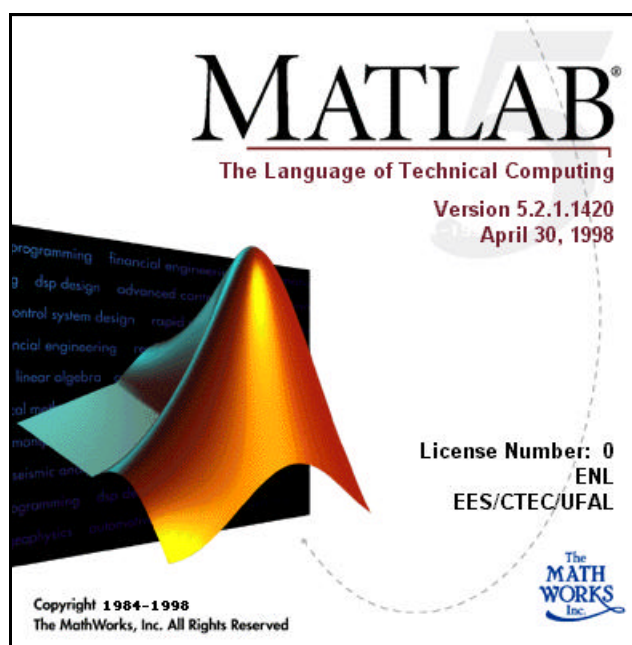




Universidade Federal de Alagoas - UFAL
Centro de Tecnologia - CTEC
Departamento de Engenharia Estrutural - EES

Cidade Universitária - Campus A. C. Simões
Tabuleiro do Martins - Maceió - AL - CEP 57072-970
Tel: (082) 214 1277 - Fax: (082) 214 1625
E-mail: ees@ctec.ufal.br - Home page: <http://www.ctec.ufal.br/ees>

INTRODUÇÃO AO MATLAB



Eduardo Nobre Lages

Professor Adjunto

E-mail: enl@ctec.ufal.br

Maceió - Alagoas
Maio de 1999

INTRODUÇÃO AO MATLAB

1. Introdução

O MATLAB (*MATrix LABoratory*) é um poderoso *software* de computação numérica, de análise e de visualização de dados, com um propósito bem mais amplo que o original “laboratório de matrizes”. O MATLAB é um sistema interativo e uma linguagem de programação para computação técnica e científica em geral.

Os comandos do MATLAB são muito próximos da forma como escrevemos expressões algébricas, tornando mais simples o seu uso. Podem ser incorporados às rotinas predefinidas, pacotes para cálculos específicos.

Uma vez inicializado o MATLAB, aparecerá na janela de comandos um *prompt* », para a versão completa (Figura 1), ou EDU», no caso da versão estudante. O *prompt* significa que o MATLAB está esperando um comando. Todo comando deve ser finalizado teclando-se *Enter*.

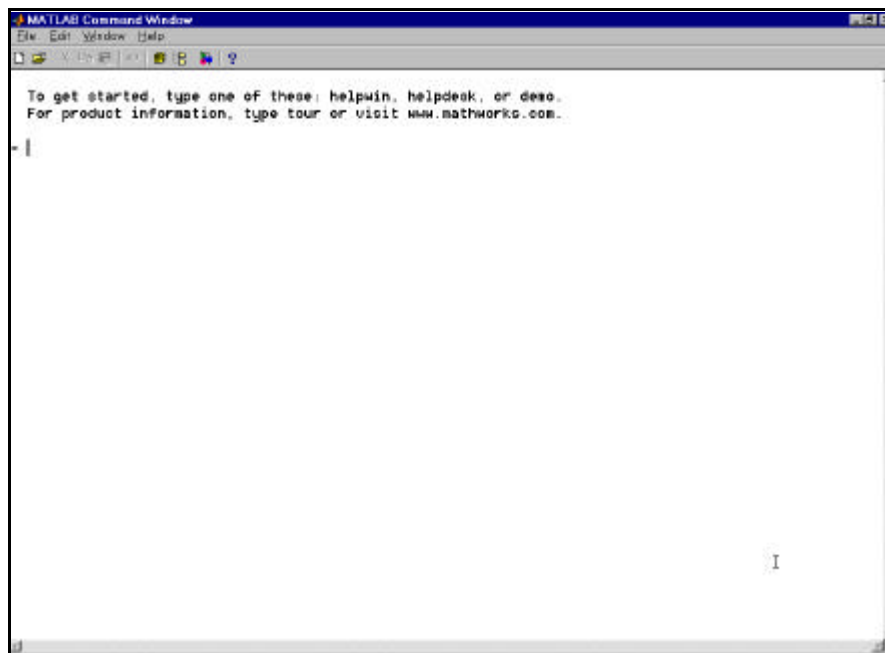


Figura 1 – Janela inicial do programa MATLAB (versão completa).

