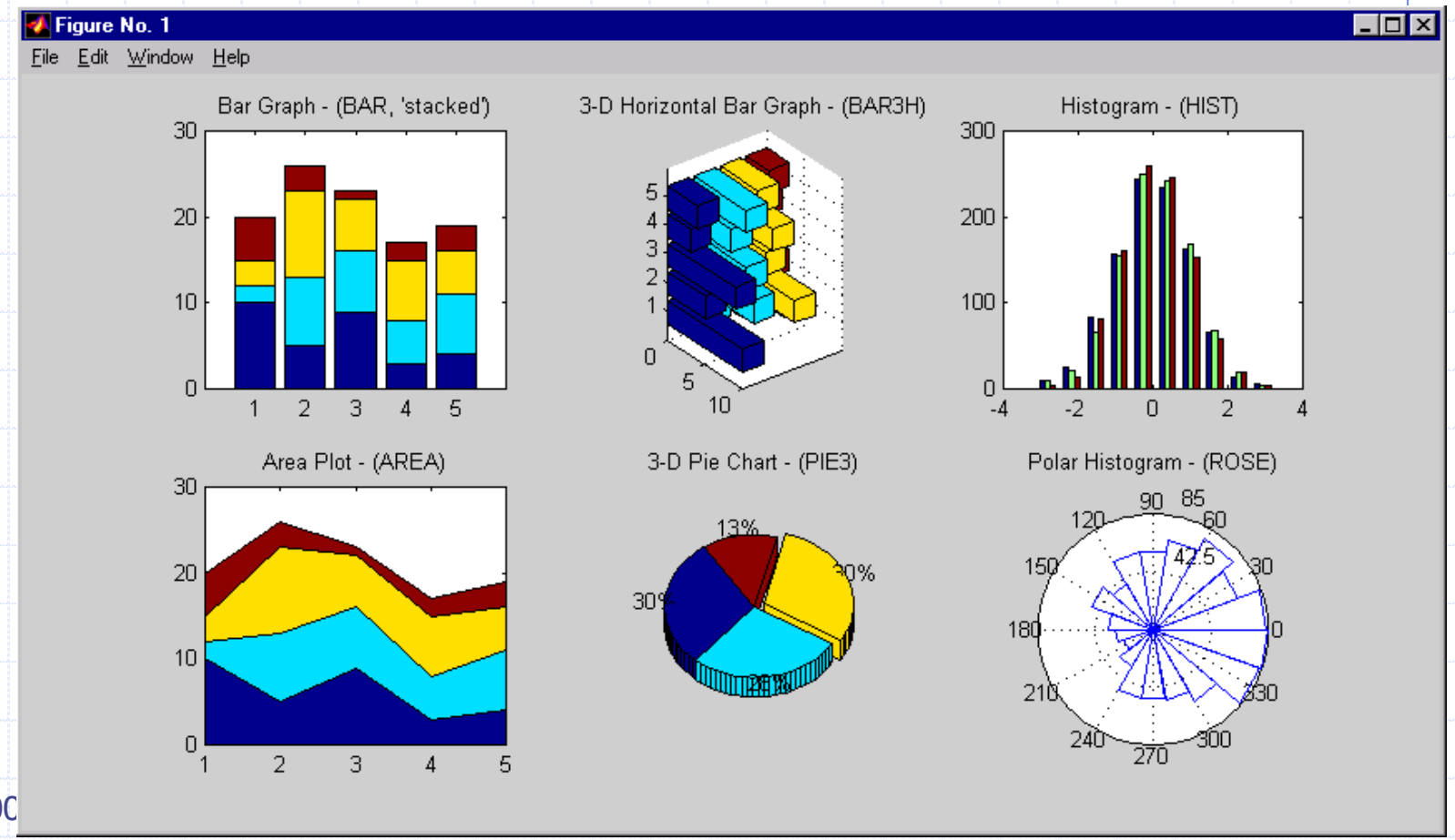
Plotando Superfícies 3-D

contourf-colorbar-plot3-waterfall-contour3-mesh-surf



Rotinas Especiais de Plotagem

bar-bar3h-hist-area-pie3-rose



Editando M-Files

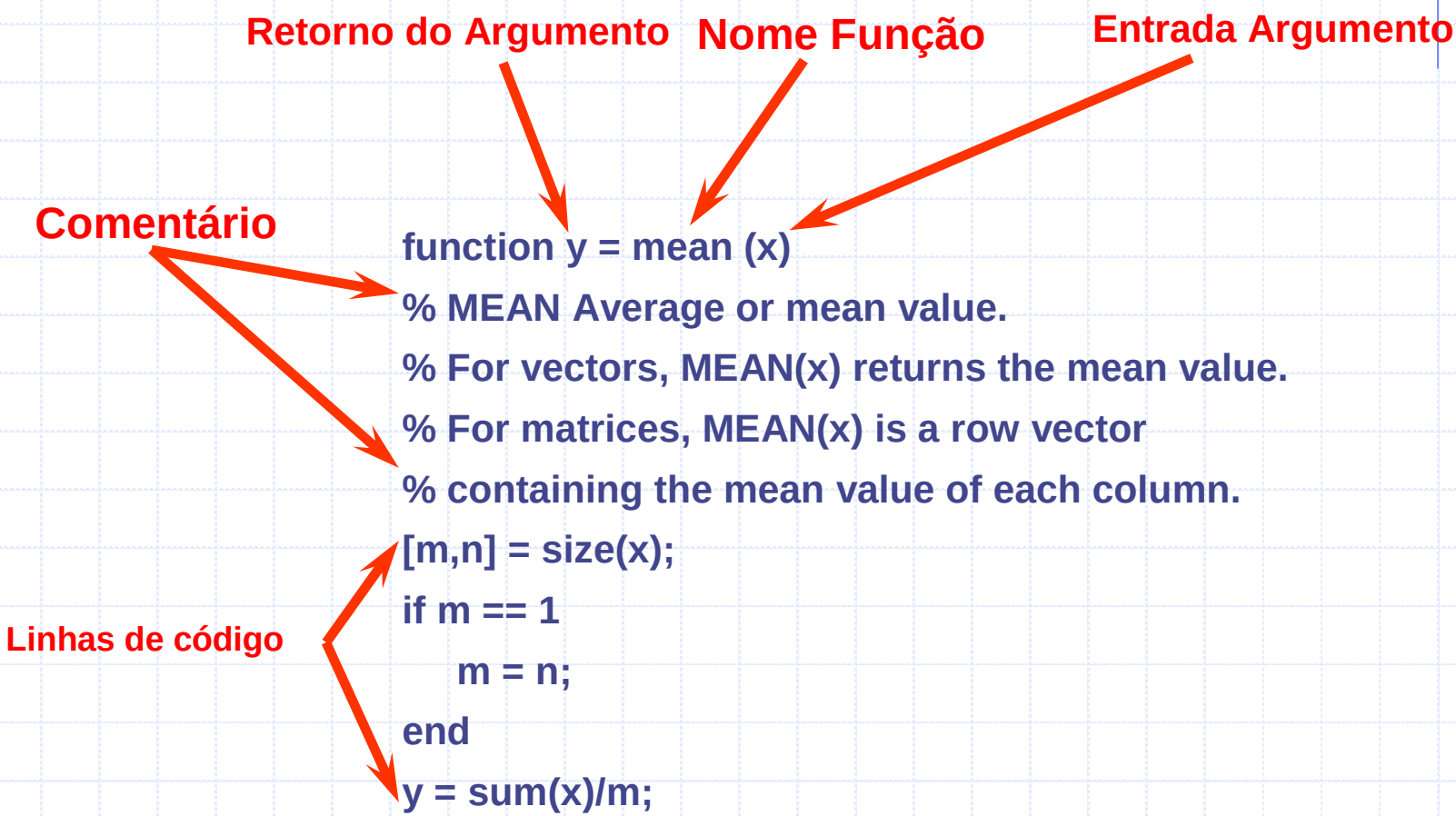
- O que é um M-File?
  - Editor: No lugar do uso da janela de comandos, um script (M-File) pode ser escrito contendo todos os comandos de linha. Sendo este ao ser submetido irá executar todos os comandos nele contidos.
  - Vantagem

Programação e Desenvolvimento de Aplicações

Script e Arquivos de Função

- Arquivos do tipo Script
  - Funciona como se estivesse digitando os comandos na janela de comando
  - As variáveis são armazenadas na área de "workspace"
- Arquivos de Funções
  - Permite que se crie suas próprias funções
  - Todas as variáveis dentro da função são locais
  - Toda informação deve ser passada para a função como parametros
  - Sub-funções são permitidas (funções, uma dentro da outra)

Partes Básicas de um arquivo de Função



Comandos de Controle

Comando: if

```
if ((attendance >= 0.90) & (grade_average >= 60))
   pass = 1;
end;
```

while Loops

```
eps = 1;
while (1+eps) > 1
    eps = eps/2;
end
eps = eps*2
```

Comandos de Controle

for Loop

```
a = zeros(k,k)  % Preallocate matrix
for m = 1:k
    for n = 1:k
        a(m,n) = 1/(m+n -1);
    end
end
```

Comando Decisão: switch

```
method = 'Bilinear';
switch lower(method)
   case {'linear','bilinear'}
      disp('Method is linear')
   case 'cubic'
      disp('Method is cubic')
otherwise
      disp('Unknown method.')
end
Method is linear
```

Características de um Arquivo de Função

- Sub-Funções
- Número quaisquer de argumentos: input/output
- ***Variáveis: Local*** e ***Global***
- Uso de "Input"
  - Input via teclado
  - "Pausa" durante a execução
- Erros e Avisos(Warnings)
  - Mostra mensagens de *erro* and "w*arning*"
- Função "Shell Escape" (Operador *!*)
- Otimização de código em MATLAB
  - Vetorização de *loops*
  - Pré-alocação *Arrays*

Arquivo de Função(arquivo M)

```
function r = ourrank(X,tol)
% rank of a matrix
s = svd(X);
if (nargin == 1)
   tol = max(size(X)) * s(1)* eps;
end
r = sum(s > tol);
```

