Relatório Estudo Dirigido – Matlab

Nome: Antonio Carlos Neto

Matrícula: 11611BCC054

1. Introdução. Após ler a introdução, relatar a principal contribuição do Matlab

R. O Matlab contribui principalmente com operações com matrizes, álgebra matricial. Entretanto, hoje em dia, fornece gráficos tridimensionais, interfaces gráficas de usuário, interação com outros softwares dentre outros.

1. Desktop. Leia o capítulo e busque quais os comandos equivalentes (dica: comandos semelhantes ao do SO).
   1. Mostrar pasta atual

R. O comando é “pwd”.

* 1. Mostrar arquivos (ls e dir). Relatar diferenças entre os dois comandos

R. O comando “ls” trás uma interface gráfica melhor, facilitando a visualização, porém o comando “dir” pode ser atribuído em uma variável, criando uma struct, facilitando a interação com a programação.

* 1. Mudar para uma subpasta

R. O comando é “cd nomedasubpasta”.

* 1. Voltar para a pasta anterior

R. O comando é “cd ..”.

* 1. Busque por ajudar por um comando, utilizando a linha de comando (ex: help max)

R. O comando de ajuda é “help nomecomando”.

1. Conceitos básicos. Execute TODOS os comandos e faça uma breve descrição utilizando a tabela abaixo. NÃO utilizar os mesmos exemplos do livro.

|  |  |  |
| --- | --- | --- |
| Comando | Saída | Descrição |
| 100/20/10 | ans = 0.5 | Operações matemáticas podem ser realizada no prompt |
| 100/(20/10) | ans = 50 | Teste de prioridade de operadores |
| 40 - (40\*2) + (20/4) | ans = -35 | Na incerteza, melhor usar parênteses |
| a = [1 2 3; 4 5 6; 7 8 9 ] | a =  1 2 3  4 5 6  7 8 9 | Podemos armazenar uma matriz em uma variável |
| sqrt(9), 3^2 | ans = 3  ans = 9 | O Matlab faz operações de exponenciação e a vírgula separa os comandos |
| log(6) | ans = 1.7918 | É necessário o uso de parênteses no log e ele considera como padrão log na base e, número de euler. |
| floor(2.5) | ans = 2 | Arredonda o número para baixo, ou seja, faz chão do número |
| exp(9) | ans = 8103.1 | Utiliza o número de euler como base |
| log10(10) | ans = 1 | Para fazer log na base 10 deve usar esse comando, sendo que não é o padrão. |
| abs(-7.855454) | ans = 7.8555 | A função abs é basicamente módulo, porém se o número for float, há um padrão de arredondamento |
| round(6.5) | ans = 7 | Arredonda o número para um inteiro, se o primeiro número depois do ponto for < 5 há um arredondamento para baixo e se for >=5 para cima. |
| ceil(5.2) | ans = 6 | Arredonda o número para cima, ou seja, função teto |
| fix(3.4) | ans = 3 | Arredonda o número em direção ao zero, ou simplesmente retira a parte decimal |
| sign(-6) | ans = -1 | Existe 3 retornos, -1 se o número for negativo, 0 se o número for zero, 1 se o número for positivo. |
| clc |  | Limpa a tela de comandos |
| clear a b |  | Destrói as variáveis, sendo que se o comando é somente “clear” ele destrói todas variáveis, se é seguido de variáveis separado por espaço, ele destrói somente aquelas variáveis |
| d = 5; format short g, d | d = 5 | Altera as propriedades de exposição, nesse exemplo ele alterou o espaçamento. |
| ↑ |  | Busca no histórico o comando mais recente. |

