
INTRODUÇÃO AO MATLAB®

Reginaldo J. Santos
Departamento de Matemática-ICEx
Universidade Federal de Minas Gerais
<http://www.mat.ufmg.br/~regi>

Agosto de 2005
última atualização em
27 de maio de 2009

Sumário

1	Introdução	3
2	Cálculos Científicos	4
	Cálculos Simples	4
2.1	Área de Trabalho	6
2.2	Variáveis	6
2.3	Funções Científicas	7
2.4	Formatos Numéricos	8
3	Variáveis e Expressões Simbólicas	10
4	Instalando o Pacote gaal	15
5	Desenhando Gráficos de Funções	17
6	Matrizes	21
6.1	Exercícios	25
	Referências	26

1 Introdução

O **MATLAB**¹ é um software destinado a fazer cálculos com matrizes (**MATLAB**[®] = **MAT**rix **LAB**oratory). Os comandos do **MATLAB** são muito próximos da forma como escrevemos expressões algébricas, tornando mais simples o seu uso. Podem ser incorporados às rotinas pré-definidas, pacotes para cálculos específicos. Escrevemos um pacote chamado **gaa1** com funções que são direcionadas para um curso de Geometria Analítica e Álgebra Linear.

O programa **MATLAB**[®] pode ser adquirido gratuitamente na compra do Guia do Usuário [1]. Por exemplo, através da internet, na livraria Blackwell's na Inglaterra (<http://bookshop.blackwell.co.uk>), o Guia do Usuário foi adquirido por US\$ 68,00 incluindo a despesa com o transporte, acompanhado de um CD com o programa.

Uma vez inicializado o **MATLAB**[®], aparecerá na janela de comandos um prompt `>>` ou `EDU>>`. O prompt significa que o **MATLAB**[®] está esperando um comando. Todo comando deve ser finalizado teclando-se **Enter**.

No **MATLAB**[®], pode-se obter ajuda sobre qualquer comando ou função. O comando

```
>> help
```

(sem o prompt `>>`) mostra uma listagem de todos os pacotes disponíveis. Ajuda sobre um pacote específico ou sobre um comando ou função específica é obtida com o comando

```
>> help nome,
```

(sem a vírgula) onde **nome** pode ser o nome de um pacote ou o nome de um comando ou função.

¹**MATLAB**[®] é marca registrada de The Mathworks, Inc.

2 Cálculos Científicos

Cálculos Simples

O MATLAB[®] faz cálculos simples e científicos como uma calculadora. Por exemplo, suponha que você vai a uma loja e compra 3 objetos que custam 25 reais cada e 5 objetos que custam 12 reais cada. Quanto custou a sua compra?

No MATLAB[®] você pode resolver este problema de pelo menos duas maneiras. A mais simples é

```
EDU>> 3*25 + 5*12
ans =
    135
```

Observe que no MATLAB[®] a multiplicação tem precedência sobre a adição. Note também que ele chamou o resultado de **ans**.

Alternativamente, você pode usar **variáveis** para armazenar informação.

```
EDU>> q1=3, p1=25, q2=5, p2=12
q1 =
     3
p1 =
    25
q2 =
     5
p2 =
    12
```

```
EDU>> total=q1*p1+q2*p2
total =
    135
```

Primeiro, criamos quatro variáveis, q_1 , p_1 , q_2 e p_2 , atribuindo a elas os seus valores respectivos. Observe que 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 `total` que recebeu o total da compra. Usamos a **vírgula** para separar os comandos que são dados em uma mesma linha. Esta separação poderia ser feita com **ponto e vírgula**. Mas, neste caso o MATLAB[®] não mostra os resultados dos comandos. No exemplo anterior teríamos

```
EDU>> q1=3; p1=25; q2=5; p2=12;
EDU>> total=q1*p1+q2*p2;
```

Em qualquer momento, podemos ver o valor que está contido em uma variável, simplesmente digitando no prompt o seu nome.

```
EDU>> total
total =
    135
```

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

2.1 Área de Trabalho

Comandos que foram dados anteriormente podem ser obtidos novamente usando as teclas \uparrow e \downarrow . Por exemplo, pressionando a tecla \uparrow uma vez você obtém o último comando digitado no prompt. Pressionando repetidamente a tecla \uparrow se obtém os comandos digitados anteriormente, um de cada vez na direção para trás. Analogamente, pressionando-se a tecla \downarrow , mas na direção para frente. Mais ainda, digitando no prompt os primeiros caracteres de um comando dado anteriormente e então pressionando-se a tecla \uparrow , obtém-se o comando mais recente tendo aqueles caracteres iniciais. Em qualquer momento, as teclas \leftarrow , \rightarrow 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**.