Multíplos Input Argumentos
Uso: ()

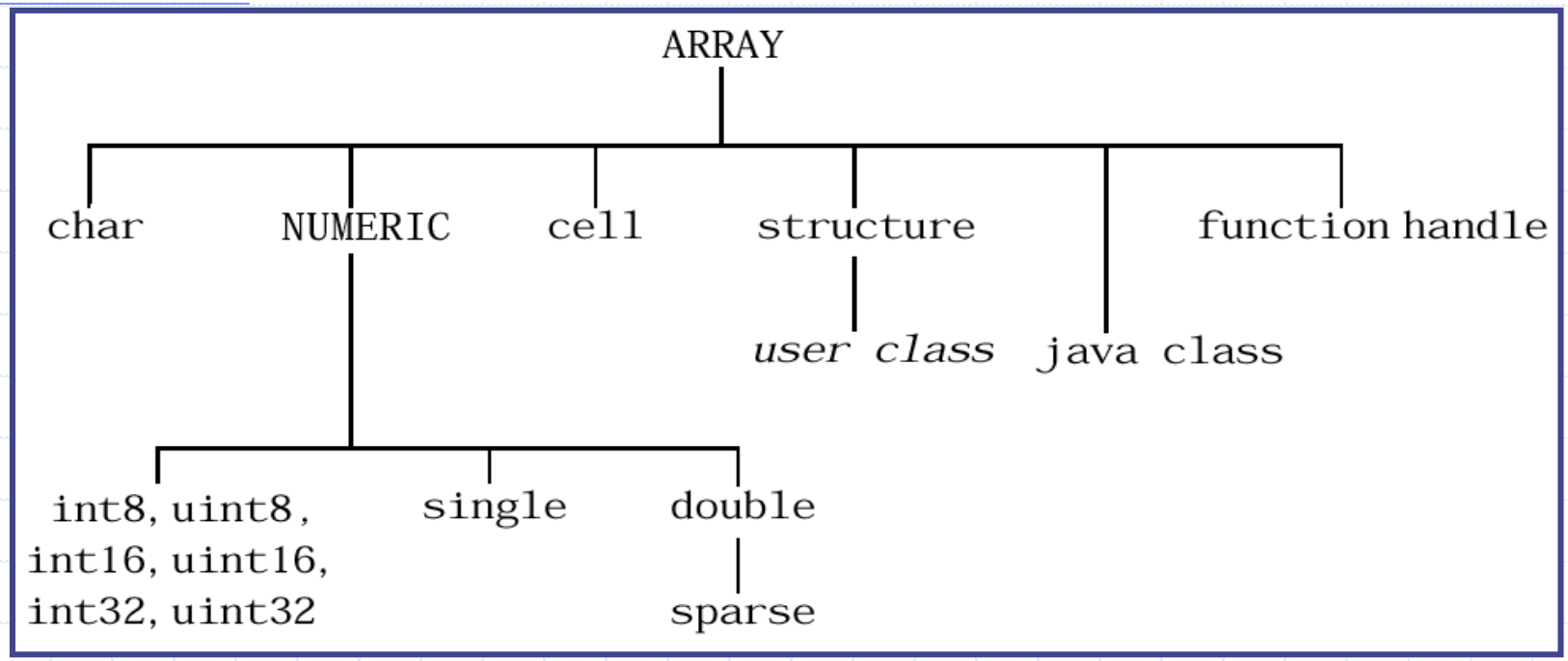
```
»r=ourrank(rand(5),.1);
```

Multíplos Output Argumentos. Uso []

```
»[m std]=ourstat(1:9);
```

```
function [mean,stdev] = ourstat(x)
[m,n] = size(x);
if m == 1
   m = n;
end
mean = sum(x)/m;
stdev = sqrt(sum(x.^2)/m - mean.^2);
```

Tipos de Dados



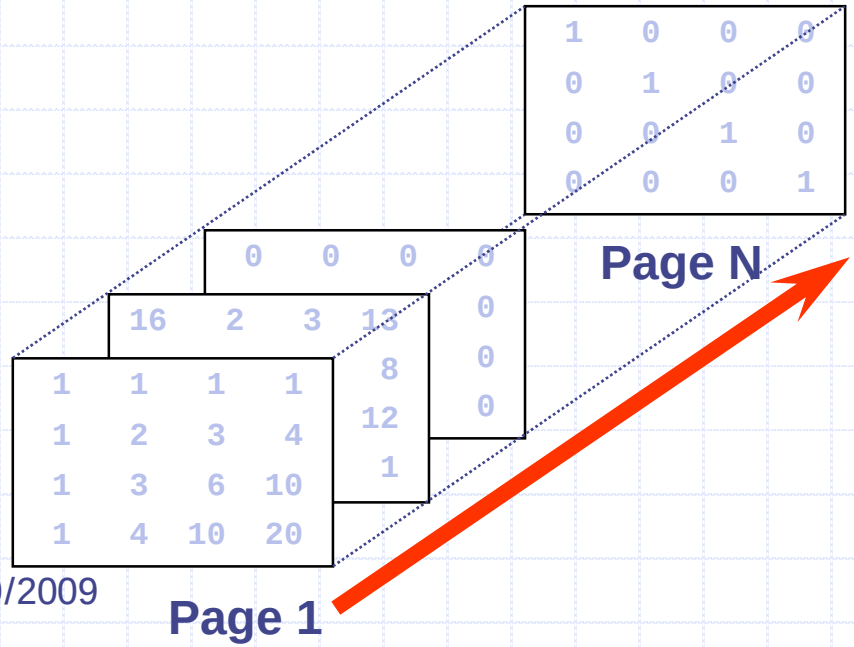
- Arrays Númerico
- Array Multidimensional
- Estruturas e “Cell Arrays”

Arrays Multidimensional

A primeira referência, vetor 1D, a linha.

A segunda referência matriz 2D, a coluna.

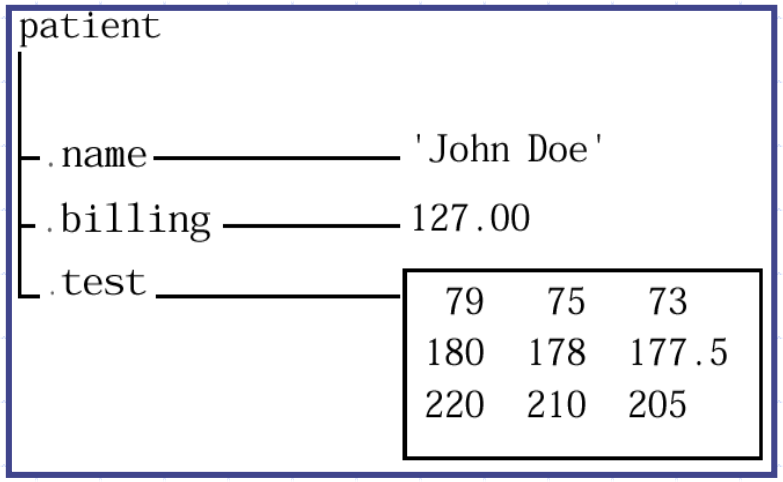
A terceira referência 3D, profundidade (Page).




```
» A = pascal(4);
» A(:,:,2) = magic(4)
A(:,:,1) =
     1     1     1     1
     1     2     3     4
     1     3     6    10
     1     4    10    20
A(:,:,2) =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
» A(:,:,9) =
  diag(ones(1,4));
```

Estruturas e Arrays de Estruturas

- Arrays com “containers” de dados chamado ***fields***.




```
» patient.name='John Doe';
» patient.billing = 127.00;
» patient.test= [79 75 73;
180 178 177.5;
220 210 205];
```

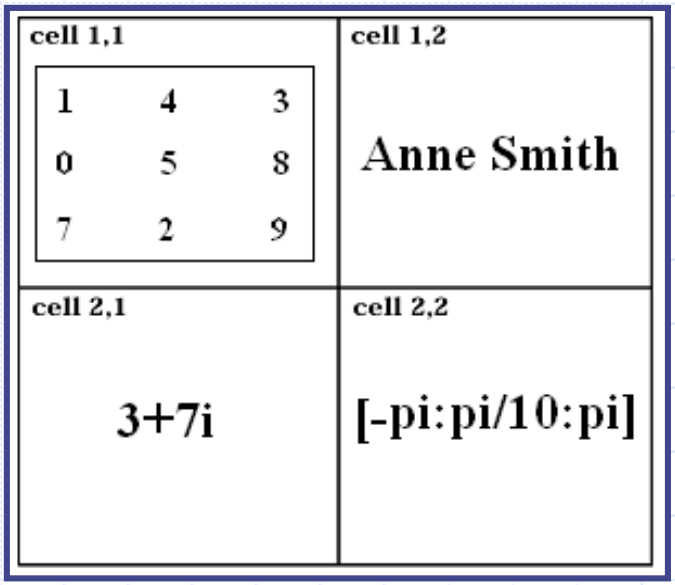
- Também, constrói estrutura de arrays usando a função ***struct***.
- Array de “*structures*”

```
» patient(2).name='Katty Thomson';
» Patient(2).billing = 100.00;
» Patient(2).test= [69 25 33; 120 128 177.5; 220
  210 205];
```

Cell Arrays

- Array nos quais os elementos são *cells* e pode conter outros arrays MATLAB de diferente tipos.

```
» A(1,1) = {[1 4 3;
0 5 8;
7 2 9]};
» A(1,2) = {'Anne Smith'};
» A(2,1) = {3+7i};
» A(2,2) = {-pi:pi/10:pi};
```



- Usa-se ***chave* {}** para apontar os elementos de um "cell array"
- Usando a função ***celldisp*** para mostrar "cell array"

Funções Numéricas Não-lineares

- ***Função inline***

Uso função ***char*** para converter *inline* objeto p/ *string*

```
» f = inline('3*sin(2*x.^2)','x')
f =
     Inline function:
     f(x) = 3*sin(2*x.^2)
» f(2)
ans =
    2.9681
```