1. Matrizes. Idem ao enunciado do exercício anterior

|  |  |  |
| --- | --- | --- |
| Comando | Saída | Descrição |
| a = [1 2 3; 4 5 6; 72 83 92] | a=  1 2 3  4 5 6  72 83 92 | Criação de uma matriz. Novas linhas separadas por; |
| 1:2:10 | ans =  1 3 5 7 9 | Primeiro argumento é o primeiro termo de uma matriz de uma linha, o segundo termo é a razão dos termos, e o terceiro termo é até onde irá essa PA. |
| linspace(2,8,3) | ans =  2 5 8 | Cria uma matriz de uma linha. Primeiro termo é o inicial, segundo termo é o final, e o terceiro termo é a quantidade de termos. Assim a matriz cria esse vetor de forma proporcional. |
| zeros(3,4) | ans =  0 0 0 0  0 0 0 0  0 0 0 0 | Cria uma matriz de zeros, do tamanho passado, sendo linha primeiro e colunas depois. |
| ones(3,4) | ans =  1 1 1 1  1 1 1 1  1 1 1 1 | Cria uma matriz de uns, do tamanho passado, sendo a linha o primeiro argumento e colunas o segundo. |
| eye(5) | ans =  Diagonal Matrix  1 0 0 0 0  0 1 0 0 0  0 0 1 0 0  0 0 0 1 0  0 0 0 0 1 | Cria uma matriz identidade de tamanho n. |
| rand(3,4) | ans =  0.536162 0.145642 0.871480 0.988243  0.579917 0.169475 0.351998 0.137536  0.990526 0.135945 0.041194 0.643372 | Cria uma matriz de números randômicos, entre 0 e 1, sendo o tamanho da matriz o argumentos passados. |
| randn(3,4) | ans =  -0.16738 1.17274 0.42916 -0.44910  -1.00275 0.27063 -1.20769 -0.31439  -0.84152 -1.58518 -0.38425 -0.50590 | Cria uma matriz de tamanho igual aos argumentos com números randômicos entre -3 e 3 |
| >>A = [1 2; 3 4]  A =  1 2  3 4  >> repmat(A,2,3) | ans =  1 2 1 2 1 2  3 4 3 4 3 4  1 2 1 2 1 2  3 4 3 4 3 4 | Cria uma matriz com base em outra matriz, repetindo ela, sendo o primeiro argumento a matriz o segundo argumento quantas vezes ele vai duplicar a matriz em linhas e o terceiro a mesma coisa em colunas |
| >>A = [1 2; 3 4]  A =  1 2  3 4  >> B = [5 6 10; 7 8 9]  B =  5 6 10  7 8 9  >> [A B] | ans =  1 2 5 6 10  3 4 7 8 9 | Cria uma matriz dado duas outras, sendo que a resultante é a concatenação das duas. Lembrando que o número de linhas das duas deve ser a mesma. |
| >>A = [1 2; 3 4]  A =  1 2  3 4  >> C = ones(2)  C =  1 1  1 1  >> [A ; C] | ans =  1 2  3 4  1 1  1 1 | Cria uma matriz dado duas outras, sendo que a resultante é a concatenação das duas verticalmente. Lembrando que o número de colunas deve ser a mesa; |
| >>A = [1 2; 3 4]  A =  1 2  3 4  >> diag(A) | ans =  1  4 | Pega a diagonal de uma matriz. |
| >> A = [1 2; 3 4]  A =  1 2  3 4  >> find(A>2) | ans =  2  4 | Verifica número por número qual é maior que um determinado número. |
| >> A = [1 2; 3 4]  A =  1 2  3 4  >> size(A) | ans =  2 2 | Retorna o tamanho da matriz. |
| >> A = [1 2; 3 4]  A =  1 2  3 4  >> transpose(A) | ans =  1 3  2 4 | Retorna uma matriz transpoSta. |
| >> A = [4 9 3; 1 5 6; 2 0 7]  A =  4 9 3  1 5 6  2 0 7  >> sort(A) | ᐀ans =  1 0 3  2 5 6  4 9 7 | Ordena as colunas da matriz, de forma crescente. |
| >> A = [4 9 3; 1 5 6; 2 0 7]  A =  4 9 3  1 5 6  2 0 7  >> sortrows(A,2) | ans =  2 0 7  1 5 6  4 9 3 | Ordena a coluna da matriz escolhida, de forma crescente . |
|  |  |  |