2.2 Variáveis

O MATLAB[®] tem certas regras para nomear as variáveis. Os nomes de variáveis devem ser nomes iniciados por letras não podem conter espaços nem caracteres de pontuação. O MATLAB[®] faz diferença entre letras maiúsculas e minúsculas. Alguns nomes são usados para variáveis predefinidas. Estas são:

`ans` - variável usada para os resultados.

`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.

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

2.3 Funções Científicas

O MATLAB[®] tem uma série de funções científicas pré-definidas. A maioria pode ser usada da mesma forma que seria escrita matematicamente. Por exemplo:

```
EDU>> x=sqrt(2)/2
```

```
x =
```

```
    0.7071
```

```
EDU>> y=acos(x)
```

```
y =
```

```
    0.7854
```

```
EDU>> y_graus=y*180/pi
y_graus =
    45.0000
```

Estes comandos calculam o arco cujo cosseno é $\sqrt{2}/2$, 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`.

2.4 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. Este foi o caso no exemplo das compras, que fizemos anteriormente. 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, que fizemos anteriormente. 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 5 dígitos.

`format long` - exibe 16 dígitos.

`format rat` - exibe no formato racional.

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 Variáveis e Expressões Simbólicas

Agora, vamos ver como podemos manipular com expressões que além de números e variáveis numéricas, contém também variáveis simbólicas. Por exemplo:

```
EDU>> syms x
EDU>> simplify((sin(x))^2+(cos(x))^2)
ans =
1
```

Estes comandos mandam o MATLAB[®] simplificar a expressão $\sin^2 x + \cos^2 x$. Primeiro precisamos dizer ao MATLAB[®] que x é uma variável simbólica, depois pedimos para simplificar a expressão que envolve x . Neste caso usamos uma **função** chamada `simplify`. A palavra função no MATLAB[®] tem um significado diferente daquele que tem na Matemática. Aqui função é um comando, que pode ter alguns argumentos de entrada e alguns de saída. Neste caso, a função `simplify` tem como argumento de entrada uma expressão simbólica e de saída também.

Uma vez definido que a variável x é uma variável simbólica, podemos definir expressões que envolvem esta variável. Por exemplo, dadas duas funções

$$f(x) = 2x^2 + 3x - 5 \quad \text{e} \quad g(x) = x^2 - x + 7,$$

podemos fazer uma série de operações algébricas envolvendo estas funções.

```
EDU>> f=2*x^2+3*x-5; g=x^2-x+7;
EDU>> f+g
ans =
3*x^2+2*x+2
EDU>> f-g
```

```

ans =
x^2+4*x-12
EDU>> f*g
ans =
(2*x^2+3*x-5)*(x^2-x+7)
EDU>> expand(ans)
ans =
2*x^4+x^3+6*x^2+26*x-35
EDU>> f/g
ans =
(2*x^2+3*x-5)/(x^2-x+7)
EDU>> expand(ans)

ans =

2/(x^2-x+7)*x^2+3/(x^2-x+7)*x-5/(x^2-x+7)

```

```
EDU>> pretty(ans)
```

$$2 \frac{x^2}{x^2 - x + 7} + 3 \frac{x}{x^2 - x + 7} - \frac{5}{x^2 - x + 7}$$

```
EDU>> f^3
```

```
ans =  
(2*x^2+3*x-5)^3  
EDU>> expand(ans)  
ans =  
8*x^6+36*x^5-6*x^4-153*x^3+15*x^2+225*x-125
```

Observe que o MATLAB[®] não faz as simplificações ou expansões automaticamente. Para isso, usamos os comandos `simplify` que simplifica e `expand` que faz a expansão. Além destes, usamos acima também o comando `pretty`, que mostra a expressão de uma forma mais fácil de enxergar. Além destes, que já vimos, existe o comando `simple`, que tenta encontrar a forma mais simples de escrever uma expressão.