Para encerrar uma sessão de trabalho do MATLAB digita-se o comando *exit* (ou *quit*) no *prompt* do programa.

Inicialmente, para que o usuário avalie as potencialidades dos recursos do MATLAB que podem ser explorados em suas futuras implementações, recomenda-se a execução do programa demonstrativo (Figura 2), acessado através do comando *demo*.

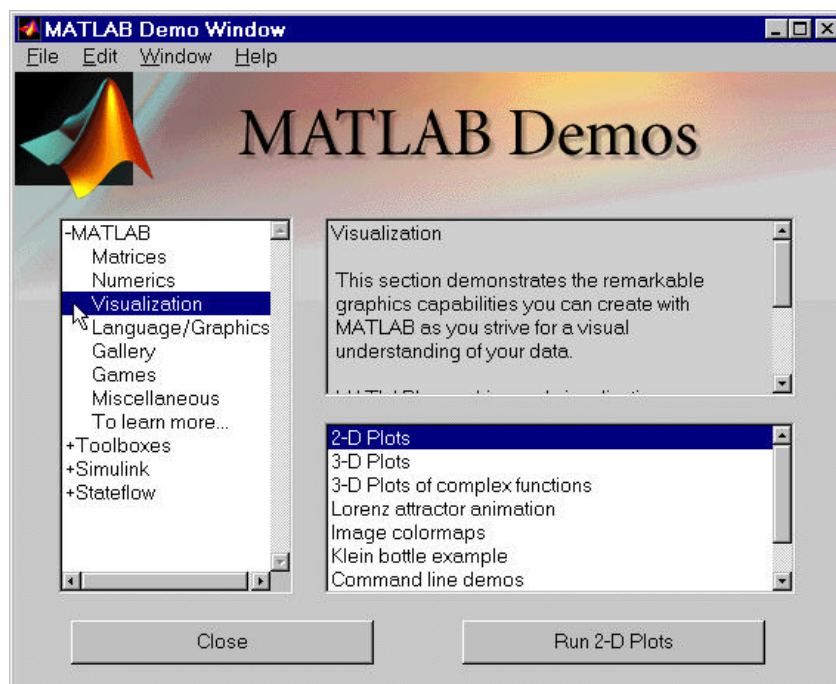


Figura 2 – Janela do demonstrativo do MATLAB.

A princípio, é interessante verificar na região à esquerda os vários blocos de demonstrativos da versão básica do MATLAB (*Matrices, Numerics, Visualization, etc*). Ao selecionar um destes blocos, na região inferior direita são listados os vários exemplos existentes para o bloco escolhido. O exemplo é executado quando aciona-se o botão inferior direito *Run*.

O MATLAB possui três ambientes de ajuda, a saber:

?? Comando *help*

Ao digitar o comando *help* o MATLAB apresenta uma listagem de todos os pacotes disponíveis (Tabela 1).

Tabela 1 – Resultado da execução do comando *help*.

```
» help

HELP topics:

matlab\general - General purpose commands.
matlab\ops - Operators and special characters.
...
For more help on directory/topic, type "help topic".
```

Para se ter ajuda sobre um pacote específico ou sobre um comando ou função específica, deve-se combinar o comando *help* e o nome do pacote, comando ou função de interesse (Tabelas 2 e 3).

Tabela 2 – Utilização do *help* com o nome de um pacote específico.

```
» help elfun

Elementary math functions.

Trigonometric.
sin      - Sine.
sinh     - Hyperbolic sine.
asin     - Inverse sine.
...

Exponential.
exp      - Exponential.
log      - Natural logarithm.
...

Complex.
abs      - Absolute value.
angle    - Phase angle.
...

Rounding and remainder.
fix      - Round towards zero.
floor    - Round towards minus infinity.
...
```

Tabela 3 – Ajuda da função *det*.

```
» help det

DET Determinant.
DET(X) is the determinant of the square matrix X.

Use COND instead of DET to test for matrix singularity.

See also COND.

Overloaded methods
help sym/det.m
```

?? Comando *helpwin*

Ao se executar o comando *helpwin*, abre-se uma nova janela textual de ajuda (Figura 3). Combinando-se o *double-click* sobre um dos itens e os botões *Back*, *Forward* e *Home* obtêm-se os vários níveis de ajuda de um pacote específico e de um comando ou função de interesse. O nome de uma função pode ser introduzida no campo superior esquerdo para se ter diretamente uma ajuda deste comando. No *pull-down menu* à direita deste campo encontram-se os tópicos relacionados com a função que está sendo explicada no momento.