1. Operações Matemáticas com matrizes

|  |  |  |
| --- | --- | --- |
| Comando | Saída | Descrição |
| >> A = ones(2)  A =  1 1  1 1  >> B = eye(2)  B =  Diagonal Matrix  1 0  0 1  >> A.\*B | ans =  1 0  0 1 | Multiplicação de matrizes elemento por elemento. |
| >> A = [10 8; 6 4]  A =  10 8  6 4  >> B = [5 4; 3 1]  B =  5 4  3 1  >> A./B | ans=  2 2  2 4 | Divisão de matrizes elemento por elemento. |
| >> A  A =  10 8  6 4  >> sqrt (A) | ans =  3.1623 2.8284  2.4495 2.0000 | Raiz quadrada elemento por elemento. |
| >> A  A =  10 8  6 4  >> B  B =  5 4  3 1  >> A>B | ans =  1 1  1 1 | Retorna uma matriz de 0s e 1s, sendo que 0 significa que o elemento de A naquela posição não é maior que o elemento de B naquela posição e 1 o contrário. |
| >> A = [10 1; 3 0; 2 5]  A =  10 1  3 0  2 5  >> min(A) | ans =  2 0 | Retorna uma matriz de uma linha com o menor elemento de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> max(A) | ans =  10 5 | Retorna uma matriz de uma linha com o maior elemento de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> mean(A) | ans =  5 2 | Retorna uma matriz de uma linha com a média de cada coluna. |
| A =  10 1  3 0  2 5  >> median(A) | ans =  3 1 | Retorna uma matriz de uma linha com a mediana de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> std(A) | ans =  4.3589 2.6458 | Retorna uma matriz de uma linha com o desvio padrão de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> var(A) | ans =  19 7 | Retorna uma matriz de uma linha com a variância de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> sum(A) | ans =  15 6 | Retorna uma matriz de uma linha com a soma de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> cumsum(A) | ans =  10 1  13 1  15 6 | Retorna uma matriz de tamanho igual a do input, a cada linha mostra a soma das linhas anteriores, coluna por coluna. |
| >> A  A =  10 1  3 0  2 5  >> prod(A) | ans =  60 0 | Retorna uma matriz de uma linha com o produto de cada coluna. |
| >> A  A =  10 1  3 0  2 5  >> cumprod(A) | ans =  10 1  30 0  60 0 | Retorna uma matriz de tamanho igual a do input, a cada linha mostra o produto das linhas anteriores, coluna por coluna. |
| >> A  A =  10 1  3 0  2 5  >> diff(A) | ans =  -7 -1  -1 5 | Retorna uma matriz de tamanho quase igual a do input, sendo que possui uma linha menos, cada linha mostra a subtração da linha anterior com a atual, coluna por coluna. A primeira linha não existe pois não existe linha anterior. |
| >> A  A =  10 1  3 0  2 5  >> cov(A) | ans =  19 -5  -5 7 | Calcula a matriz de covariância, assumindo que cada coluna em A representa os resultados de uma variável.(COPIADO DO LIVRO) |
| >> A  A =  10 1  3 0  2 5  >> corrcoef(A) | ans =  1.00000 -0.43355  -0.43355 1.00000 | Calcula uma matriz de coeficientes de correlação, assumindo que cada coluna em X representa os resultados de uma variável.(COPIADO DO LIVRO) |
| >> A = [2 5; 3 7]  A =  2 5  3 7  >> inv(A) | ans =  -7.0000 5.0000  3.0000 -2.0000 | Calcula matriz inversa. |
| >> A = [2 5; 3 7]  A =  2 5  3 7  >> A^-1 | ans =  -7.0000 5.0000  3.0000 -2.0000 | Calcula matriz inversa. |
| >> A = [4 9 3; 1 5 6; 2 0 7]  A =  4 9 3  1 5 6  2 0 7  >> norm(A) | ans = 13.222 | Retorna a normal da matriz. |
| >> A = [1, 2 ; 3, 4]  A =  1 2  3 4  >> det(A) | ans = -2 | Retorna o determinante. |
| >> A = [1, 2 ; 3, 4]  A =  1 2  3 4  >> trace(A) | ans = 5 | Retorna a soma de todos elementos da matriz principal. |

1. Dados

OBS: Instalar o pacote io, baixado no site <https://octave.sourceforge.io/io/index.html>, e instalar pela linha de comando no octave, pkg install io-2.4.11.tar.gz, pkg load io. Lembrando de estar na pasta do arquivo baixado.

|  |  |  |
| --- | --- | --- |
| Comando | Saída | Descrição |
| >> importedData = load('dados.txt');  >> importedData | importedData =  1 2 3  4 5 6  7 8 9  10 11 12 | Carrega um arquivo txt para uma matriz. |
| data = xlsread('dadosx.xlsx') | data =  1 2 3  4 5 6  7 8 9 | Carrega um arquivo xlsx para uma matriz. |
|  |  |  |

1. Gráficos. Fazer exemplos semelhantes e mostrar os gráficos
2. Scripts
3. Funções
4. Fluxo