O MATLAB[®] pode realizar operações mais avançadas sobre expressões simbólicas. A função `compose` calcula a composição das funções $f(x)$ e $g(x)$ em $f(g(x))$, a função `finverse` encontra a inversa funcional de uma expressão e a função `subs` substitui uma variável por um número (ou por outra variável) em uma expressão. Por exemplo:

```
EDU>> f=1/(1-x^2); g=sin(x);  
EDU>> compose(f,g)  
ans = 1/(1-sin(x)^2)  
EDU>> compose(g,f)  
ans = sin(1/(1-x^2))  
EDU>> finverse(g)  
ans = asin(x)  
EDU>> subs(f,x,2)  
ans = -1/3
```

O MATLAB[®] pode resolver equações. Por exemplo, para resolver a equação

$$ax^2 + bx + c = 0,$$

algebricamente, podemos usar os comandos:

```
EDU>> syms a b c x
EDU>> solve(a*x^2+b*x+c)
ans =
[1/2/a*(-b+(b^2-4*a*c)^(1/2))]
[1/2/a*(-b-(b^2-4*a*c)^(1/2))]
```

O MATLAB[®] pode exibir este resultado de uma forma mais fácil de enxergar usando a função `pretty`.

```
EDU>> pretty(ans)
```

$$\begin{array}{l} [\qquad \qquad \qquad 2 \qquad \qquad \qquad 1/2] \\ [\quad -b + (b^2 - 4 a c) \quad] \\ [1/2 \text{ -----}] \\ [\qquad \qquad \qquad a \qquad \qquad \qquad] \\ [\qquad \qquad \qquad] \\ [\qquad \qquad \qquad 2 \qquad \qquad \qquad 1/2] \\ [\quad -b - (b^2 - 4 a c) \quad] \\ [1/2 \text{ -----}] \\ [\qquad \qquad \qquad a \qquad \qquad \qquad] \end{array}$$

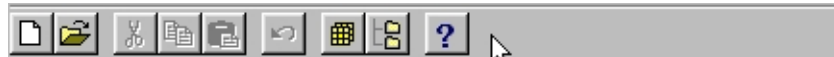
Abaixo segue um resumo das funções para manipulação de expressões algébricas:

`diff(f)` - calcula a derivada de `f`.
`compose(f,g)` - determina a composta $f(g(x))$.
`expand(expr)` - expande uma expressão `expr`.
`finverse(expr)` - determina a inversa funcional da expressão `expr`.
`pretty(expr)` - exibe a expressão `expr` numa forma mais bonita.
`simple` - procura encontrar uma forma mais simples de escrever uma expressão `expr`.
`simplify(expr)` - simplifica a expressão `expr`.
`solve(expr)` - acha a(s) solução(es) da equação `expr=0`.
`subs(expr,x,a)` - substitui na expressão `expr` a variável `x` por `a`.
`syms x y z a b` - define as variáveis simbólicas `x`, `y`, `z`, `a` e `b`.

Existem várias outras funções para manipulação de expressões algébricas. Você pode obter informações sobre elas digitando `help symbolic`. Uma função interessante que mostra as capacidades do MATLAB[®] em tratar com funções matemáticas é `funtool` que é uma calculadora para funções.

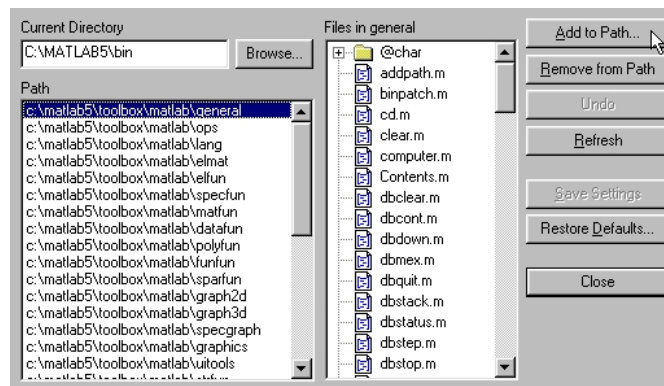
4 Instalando o Pacote gaal

1. Faça o “download” do pacote gaal que está no site <http://www.mat.ufmg.br/~regi>.
2. Extraia os arquivos em `MATLAB\toolbox`. Deve ser criada uma pasta aí com nome `gaal`.
3. Inicialize o MATLAB®, se já não o tiver feito.
4. Na barra de ferramentas