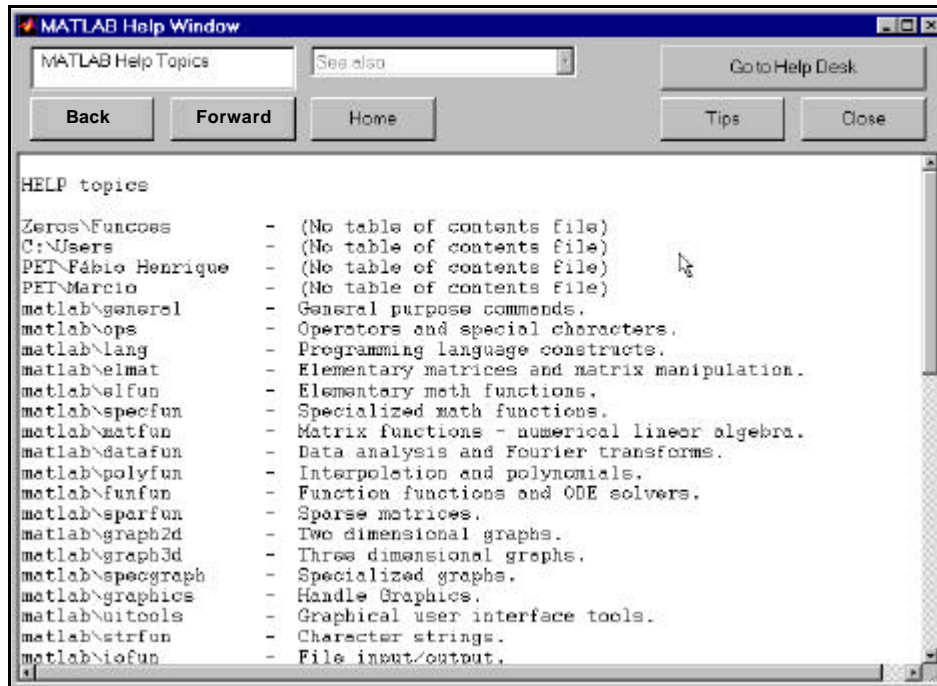


Figura 3 – Janela acionada pelo comando *helpwin*.

?? Comando *helpdesk*

Ao se executar o comando *helpdesk*, dispara-se um programa de navegação instalado no seu computador (*Netscape*, *Internet Explorer*, etc) com um ambiente de ajuda mais completo que utiliza a linguagem de hipertexto (Figura 4).



Figura 4 – Janela acionada pelo comando de ajuda *helpdesk*.

Neste ambiente de ajuda é possível obter uma listagem das funções por bloco de especificidade (MATLAB/Functions/by Subject) ou a listagem de todas as funções por ordem alfabética (MATLAB/Functions/by Index). A ajuda específica de um determinado comando pode ser feito diretamente no campo abaixo de *Go to MATLAB function*. Além disto, estão disponíveis vários documentos explicativos (*Online Manuals*).

2. Cálculos Científicos

Cálculos Simples

A primeira forma de se utilizar o MATLAB é como uma calculadora científica através da digitação de comandos diretamente no seu *prompt*, a exemplo de

```
» 3*5+12/4
ans =
    18
```

De acordo com os conhecimentos de programação, observa-se que no MATLAB a multiplicação tem precedência sobre a adição. Nota-se também que o resultado da operação foi atribuído à variável *ans*. Uma outra forma de se fazer estes cálculos seria trabalhar com variáveis auxiliares, como mostrado a seguir:

```
» v1=3, v2=5, v3=12, v4=4
v1 =
    3
v2 =
    5
v3 =
   12
v4 =
    4
» resultado=v1*v2+v3/v4
resultado =
    18
```

Primeiro, são criadas quatro variáveis auxiliares (*v1*, *v2*, *v3* e *v4*) e atribuídas a elas os seus valores respectivos. No MATLAB o sinal de igual tem um sentido diferente daquele da Matemática. Aqui, igual significa atribuição. O que estiver à direita do sinal de igual é “colocado” na variável que estiver à esquerda. Finalmente, criamos uma variável chamada *resultado* que recebeu o total da operação.