- Integração Numérica usando ***quad***

```
» Q = quad('1./(x.^3-2*x-5)',0,2);
» F = inline('1./(x.^3-2*x-5)');
» Q = quad(F,0,2);
» Q = quad('myfun',0,2)
```

```
function y = myfun(x)
y = 1./(x.^3-2*x-5);
```

Nota: a função ***quad*** usa quadratura de Simpson adaptativa

Funções Numéricas Não-Linear

- *fzero acha o zero de uma função de uma variável simples*

```
[x,fval]= fzero(fun,x0,options)
```

  - fun é uma função "inline" ou "m-function"
- ***fminbnd*** minimiza uma função simples num intervalo fixo: $x_1<x<x_2$

```
[x,fval]= fminbnd(fun,x1,x2,options)
```

- ***fminsearch*** minimize uma função de várias variáveis

```
[x,fval]= fminsearch(fun,x0,options)
```

- Use ***optimset*** to determine *options* parameter.

```
options = optimset('param1',value1,...)
```

Equações Diferenciais Ordinárias (Problema de Valor Inicial)

- Uma EDO Explicita com VI:

$$y' = f(t, y)$$
$$y(t_0) = y_0$$

- Usando *ode45* para função não-stiff e *ode23t* para função stiff.

```
[t,y] = solver(odefun,tspan,y0,options)
```

```
function dydt = odefun(t,y)
```

```
Initialvlue
```

```
[initialtime  finaltime]
```

- Uso de ***odeset*** p/ definir opções de parâmetros

Exemplo de EDO

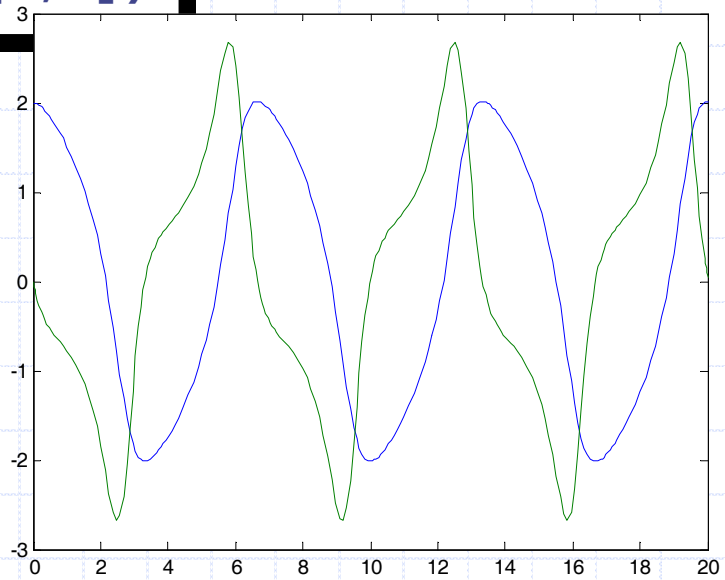
$$y_1'' - (1 - y_1^2)\, y_1' + y_1 = 0$$

$$y_1' = y_2$$

$$y_2' = (1 - y_1^2) y_2 - y_1$$

```
function dydt=myfunc(t,y)
dydt=zeros(2,1);
dydt(1)=y(2);
dydt(2)=(1-y(1)^2)*y(2)-y(1);
```

```
» [t,y]=ode45('myfunc',[0 20],[2;0])
```

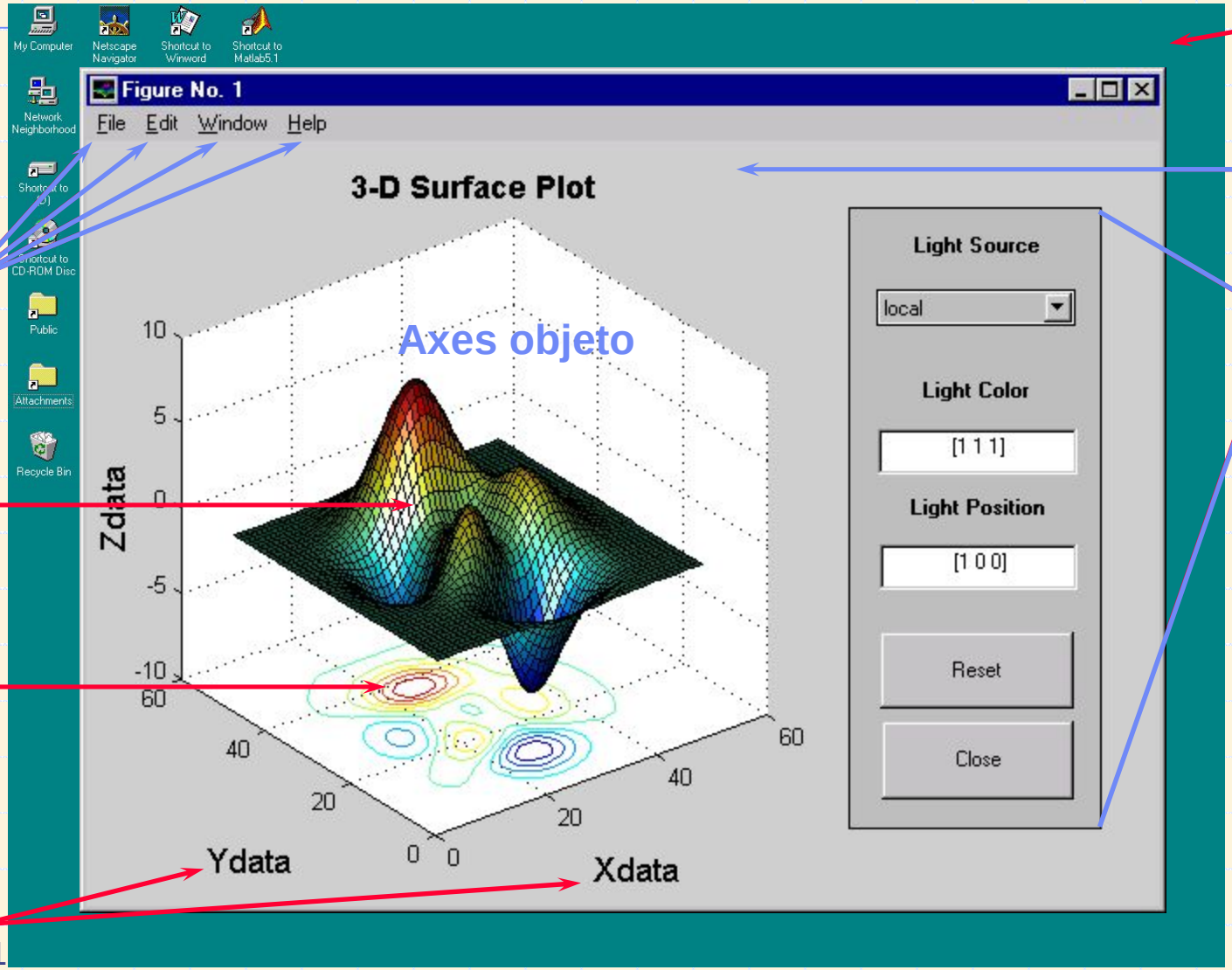


Nota:
Help em ***odeset*** *p/* opções, mais precisão e outros utilidades como plotagem durante a solução.

Gráficos “Handle”

- Gráficos em MATLAB consistem de o*bjetos:*
  - *root, figure, axes, image, line, patch, rectangle, surface, text, light*
- Cria Objetos
- Modifica Propriedades dos objetos
- Obtém “Handles” dos objetos
- Conhecer as Propriedades dos Objetos
- Modifica Propriedades dos Objetos
  - Usando Linha de *Comando*
  - Usando *Propriedades de Editor*

Objetos Gráficos



Root objeto

Figura objeto

UIMenu objetos

Axes objeto

UIControle objetos

Surface objeto

Line objetos

Texto objetos

Obtendo um “Handle” Objeto

1. Criando

```
h_line = plot(x_data, y_data, ...)
```

2. Funções utilitárias
   - 0 - root object handle
   - gcf - current figure handle
   - gca- current axis handle
   - gco- current object handle

Qual é objeto atual?
Último objeto criado
- **ou**
- **Último objeto clicked**

3. FINDOBJ

```
h_obj = findobj(h_parent, 'Property', 'Value', ...)
```

Default = 0 (root object)

Modificando Propriedades de Objetos

- Obtendo uma lista das propriedades:

```
get(h_object)
```

- Obtendo uma lista das propriedades passives de mudanças:

```
set(h_object)
```