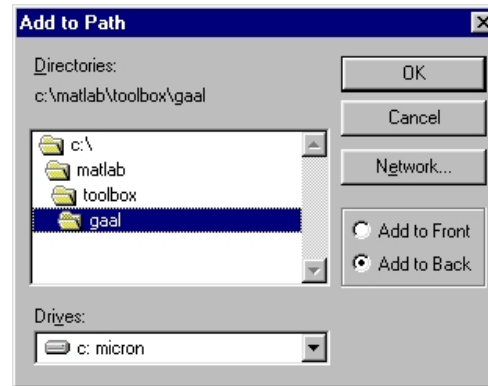


clique com o botão esquerdo do mouse no botão 

5. Clique no botão **Add to Path...**



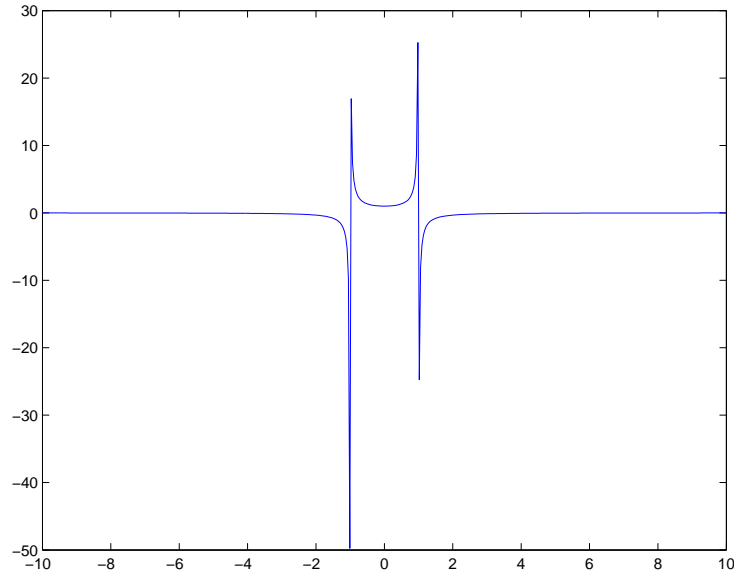
Clique duas vezes com o botão esquerdo do mouse em `toolbox` e depois o mesmo em `gaal`. Depois clique em **Add to Back**



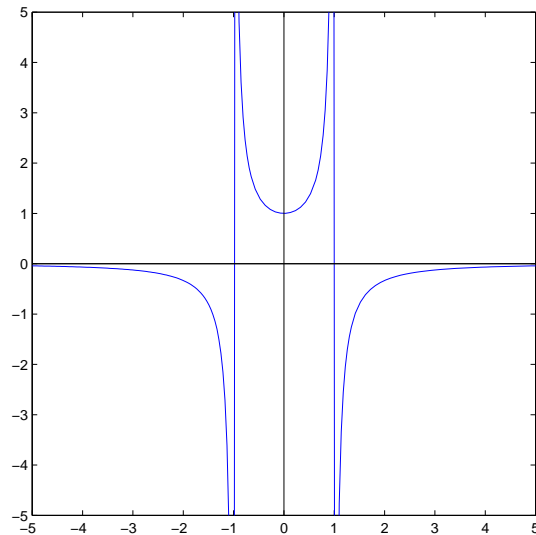
6. Clique em **OK**. Depois, em **Save Settings** e por último em **Close**
7. Verifique se o MATLAB[®] adicionou o pacote `gaal` aos outros, digitando no prompt `winhelp`. Ele deve aparecer na última linha ou na primeira. Caso contrário repita o processo acima, com mais cuidado.
8. Para informações sobre o pacote `gaal` digite no prompt do MATLAB[®]
`help gaal`

5 Desenhando Gráficos de Funções

Para desenhar o gráfico de uma função de uma variável, existe no pacote `gaal` a função `plotf1` (use `help plotf1` para saber seu uso). Para colocar os eixos coordenados temos no pacote `gaal` a função `eixos`. Usando a função $f(x) = 1/(1 - x^2)$ que definimos acima temos:
EDU>> `plotf1(f, [-10,10],200)`