A vírgula é utilizada para separar os comandos que são dados em uma mesma linha. Esta separação poderia ser feita com ponto e vírgula. Neste caso, o MATLAB não mostra os resultados dos comandos. No exemplo anterior teríamos

```
» v1=3; v2=5; v3=12; v4=4;
» resultado=v1*v2+v3/v4;
```

Em qualquer momento, seria possível verificar o valor que está contido em uma variável, simplesmente digitando no *prompt* o seu nome.

```
» resultado
resultado =
    18
```

O MATLAB oferece as seguintes operações aritméticas:

- >> a+b soma a e b. Por exemplo, 5+6.
- >> a-b subtrai a de b. Por exemplo, 15-12.
- >> a*b multiplica a por b. Por exemplo, 3.14*0.15.
- >> a^b calcula a elevado a b. Por exemplo, 5^(1/2).

A ordem com que são avaliadas as expressões é dada pela seguinte regra: expressões são avaliadas da esquerda para a direita, com a potência tendo a mais alta precedência, seguida pela multiplicação e divisão que têm igual precedência, seguidas pela adição e subtração que têm igual

precedência. Parênteses podem ser usados para alterar esta ordem. Sendo que neste caso, os parênteses mais internos são avaliados antes dos mais externos.

Área de Trabalho

Durante uma sessão, o MATLAB disponibiliza um recurso de recuperação de comandos já introduzidos, utilizando as teclas `?` e `?`. Por exemplo, pressionando a tecla `?` uma vez você obtém o último comando digitado no *prompt*. Pressionando repetidamente a tecla `?` se obtém os comandos digitados anteriormente, um de cada vez na direção para trás. Analogamente, pressionando-se a tecla `?`, mas na direção para frente. Mais ainda, digitando no *prompt* os primeiros caracteres de um comando dado anteriormente e então pressionando-se a tecla `?`, obtém-se o comando mais recente tendo aqueles caracteres iniciais. Em qualquer momento, as teclas `?` e `?` podem ser usadas para se mover o cursor dentro de um comando, no *prompt*. Desta forma um comando pode ser corrigido, além das teclas *Delete* e *Backspace*.

Variáveis

O MATLAB tem certas regras para nomear as variáveis. Os nomes de variáveis devem ser iniciados por letras e não podem conter espaços nem caracteres de pontuação. O MATLAB faz diferença entre letras maiúsculas e minúsculas. Existem alguns nomes de variáveis predefinidas, a saber:

ans - variável usada para os resultados de comandos que não foram atribuídos a nenhuma variável.

pi - número π .

eps - menor número tal que, quando adicionado a 1, cria um número maior que 1 no computador.

flops - armazena o número de operações em ponto flutuante realizadas.

inf - significa infinito.

NaN ou *nan* - significa não é um número, por exemplo, 0/0.

i e *j* - unidade imaginária $\sqrt{-1}$.

nargin - número de argumentos de entrada de uma função.

nargout - número de argumentos de saída de uma função.

realmin - menor número que o computador pode armazenar.

realmax - maior número que o computador pode armazenar.

Os conteúdos das variáveis do tipo texto (*string*) são delimitados através do uso de apóstrofes. No bloco seguinte são definidas duas variáveis *strings* e utilizada uma função de concatenação de texto (verificar outras funções no pacote *matlab\strfun*).

```
» nome='Eduardo'

nome =
    Eduardo

» sobrenome=' Lages'

sobrenome =
    Lages

» nome_completo=strcat(nome,sobrenome)

nome_completo =
    Eduardo Lages
```

As variáveis podem ser redefinidas a qualquer momento, bastando para isso atribuí-las um novo valor.

Para se ter uma listagem das variáveis definidas no espaço de trabalho do MATLAB utilizam-se os comandos *who* (listagem simples) e *whos* (listagem mais detalhada).

As variáveis do espaço de trabalho do MATLAB podem ser excluídas incondicionalmente usando-se o comando *clear*. Por exemplo:

```
>> clear nome_da_variável - exclui somente a variável especificada.
```

```
>> clear nome_da_variável_1 nome_da_variável_2 - exclui as variáveis especificadas.
```

```
>> clear - exclui todas as variáveis do espaço de trabalho.
```

É possível armazenar em arquivo as variáveis criadas no decorrer de uma sessão do MATLAB para recuperá-las em uma outra sessão. Para isto, recorra à opção “*File/Save Workspace As*” da barra de menu (Figura 5). Em seguida, na janela padrão do ambiente *Windows* para definição de arquivo, selecione o diretório e o nome do arquivo (extensão “.mat”) onde serão salvas as variáveis correntes. Para recuperar as variáveis salvas em alguma outra sessão, utilize a opção “*File/Load Workspace*”, que se encontra acima da opção “*File/Save Workspace As*”.

