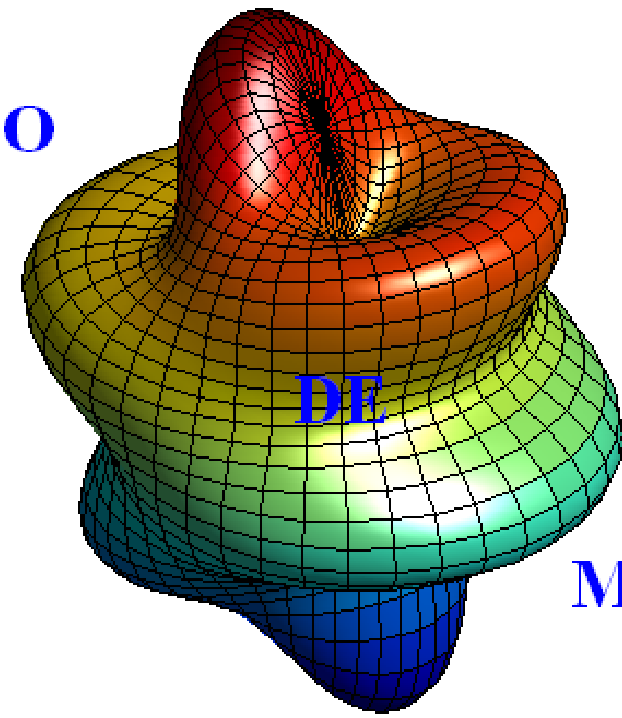
UNIVERSIDADE FEDERAL DO PARÁ
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
PROGRAMA DE EDUCAÇÃO TUTORIAL
SEMANA DOS 40 ANOS DE ENGENHARIA ELÉTRICA

CURSO DE MATLAB



NÍVEL BÁSICO
CAPÍTULO II

P.E.T. E.E.

PROGRAMA DE EDUCAÇÃO TUTORIAL
ENGENHARIA ELÉTRICA -UFPA
BELÉM-2004

ÍNDICE:

CAPÍTULO 2 – MATRIZES , POLINÔMIOS E FUNÇÕES ESPECIAIS

2.1- MATRIZES

2.1.1- Criação de Matrizes Especiais

- **Magic Square -** Uma matriz **magic square** de ordem n é uma matriz n x n constituída de números inteiros de 1 a n^2. Os elementos a_{ij} da matriz estão dispostos de forma tal que o somatório de cada linha é igual ao somatório de uma coluna.

<u>**Sintaxe:**</u> **magic (n) -** matriz **square magic** de ordem n.
<u>**Exemplo:**</u> **magic(3),magic(2),magic(4)**

```
ans =     8    1    6
          3    5    7
          4    9    2
```

- **Zeros -** Esta função gera uma matriz cujos elemento a_{ij} são nulos.

<u>**Sintaxe**</u>: **zeros(n) g**era uma matriz zero, quadrada, de ordem n; **zeros(m,n)** gera uma matriz zero de ordem m x n.
<u>**Exemplo:**</u> **zeros(2),zeros(3),zeros(2,3)**

- **Ones -** A função ones gera uma matriz cujo valor dos elementos a_{ij} é unitário.

<u>Sintaxe</u>: ones(n) gera uma matriz quadrada de ordem n; **ones(m,n)** gera uma matriz de ordem m x n.
<u>Exemplo:</u> ones(2), ones(4),ones(4,5)

- **Eye -** A matriz identidade pode ser gerada pelo MATLAB através da função **eye**. Uma matriz identidade é uma matriz escalar de qualquer ordem cujos elementos a_{ij} são iguais a 1 para i = j.

<u>Sintaxe</u>: **eye(n)** gera uma matriz identidade de ordem n. Já o formato **eye (m,n)** gera uma matriz de ordem m x n .
<u>Exemplo:</u> eye(3),eye(4),eye(2,3)

- **Pascal -** Cria uma matriz cujas diagonais lembram o triângulo de Pascal. Assim, se usarmos o comando **pascal(4)**, a seguinte matriz é gerada:

1	**1**	**1**	**1**
1	**2**	**3**	**4**
1	**3**	**6**	**10**
1	**4**	**10**	**20**

- **Rand** – Cria matriz de números pseudo-aleatórios uniformemente distribuídos entre 0 e 1.

<u>Sintaxe:</u> rand(n), rand(n,m)
<u>Exemplo:</u> rand(2),rand(3,2)

2.1.2- Manipulação com Matrizes

- **Transposta -** A transposta de uma matriz é uma nova matriz onde as colunas são formadas pelas linhas da matriz original.

<u>Sintaxe:</u> A'
<u>Exemplo</u>: x = magic(3) , x'; y = rand(4) , y'

- **Matriz Inversa -** Por definição o inverso de uma matriz quadrada A é a matriz A^{-1}.

<u>Sintaxe:</u> inv(A).
<u>Exemplo:</u> inv(x), inv(y), inv(x'), inv(y')

- **Determinante -** No MATLAB, o comando utilizado para se achar o determinante de uma matriz é **det(A)**.