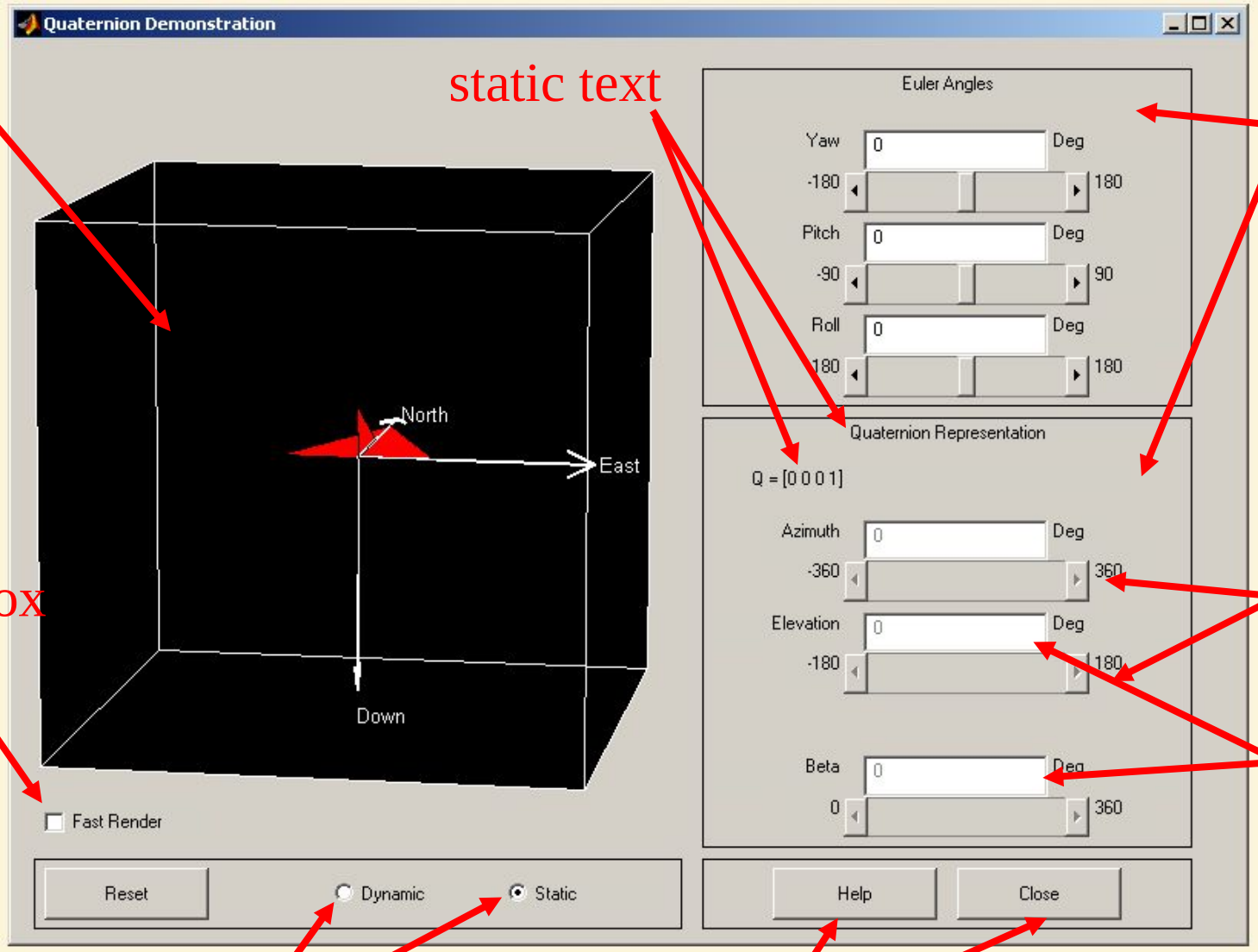
- Modificando as propriedades de objeto
  - Usando a Linha de Comando

```
set(h_object,'PropertyName','New_Value',...)
```

  - Usando Editor de Propriedade

Graphical User Interface: GUI

- O que é GUI?
- O que é *figure* and *.fig?
- Usando comando *guide*
- GUI, controles
- GUI, menus