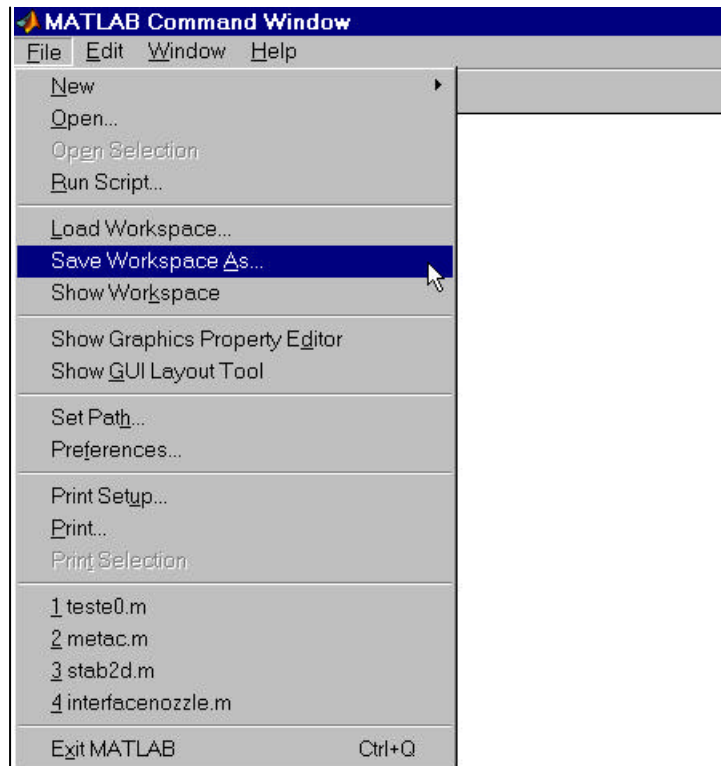


Figura 5 – Localização da opção “*File/Save Workspace As*”.

Funções Científicas

O MATLAB tem uma série de funções científicas predefinidas. A maioria pode ser usada da mesma forma que seria escrita matematicamente. Por exemplo:

```

» co=3
co =
    3
» ca=4
ca =
    4
» h=sqrt(co^2+ca^2)
h =
    5
» a_rad=acos(ca/h)
a_rad =
    0.9273
» a_graus=a_rad*180/pi
a_graus =
    53.1301
  
```

Estes comandos calculam a hipotenusa de um triângulo retângulo e o ângulo entre a hipotenusa e o cateto adjacente, inicialmente em radianos, depois em graus.

Abaixo segue uma lista de funções científicas disponíveis:

abs(x) - valor absoluto de x.
 acos(x) - arco cujo cosseno é x.
 asin(x) - arco cujo seno é x.
 atan(x) - arco cuja tangente é x.
 cos(x) - cosseno de x.
 exp(x) - exponencial e^x .
 gcd(x,y) - máximo divisor comum de x e y.
 lcm(x,y) - mínimo múltiplo comum de x e y.
 log(x) - logaritmo de x na base e.
 log10(x) - logaritmo de x na base 10.
 rem(x,y) - resto da divisão de x por y.
 sin(x) - seno de x.
 sqrt(x) - raiz quadrada de x.
 tan(x) - tangente de x.

Formatos Numéricos

Quando o MATLAB mostra um resultado numérico ele segue certas regras. No caso de nenhum formato estar definido, se um resultado é um número inteiro, o MATLAB mostra como um inteiro. Quando um resultado é um número real, o MATLAB mostra uma aproximação com até quatro casas decimais. Este foi o caso no exemplo do arco cosseno. Se os dígitos significativos estiverem fora desta faixa, o MATLAB mostra o resultado em notação científica. Você pode definir um formato diferente. Abaixo seguem algumas possibilidades:

format short - exibe 4 dígitos.
 format long - exibe 16 dígitos.
 format rat - exibe no formato racional.
 format + - exibe positivo, negativo ou zero.
 format hex - exibe em notação hexadecimal.
 format bank - exibe 2 dígitos.