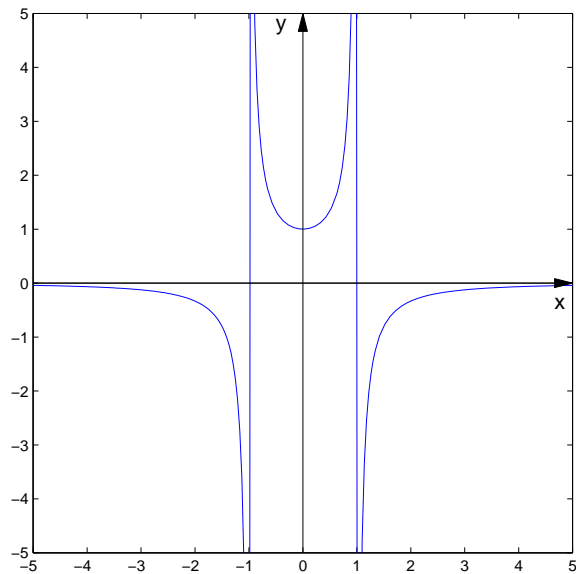


EDU>> `eixos`



Neste caso como os eixos já estavam traçados, o resultado não foi bom. Então, vamos limpar a figura com o comando `clf` e com a seta para cima ↑ vamos recuperar comandos que havíamos digitado anteriormente.

```
EDU>> clf
EDU>> plotf1(f, [-10,10])
EDU>> axis([-5,5,-5,5])
EDU>> eixos
```



6 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 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]
```

Podemos acessar os elementos de uma matriz usando os comandos

```
EDU>> A(2,3)
```

```
ans =
```

```
6
```

```
EDU>> A(2,:) 
```

```
ans =
```

```
4      5      6
```

```
EDU>> A(:,1)
```

```
ans =
```

```
1
```

```
4
```

```
EDU>> 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

```
EDU>> B=[A, [7;8]]
```

```
B =
```

```
    1    2    3    7
    4    5    6    8
```

```
EDU>> [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

```
EDU>> A=[1,2;3,4]; B=[-3;1]; C=[3,5;-5,2];
```

```
EDU>> A+C
```

```
ans =
```

```
    4    7
   -2    6
```

```
EDU>> 3*A
```

```
ans =
```

```
    3    6
    9   12
```

```
EDU>> C*A
ans =
    18    26
     1    -2
EDU>> B*A
??? Error using ==> *
Inner matrix dimensions must agree.
EDU>> A^2
ans =
     7    10
    15    22
EDU>> (A*C)^2
ans =
   -50   144
  -176   430
```

O MATLAB[®] tem funções que geram matrizes especiais

```
EDU>> I=eye(3)
I =
     1     0     0
     0     1     0
     0     0     1
EDU>> O=zeros(3,1)
O =
     0
```

0
0

Além destas, o pacote `gaal` oferece as funções `randi` para gerar uma matriz com elementos inteiros aleatórios e `matvand` que cria a matriz de Vandermonde.

```
EDU>> A=randi(3)
```

A =

```
    0    3    2
    2    1    0
   -2   -1   -1
```

```
EDU>> B=randi(3,1)
```

B =

```
    4
    2
   -3
```

```
EDU>> C=matvand(B,2)
```

C =

```
   16    4    1
    4    2    1
    9   -3    1
```

Abaixo um resumo das operações matriciais

`>> A=[a11,a12,...,a1n;a21,a22,...; ...,amn]` cria uma matriz, m por n , usando os elementos a_{11} , a_{12} , ..., a_{mn} e a armazena numa variável de nome A.

`>> A+B` é a soma de A e B,

`>> A-B` é a diferença A menos B,

`>> A*B` é o produto de A por B,

`>> num*A` é o produto do escalar `num` por A,

`>> A.'` é a transposta de 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 .

6.1 Exercícios

1. Sejam

$$A = \begin{bmatrix} 1 & -3 & 0 \\ 0 & 4 & -2 \end{bmatrix} \quad \text{e} \quad X = \begin{bmatrix} 3 \\ 2 \\ 5 \end{bmatrix}.$$

Verifique que $AX = 3A_1 + 2A_2 + 5A_3$, onde A_j é a j -ésima coluna de A , para $j = 1, 2, 3$.

2. Encontre um valor de x tal que $AB^t = 0$, onde

$$A = \begin{bmatrix} x & 4 & -2 \end{bmatrix} \quad \text{e} \quad B = \begin{bmatrix} 2 & -3 & 5 \end{bmatrix}.$$

3. Use o MATLAB[®] para encontrar o menor inteiro $k > 1$ tal que $A^k = A$, onde

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix};$$

4. Use o MATLAB[®] para calcular os membros da seqüência $A, A^2, A^3, \dots, A^k, \dots$, para

(a)

$$A = \begin{bmatrix} 1 & 1/2 \\ 0 & 1/3 \end{bmatrix};$$

(b)

$$A = \begin{bmatrix} 1/2 & 1/3 \\ 0 & -1/5 \end{bmatrix}.$$

A seqüência parece estar convergindo para alguma matriz? Se estiver, para qual?

Referências

- [1] Mathworks Inc. *Student Edition of MATLAB Version 5 for Windows*. Prentice Hall, Upper Saddle River, New Jersey, 1997.