Axes

static text

Frames

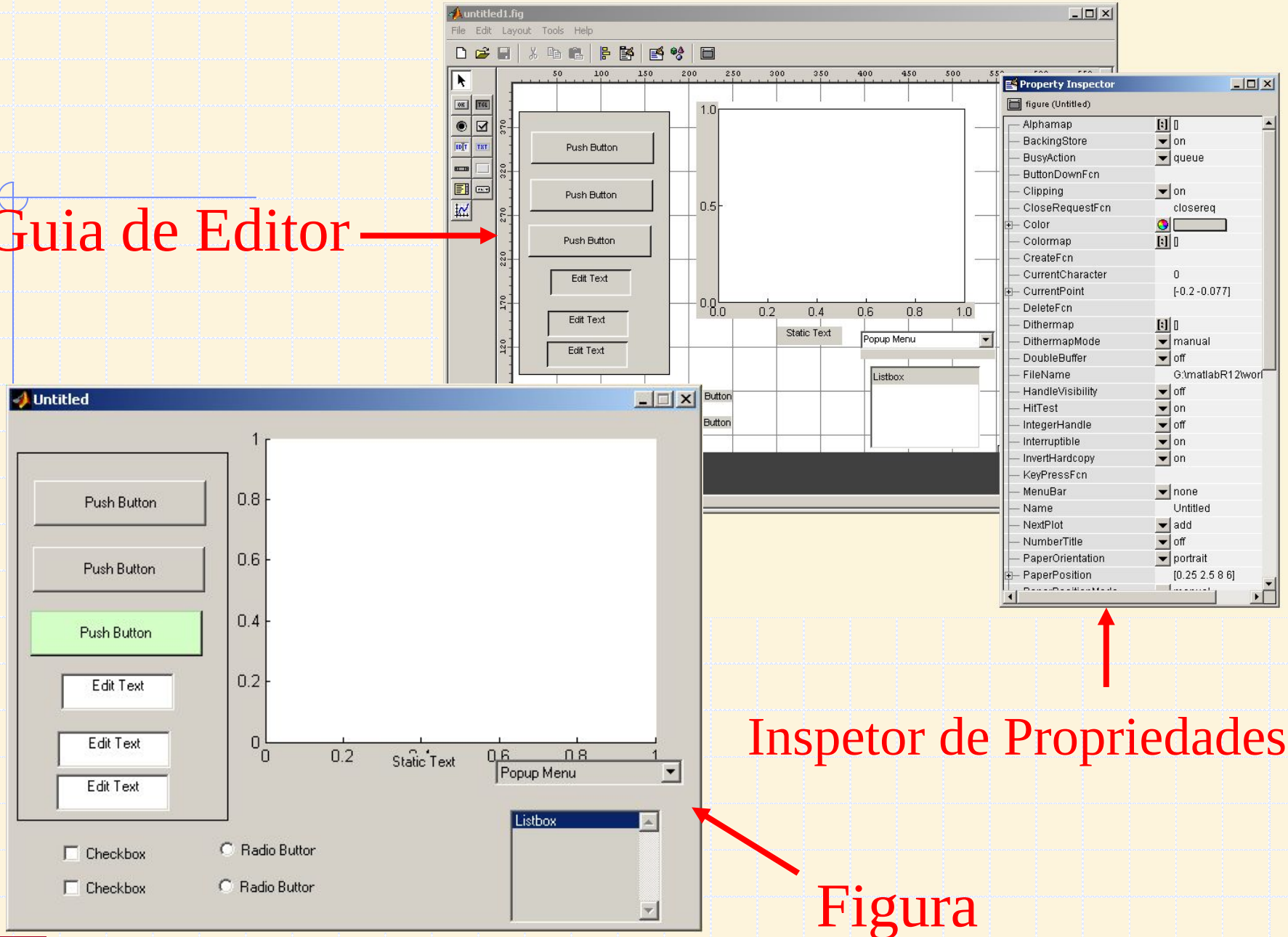
Checkbox

Slider

Edit text

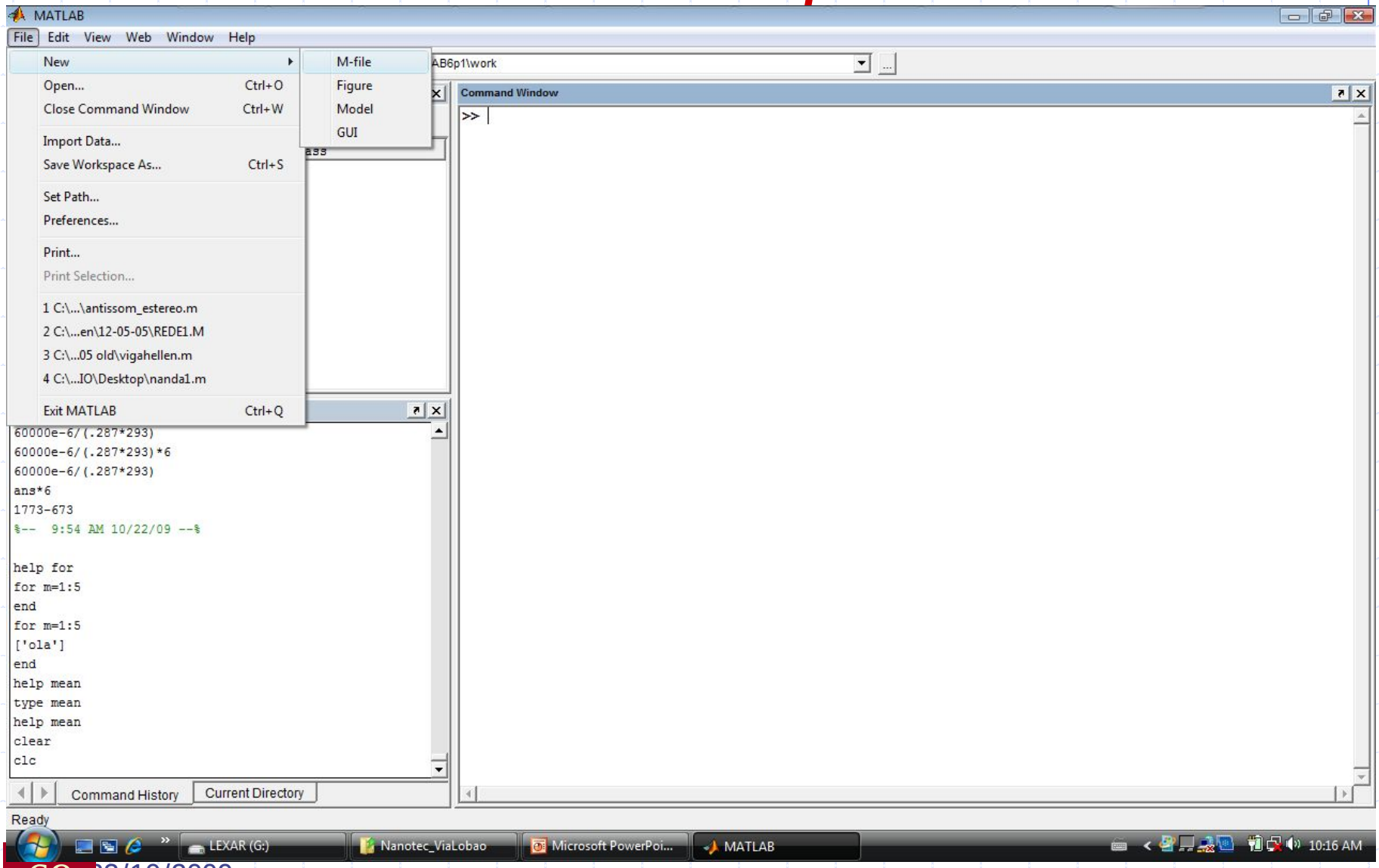
Radio Buttons

Push Buttons

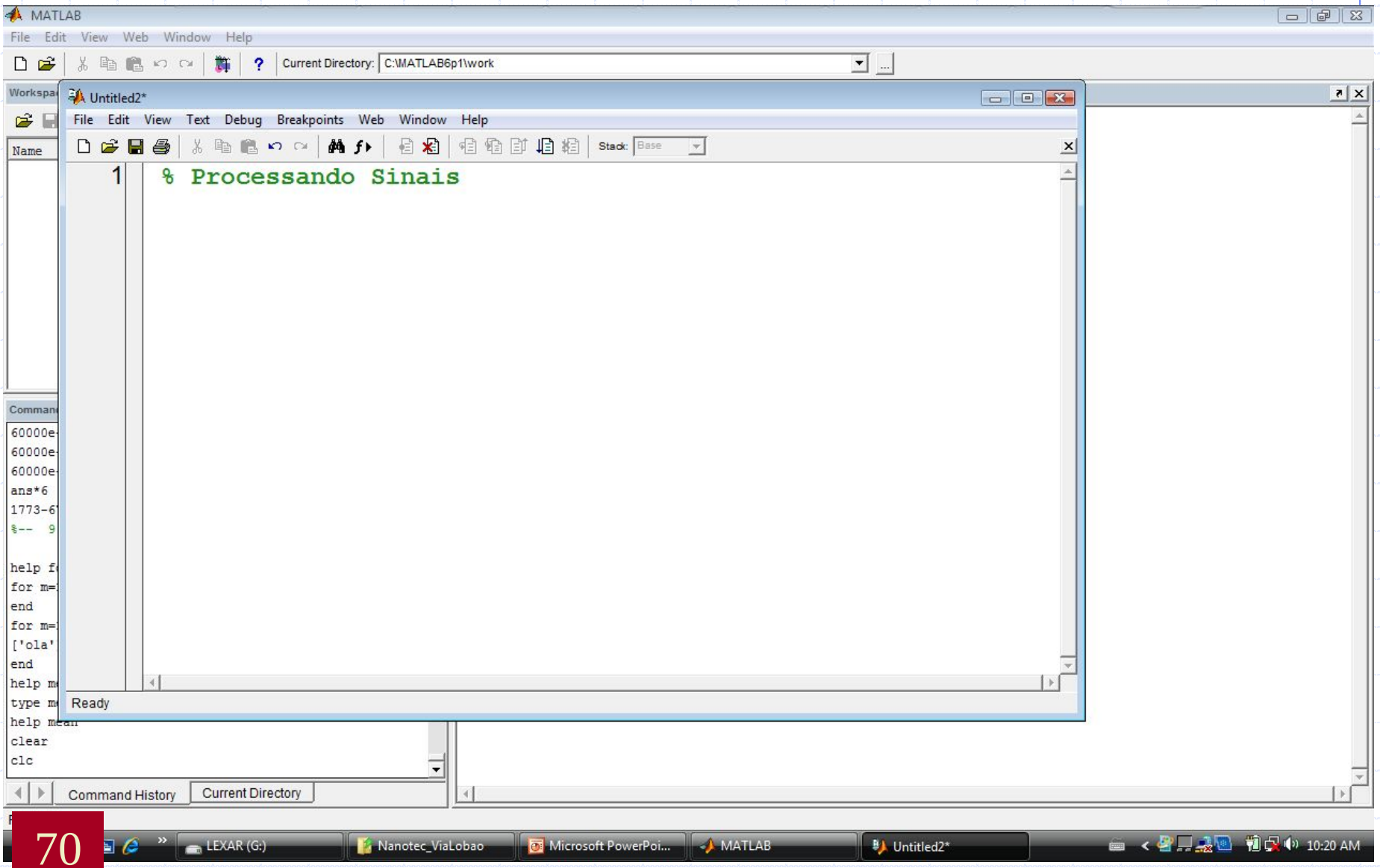

Guia de Editor
Inspetor de Propriedades
Figura
untitled1.fig
File Edit Layout Tools Help
Push Button
Push Button
Push Button
Edit Text
Edit Text
Edit Text
Static Text
Popup Menu
Listbox
Button
Button
Property Inspector
figure (Untitled)
Alphamap
BackingStore
BusyAction
ButtonDownFcn
Clipping
CloseRequestFcn
Color
Colormap
CreateFcn
CurrentCharacter
CurrentPoint
DeleteFcn
Dithermap
DithermapMode
DoubleBuffer
FileName
HandleVisibility
HitTest
IntegerHandle
Interruptible
InvertHardcopy
KeyPressFcn
MenuBar
Name
NextPlot
NumberTitle
PaperOrientation
PaperPosition
Untitled
Push Button
Push Button
Push Button
Edit Text
Edit Text
Edit Text
Static Text
Popup Menu
Listbox
Checkbox
Checkbox
Radio Buttor
Radio Buttor