É importante salientar que o MATLAB não muda a sua forma de representar os números internamente, quando formatos diferentes de exibição de números são escolhidos.

3. Matrizes

Para criar uma variável onde é armazenada uma matriz, basta escrever os elementos da matriz entre colchetes [...], sendo os elementos de uma mesma linha da matriz separados por vírgula (ou espaço em branco) e as linhas separadas por ponto e vírgula. Por exemplo, para armazenar a matriz

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

numa variável de nome A usamos o comando

```
» A=[1,2,3;4,5,6]
```

```
A =
```

```
 1  2  3
 4  5  6
```

```
» A=[1 2 3;4 5 6]
```

```
A =
```

```
 1  2  3
 4  5  6
```


É possível acessar os elementos de uma matriz usando os comandos

```

» A(2,3)

ans =
    6

» A(2,:)

ans =
    4    5    6

» A(:,1)

ans =
    1
    4

» A(:,1:2)

ans =
    1    2
    4    5

```

O primeiro comando foi para mostrar o elemento (1,2) da matriz A. O segundo é para exibir a 2ª linha e o terceiro é para exibir a 1ª coluna. Finalmente, o quarto comando é para exibir a submatriz formada pela 1ª e 2ª colunas da matriz A.

As matrizes podem ser concatenadas

```

» B=[A,[7;8]]

B =
    1    2    3    7
    4    5    6    8

» [A:[7,8,9]]

ans =
    1    2    3
    4    5    6
    7    8    9

```

As operações matriciais são executadas de forma semelhante a que são executadas operações escalares

```

» A=[1,2;3,4]; B=[-3;1]; C=[3,5;-5,2];
» A+C

ans =
    4    7
   -2    6

» 3*A

ans =
    3    6
    9   12

» C*A

ans =
   18   26
    1   -2

```

```

» B*A
??? Error using ==> *
Inner matrix dimensions must agree.

» A^2

ans =
    7   10
   15   22

» (A*C)^2

ans =
   -50  144
  -176  430

```

O MATLAB tem funções que geram matrizes especiais, a saber:

eye(m,n) - gera matriz com m-linhas e n-colunas com valor unitário nos elementos de índices iguais e zero para os demais elementos.
zeros(m,n) - gera matriz nula com m-linhas e n-colunas.
rand(m,n) - gera matriz com m-linhas e n-colunas com elementos randômicos.
hilb(n) - gera matriz de Hilbert de ordem nxn.
magic(n) - gera uma matriz mágica de ordem nxn com soma igual dos elementos das linhas e das colunas.


Abaixo um resumo das operações matriciais

```

>> A=[a11,a12,...,a1n;a21,a22,...; ... ,amn] cria uma matriz, m por n, usando os elementos a11,
a12, ..., amn e a armazena numa variável de nome A.
>> A+B é a soma de A e B,
>> A*B é o produto de A por B,
>> A.' é a transposta de A,
>> A-B é a diferença A menos B,
>> num*A é o produto do escalar num por A,
>> A^ k é a potência A elevado a k.
>> A_j=A(:,j) é a coluna j da matriz A, >> A_i=A(i,:) é a linha i da matriz A.

```

4. Arquivos de Comandos

Um outro modo de se trabalhar com o MATLAB é construir um bloco de comandos para ser executado em conjunto. Para tal, são construídas as *functions* e *scripts* do MATLAB. Os mesmos são arquivos texto com a extensão ".m". Para gerá-los, o usuário pode utilizar qualquer editor de texto, tipo ASCII, podendo ser utilizado o editor embutido no próprio MATLAB (Figura 6). Este editor é acionado através da opção "File/New/M-File" ou através do ícone .

Uma *function* é um *m-file* contendo um conjunto de comandos que para ser executado necessita de parâmetros de entrada e/ou fornece parâmetros de saída. Quando isto acontecer, a primeira linha deste arquivo texto deve atender à seguinte sintaxe:

```
function [variáveis_saída]= nome_da_função (variáveis_entrada)
```

Quando só existe um parâmetro de saída, é permitido omitir os colchetes. Nos quadros abaixo, exemplificam-se os corpos dos arquivos de funções com um e dois parâmetros de saída.

```
function vmedia= minha_media(n1,n2)
% Esta função calcula a média aritmética de dois números.
vmedia=(n1+n2)/2;
```

```
function [vmedia,vdiferença]= minha_media_diferenca(n1,n2)
% Esta função calcula a média aritmética de dois números e a diferença entre eles.
vmedia=(n1+n2)/2;
vdiferenca=n1-n2;
```

Conforme exemplificado nos blocos anteriores, o caracter “%” é utilizado para introduzir comentários no corpo das funções. Recomenda-se o uso de comentários ao longo dos programas para facilitar o entendimento dos seus funcionamentos.

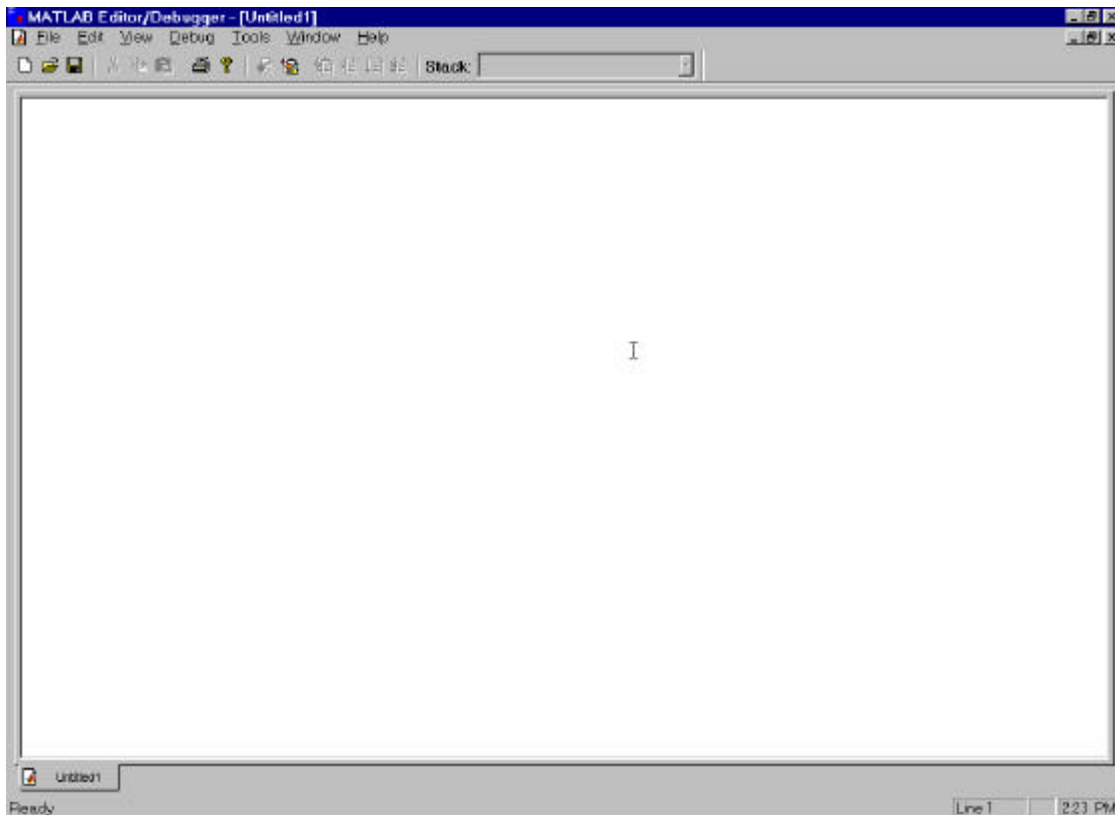



Figura 6 – Janela do editor de texto embutido no MATLAB.

Os arquivos devem ser gravados com os mesmos nomes das funções e com a extensão “.m”, no caso *minha_media.m* e *minha_media_diferenca.m*, respectivamente. Para dispor desta nova função a partir da linha de comandos do MATLAB, é preciso disponibilizar para o MATLAB o diretório onde este arquivo está gravado. Isto é feito através da opção “File/Set Path” ou através do ícone . Neste instante, abre-se uma janela de definição dos diretórios disponíveis para o MATLAB (Figura 7).