Exemplo Aplicativo em Processamento de Sinais

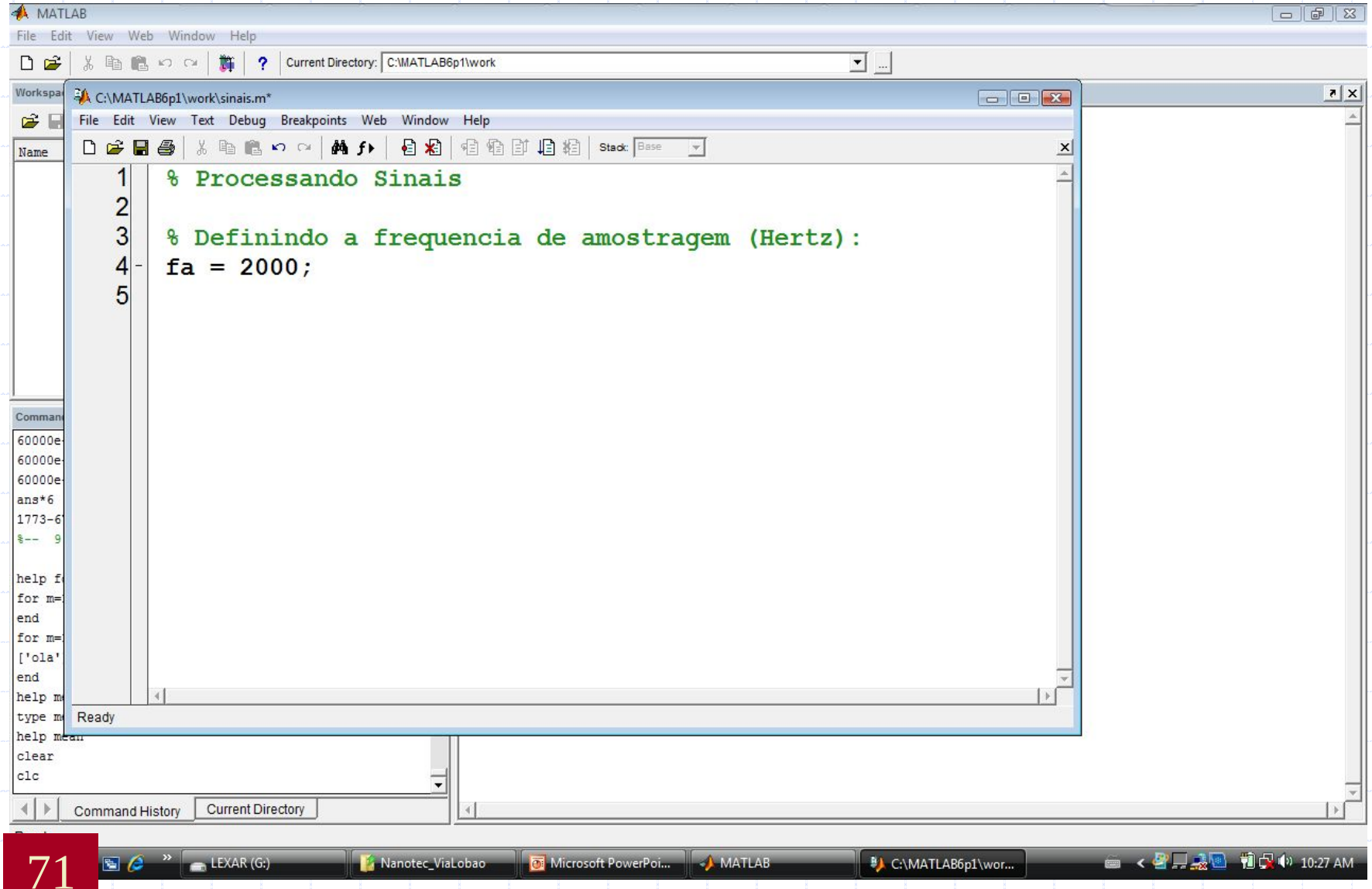
No editor crie o *script* sinais.m

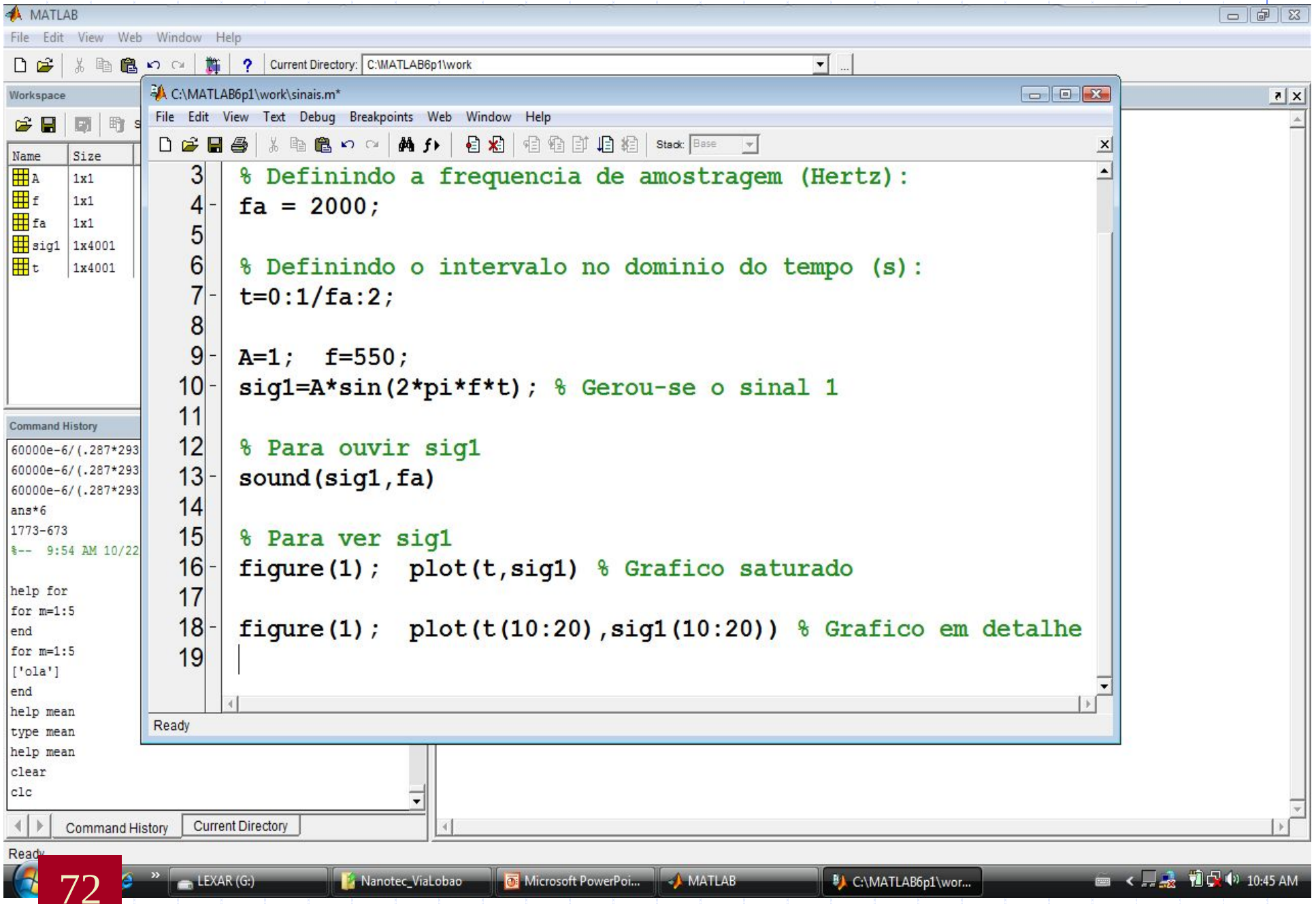


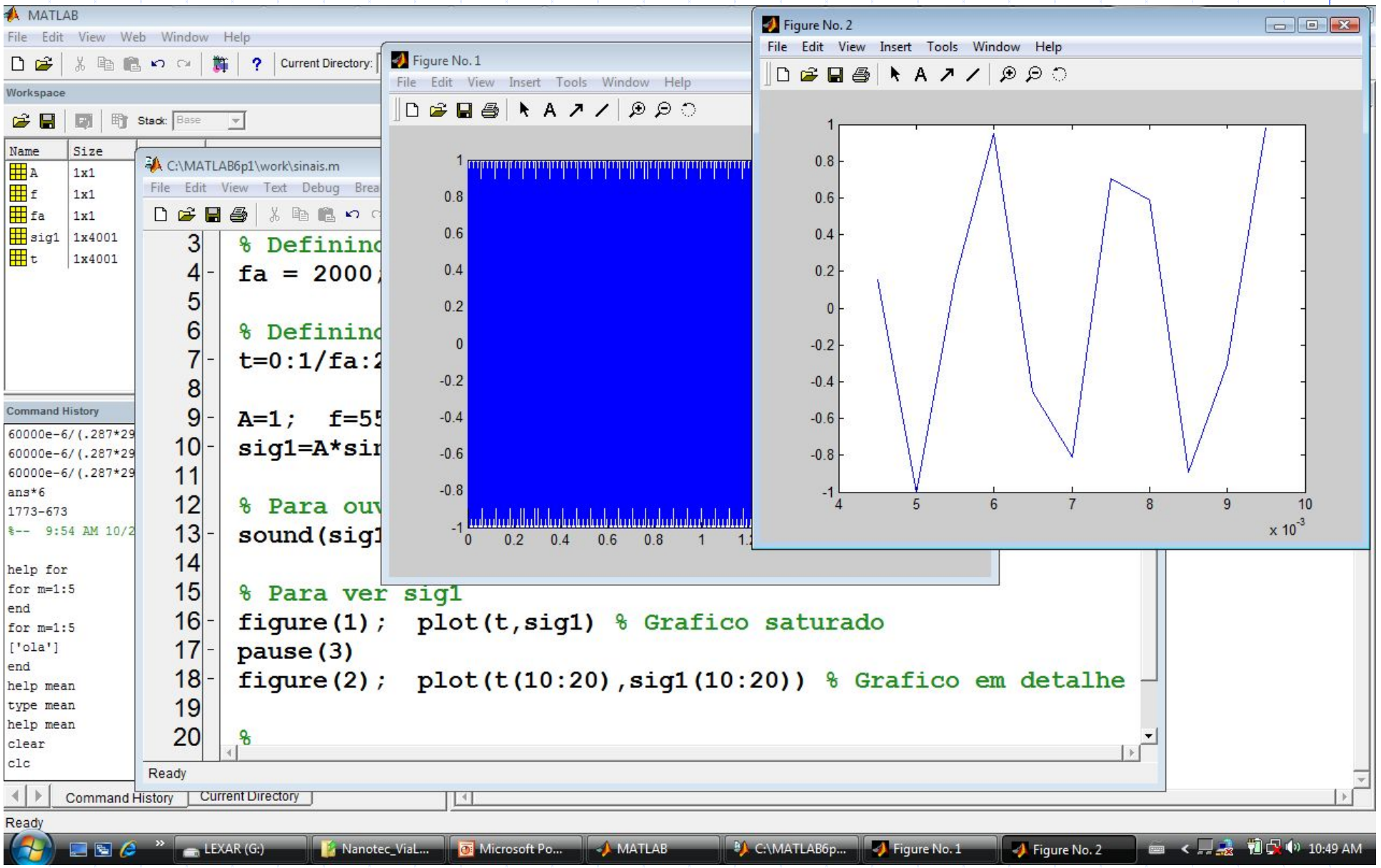
Nome temporário: Untitled

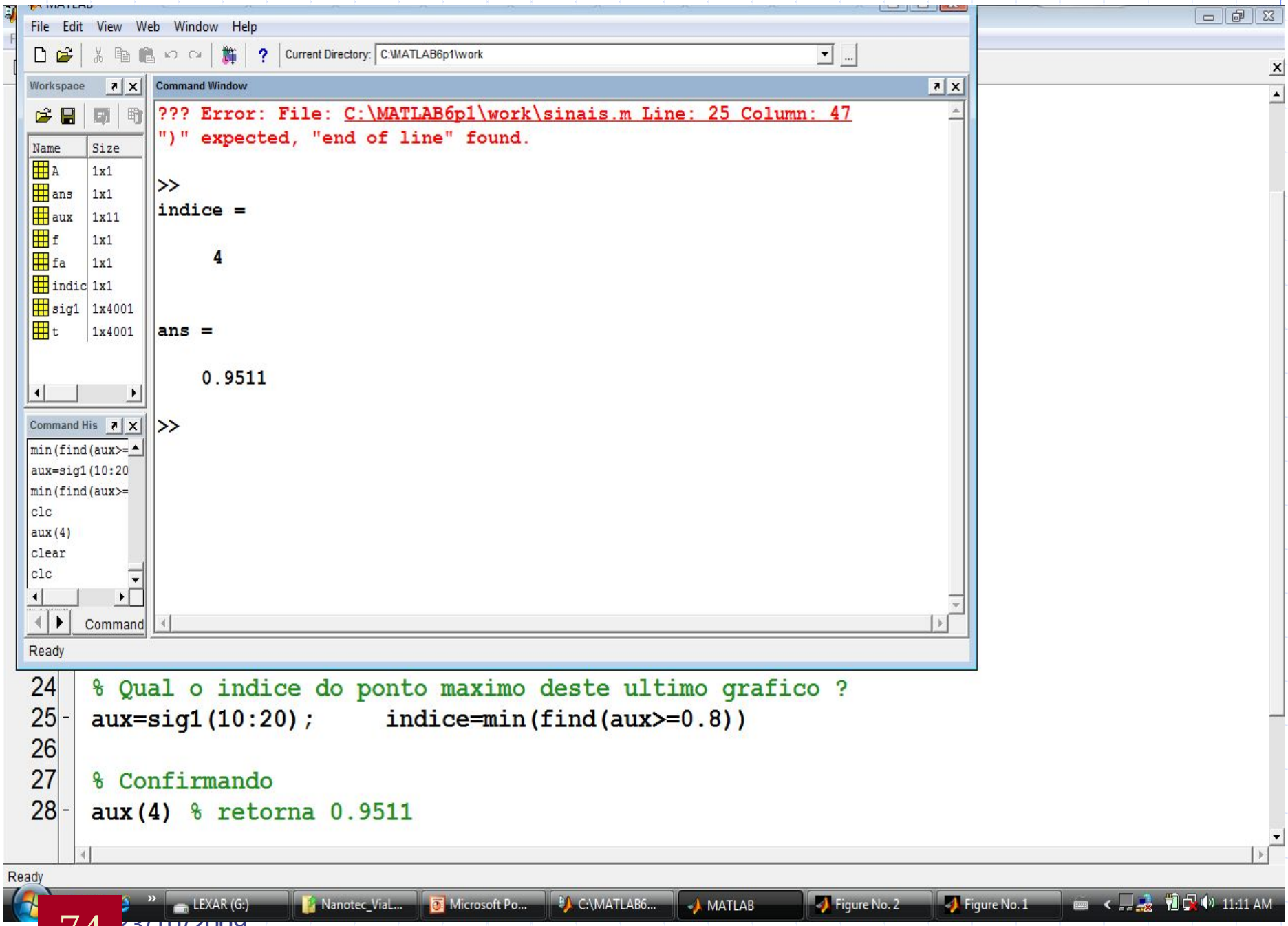


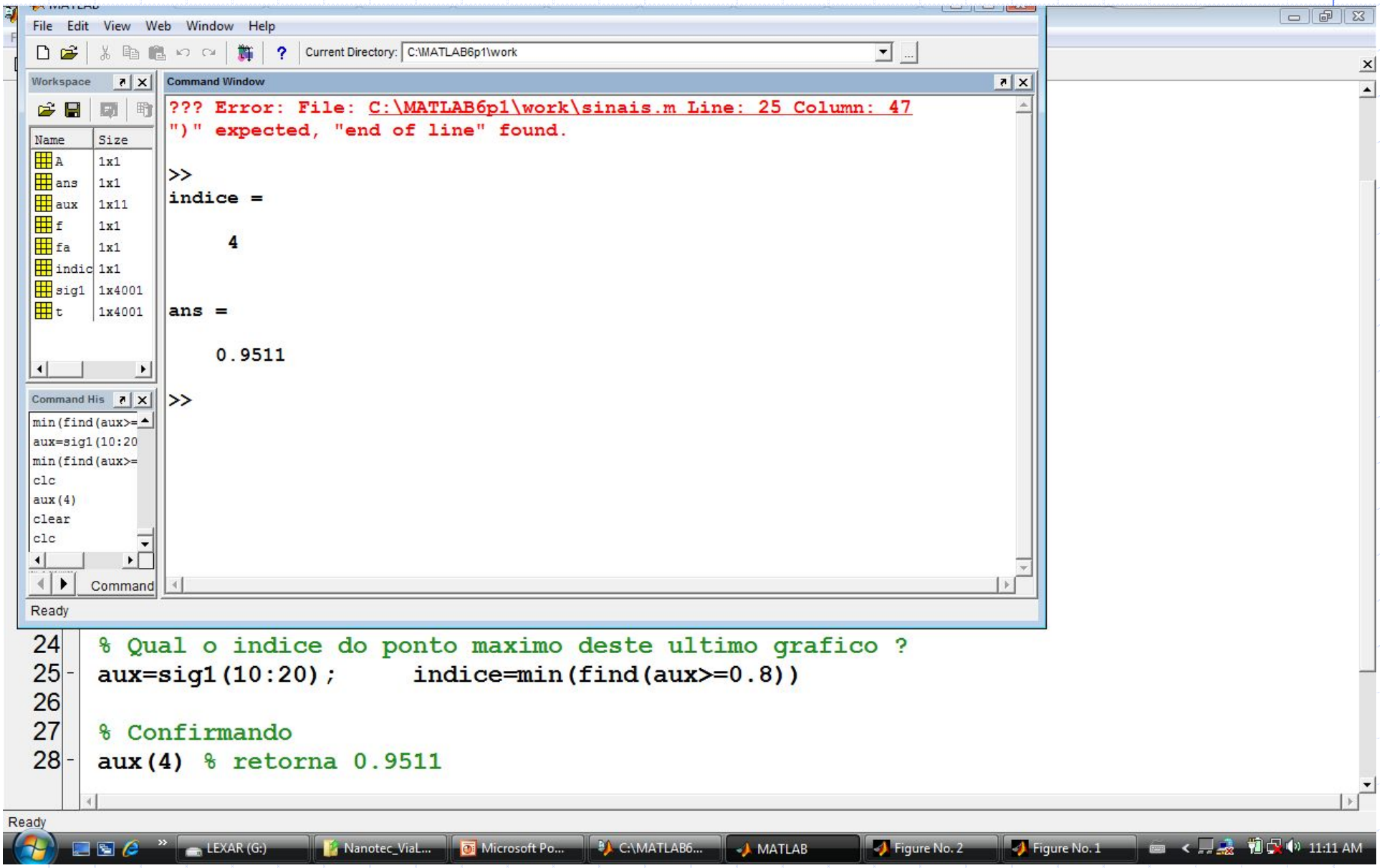
Nome “salvo”: sinais




MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work
Workspace
Name Size
A 1x1
f 1x1
fa 1x1
sig1 1x4001
t 1x4001
Command History
60000e-6/(.287*293
60000e-6/(.287*293
60000e-6/(.287*293
ans*6
1773-673
%-- 9:54 AM 10/22
help for
for m=1:5
end
for m=1:5
['ola']
end
help mean
type mean
help mean
clear
clc
Command History
Current Directory
Ready
C:\MATLAB6p1\work\sinais.m*
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
3 % Definindo a frequencia de amostragem (Hertz):
4 fa = 2000;
5
6 % Definindo o intervalo no dominio do tempo (s):
7 t=0:1/fa:2;
8
9 A=1; f=550;
10 sig1=A*sin(2*pi*f*t); % Gerou-se o sinal 1
11
12 % Para ouvir sig1
13 sound(sig1,fa)
14
15 % Para ver sig1
16 figure(1); plot(t,sig1) % Grafico saturado
17
18 figure(1); plot(t(10:20),sig1(10:20)) % Grafico em detalhe
19
Ready
LEXAR (G:)
Nanotec_ViaLobao
Microsoft PowerPoi...
MATLAB
C:\MATLAB6p1\wor...
10:45 AM


MATLAB
File Edit View Web Window Help
Current Directory:
Workspace
Stack: Base
Name Size
A 1x1
f 1x1
fa 1x1
sig1 1x4001
t 1x4001
Command History
60000e-6/(.287*29
60000e-6/(.287*29
60000e-6/(.287*29
ans*6
1773-673
%-- 9:54 AM 10/2
help for
for m=1:5
end
for m=1:5
['ola']
end
help mean
type mean
help mean
clear
clc
C:\MATLAB6p1\work\sinais.m
File Edit View Text Debug Brea
3 % Defininc
4 - fa = 2000;
5
6 % Defininc
7 - t=0:1/fa:2
8
9 - A=1; f=5
10 - sig1=A*sir
11
12 % Para ouv
13 - sound(sig1
14
15 % Para ver sig1
16 - figure(1); plot(t,sig1) % Grafico saturado
17 - pause(3)
18 - figure(2); plot(t(10:20),sig1(10:20)) % Grafico em detalhe
19
20 %
Ready
Figure No. 1
File Edit View Insert Tools Window Help
Figure No. 2
File Edit View Insert Tools Window Help
x 10^-3
Command History
Current Directory
Ready
LEXAR (G:)
Nanotec_ViaL...
Microsoft Po...
MATLAB
C:\MATLAB6p...
Figure No. 1
Figure No. 2
10:49 AM

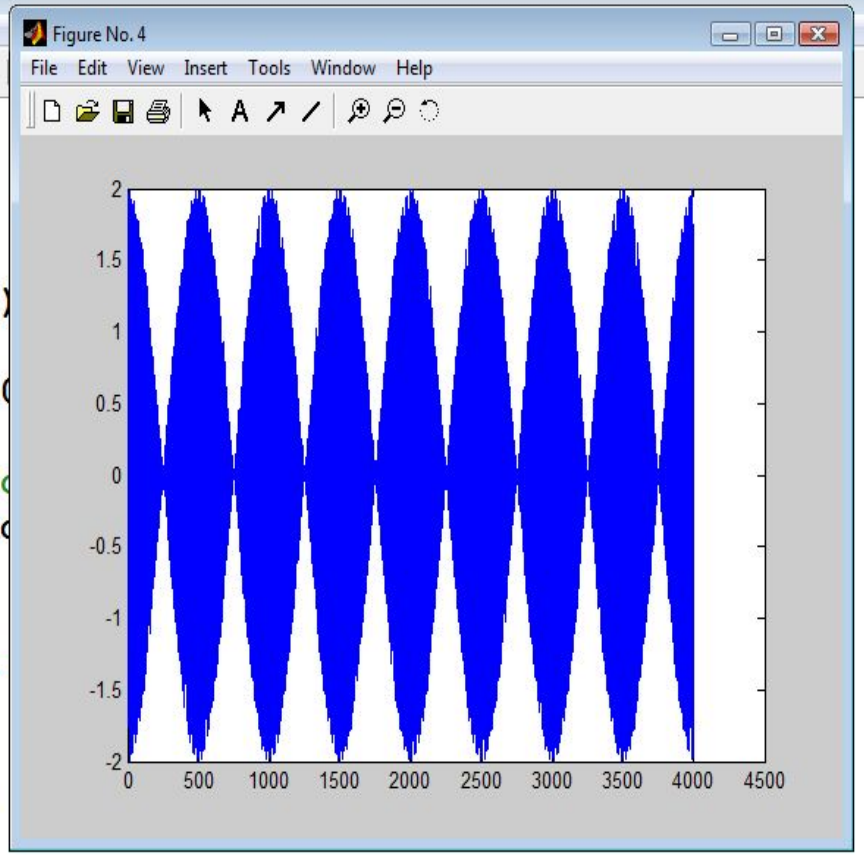

File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work
Workspace
Name Size
A 1x1
ans 1x1
aux 1x11
f 1x1
fa 1x1
indic 1x1
sig1 1x4001
t 1x4001
Command His
min(find(aux>=
aux=sig1(10:20
min(find(aux>=
clc
aux(4)
clear
clc
Command
Command Window
??? Error: File: C:\MATLAB6p1\work\sinais.m Line: 25 Column: 47
")" expected, "end of line" found.
>>
indice =
4
ans =
0.9511
>>
Ready
24 % Qual o indice do ponto maximo deste ultimo grafico ?
25 - aux=sig1(10:20); indice=min(find(aux>=0.8))
26
27 % Confirmando
28 - aux(4) % retorna 0.9511
Ready
LEXAR (G:)
Nanotec_ViaL...
Microsoft Po...
C:\MATLAB6...
MATLAB
Figure No. 2
Figure No. 1
11:11 AM