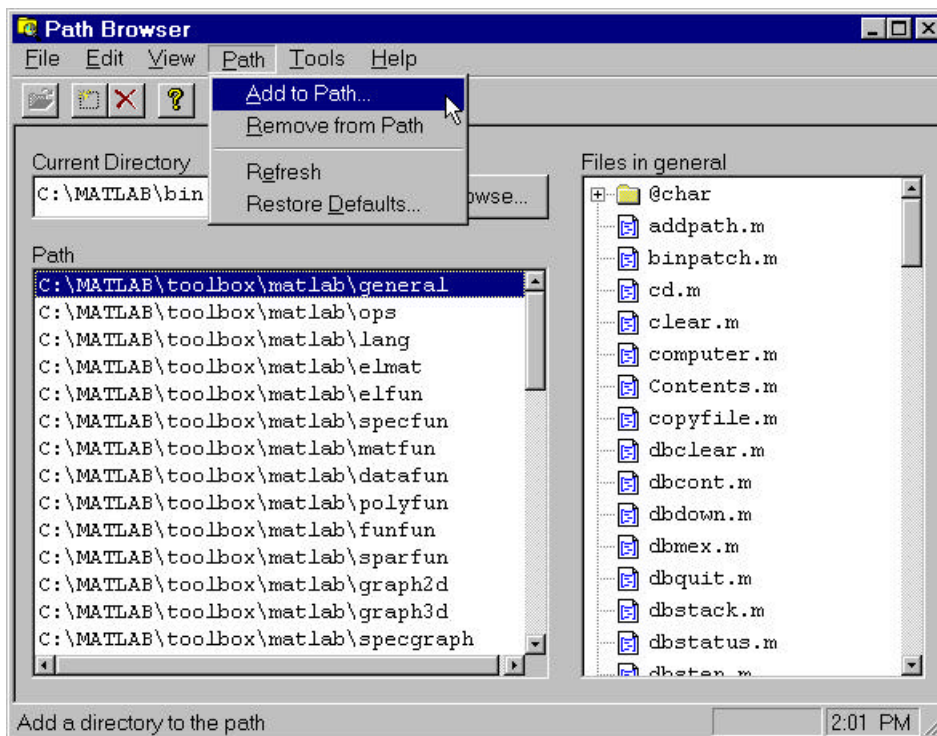


Figura 7 – Janela de definição dos diretórios disponíveis para o MATLAB.

O usuário deve selecionar a opção “Path/Add to Path” que dispara uma nova janela com definição de um novo diretório (Figura 8). No botão com “...”, o usuário vai escolher na árvore de diretórios do disco rígido aquele que será incluído na lista de diretórios disponíveis para o MATLAB. Após confirmar a escolha no botão “OK”, ao fechar a janela “Path Browser” o usuário será consultado

para saber se a inclusão feita deve ou não valer para futuras sessões do MATLAB (Figura 9). Caso o usuário queira disponibilizar o diretório escolhido para futuras sessões, ele deve apertar o botão de “Sim” (“Yes”).



Figura 8 – Janela de seleção de um novo diretório.

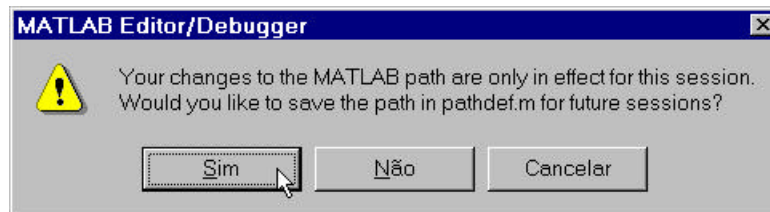


Figura 9 – Janela de confirmação se o diretório escolhido estará disponível em futuras sessões.

Um *script* é um *m-file* sem a declaração da palavra reservada *function* no topo do arquivo. Um *script* se comporta diferente. Imagine que você já dispõe de uma lista de variáveis na sua sessão atual. Suponhamos que *x* é uma dessas variáveis. Quando você escreve um programa com a declaração *function*, e você usa a variável *x* dentro do programa, o programa não usará o valor desta variável da sua sessão (a menos que *x* seja passada como parâmetro de entrada), e sim o valor apropriado do programa. Além disso, a menos que você declare um valor novo para *x* na sua sessão, o programa não mudará o valor de *x* da sessão. Isto é muito cômodo uma vez que você não precisa se preocupar com as variáveis da sessão. Tudo isso acontece por causa da declaração *function*. Se você não faz a declaração de *function*, então podem ser alteradas as variáveis em sua sessão.

5. Referências

- R. Smith (1988) “MATLAB Tutorial”. <http://www.math.ufl.edu/help/matlab-tutorial/index.html#SEC5> (November).
- D. Hanselman & B. Littlefield (1997) “MATLAB – Versão Estudante – Guia do Usuário – Versão 4”. MAKRON Books do Brasil. São Paulo.
- R. J. Santos (1998) “Introdução ao MATLAB”. <http://www.mat.ufmg.br/~regi/topicos/intmatl.html> (Junho).