File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work
Workspace
Name Size
A 1x1
ans 1x1
aux 1x11
f 1x1
fa 1x1
indic 1x1
sig1 1x4001
t 1x4001
Command His
min(find(aux>=
aux=sig1(10:20
min(find(aux>=
clc
aux(4)
clear
clc
Command
Command Window
??? Error: File: C:\MATLAB6p1\work\sinais.m Line: 25 Column: 47
")" expected, "end of line" found.
>>
indice =
4
ans =
0.9511
>>
Ready
24 % Qual o indice do ponto maximo deste ultimo grafico ?
25 aux=sig1(10:20); indice=min(find(aux>=0.8))
26
27 % Confirmando
28 aux(4) % retorna 0.9511
Ready
LEXAR (G:)
Nanotec_ViaL...
Microsoft Po...
C:\MATLAB6...
MATLAB
Figure No. 2
Figure No. 1
11:11 AM

C:\MATLAB6p1\work\sinais.m

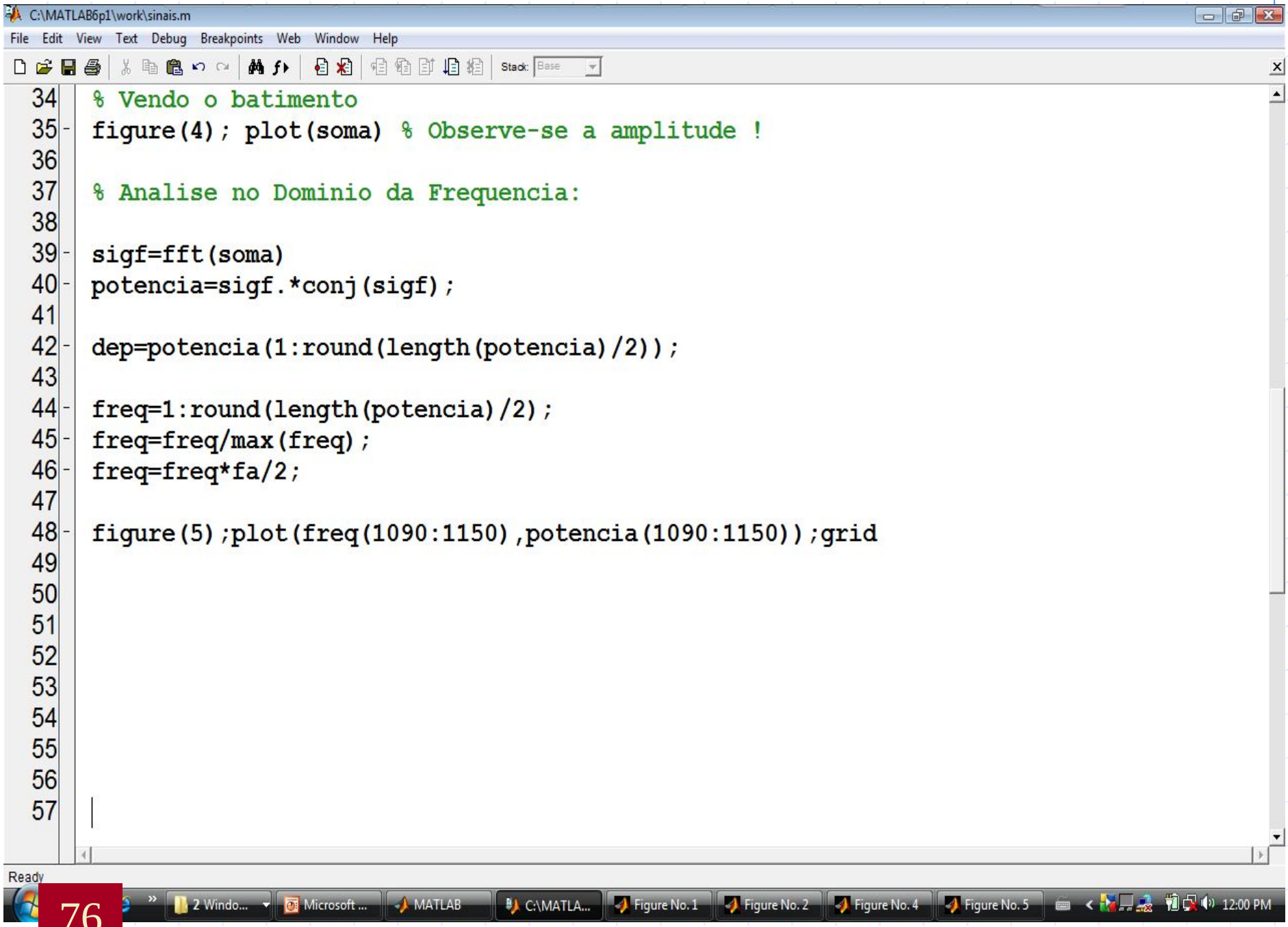
File Edit View Text Debug Breakpoints Web Window Help

```
12  % Para ouvir sig1
13  sound(sig1,fa)
14
15  % Para ver sig1
16  figure(1);  plot(t,sig1)
17  pause(3)
18  figure(2);  plot(t(10:2(
19
20  % Qual o indice do ponto
21  aux=sig1(10:20);      inc
22
23  % Confirmando
24  aux(4) % retorna 0.9511
25
26  % Composicao de Sinais
27  sig2=A*sin(2*pi*554*t);
28  soma=sig1+sig2;
29
30  % Ouvindo-os
31  sound(sig2,fa)
32  pause(6); sound(soma,fa)
33
34  % Vendo o batimento
35  figure(4); plot(soma)
```



Ready

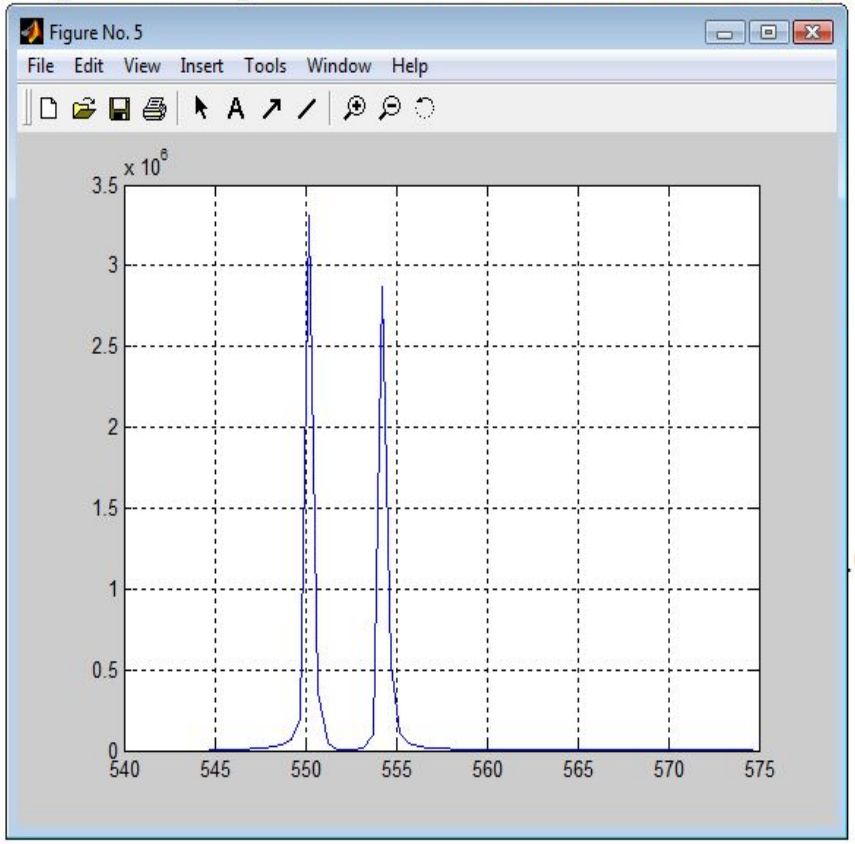
LEXAR (G:) | Nanotec_ViaLobao | Microsoft PowerP... | C:\MATLAB6p1\w... | MATLAB | Figure No. 4 | 11:25 AM


C:\MATLAB6p1\work\sinais.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
34 % Vendo o batimento
35 figure(4); plot(soma) % Observe-se a amplitude !
36
37 % Analise no Dominio da Frequencia:
38
39 sigf=fft(soma)
40 potencia=sigf.*conj(sigf);
41
42 dep=potencia(1:round(length(potencia)/2));
43
44 freq=1:round(length(potencia)/2);
45 freq=freq/max(freq);
46 freq=freq*fa/2;
47
48 figure(5);plot(freq(1090:1150),potencia(1090:1150));grid
49
50
51
52
53
54
55
56
57
Ready
2 Windo...
Microsoft ...
MATLAB
C:\MATLA...
Figure No. 1
Figure No. 2
Figure No. 4
Figure No. 5
12:00 PM

C:\MATLAB6p1\work\sinais.m

File Edit View Text Debug Breakpoints Web Window Help

```
34  % Vendo o batimento
35  figure(4); plot(soma) % Observe-se a amplitude !
36
37
38
39
40
41
42                                                    ;
43
44
45
46
47
48                                                    090:1150));grid
49
50
51
52
53
54
55
56
57
```



Ready

2 Windo... | Microsoft ... | MATLAB | C:\MATLA... | Figure No. 1 | Figure No. 2 | Figure No. 4 | Figure No. 5 | 12:02 PM

Conclusão

- Matlab é uma linguagem de computação.
- Matlab, é uma linguagem de alto desempenho e alto nível.
- Matlab suporta GUI, API.
- Matlab possue “Toolboxes” p/ aplicações específicas.
- Matlab é a melhor Linguagem!!!!

Mais informações...

- http://www.mathworks.com/support
  - FAQ sobre Mathworks.
- comp.soft-sys.matlab "Newsgroup"
  - Usando página: Google Groups
    http://groups.google.com/