<u>Exemplo:</u> det(x), det(y), det(inv(x),det(inv(rand(3)))

- **Comando rot90 -** Uma matriz A pode sofrer uma rotação de 90° usando-se o comando **rot90**.

<u>Exemplo</u> : B = rot90(x) ,rot90(y), C = rot90(A,2)

- **Comando fliplr-** Esse comando troca o lado esquerdo com o direito de uma matriz.

<u>Exemplo:</u> fliplr(x), fliplr(y)

- **Comando flipud -** Esse comando troca a parte de cima com a parte de baixo de uma matriz.

<u>Exemplo</u> : **flipud(x),flipud(y),flipud(inv(x'))**

- **Comando diag -** Esse comando extrai os elementos da diagonal principal da matriz **x** e os coloca em um vetor coluna. Desta forma, temos:

<u>Exemplo:</u> **diag(x), diag(y), diag(inv(rand(3)))**

- **Comando triu -** Este comando trata uma matriz preenchendo com zeros nos lugares dos antigos elementos localizados abaixo da diagonal principal.

<u>Exemplo</u> **: triu (x),triu(y), triu(rand(2))**

2.1.3- Operações com Matrizes

2.1.3.1- Adição e Subtração

Matrizes de mesma dimensão podem ser somadas ou subtraídas. Considere as seguintes matrizes **A** e **B:**

A = [2 3;4 5;6 7] e **B = [1 0;2 3;0 4]**

Para a realização da soma dessas duas matrizes utiliza-se o seguinte comando: **C = A+ B** , e para a realização da subtração utiliza-se este comando: **C = A-B.**

Exemplo: y = A -1 , y = B + 3 – C

2.1.3.2- Multiplicação

A multiplicação de duas matrizes corresponde ao somatório de produtos das linhas i da primeira matriz e das colunas j da segunda matriz. Como o somatório de produtos requer que os vetores tenham o mesmo número de elementos, então o número de colunas de A deve ser igual ao número de linhas de B.

No MATLAB podem ser usados os seguintes comandos:

A = [2 5 1;0 3 –1];
B = [1 0 2;-1 4 –2;5 2 1];
C = A * B

Exemplo: x = [1;2;3] , y = [4;5;6] , x' * y , x*y', y*x' , 5*x , 3*y

2.1.3.3- Exponenciação Matricial

A exponenciação de uma matriz **A**, de dimensão n x n, é uma matriz n x n obtida através da função **expm(A)**. Isto é:

$$\mathbf{expm(A) = 1 + A + A^2/2! + A^3/3! + \dots}$$

2.1.3.4- Obtenção dos Quadrados dos Elementos de uma Matriz

Para uma matriz A, o comando A .^ produz uma nova matriz, cujos novos elementos são o quadrado de seus elementos correspondentes em A.

Exemplo: A = [1 2 ; 3 4] , B = [1+j 2-2j; 3 + 4j 5 –j] , A .^ 2 , B . ^2

2.1.3.5- Multiplicação e Divisão de Arranjos

A multiplicação de arranjos elementos a elemento é identificada pelo símbolo " .* ". Se **x** e **y** tiverem a mesma dimensão, então, a execução do comando **x .*y** resulta no arranjo cujos elementos são o produto de cada um dos elementos individuais de **x** e **y**.
Exemplo: x = [1 2 3] , y = [4 5 6] , z = x.*y

2.1.3.6- Autovalores

Se **A** é uma matriz nxn, então os n números λ que satisfazem a relação $\mathbf{Ax = \lambda x}$ **,** são os autovalores de **A.** Eles podem ser determinados usando-se o comando **eig(A)** que retorna os autovalores sob a forma de um vetor coluna.
Exemplo: A = [0 1 ; -1 0] , eig (A)

2.2- ANÁLISE POLINOMIAL

2.2.1- Representação

Como exemplo vamos tomar o seguinte polinômio:
$\mathbf{f(x) = 3x^4 - 0.5x^3 + x - 5.2}$
Se x assumir valores escalares, podemos escrever:
f(x) = 3*x ^4 - 0.5*x ^3 + x - 5.2;
Se x for um vetor ou uma matriz devemos escrever:
f(x) = 3* x .^4 - 0.5* x .^3 + x - 5.2;
onde o tamanho da matriz f será o mesmo da matriz x.

2.2.2- Equação Característica

As raízes da equação característica são idênticas aos autovalores da matriz **A** . A equação característica da matriz **A** é calculada através de **p = poly(A) ,**
Por exemplo, se uma matriz **A** for igual a **[0 1 0;0 0 1;-6 -11 -6]** então o comando **poly(A)** conduz a: **1 6 11 6.**

2.2.3- Raízes de Polinômios

Achar as raízes de um polinômio, isto é, os valores para os quais o polinômio é igual a zero, é um problema comum em muitas áreas do conhecimento, como por exemplo, achar as raízes de equações que regem o desempenho de um sistema de controle de um braço robótico, ou analisando a resposta de um motor, e analisando a estabilidade de um filtro digital.

Estas raízes podem ser obtidas através do comando **r = roots(p)**, sendo p um vetor linha qualquer. As raízes podem ser rearrumadas de novo no polinômio original com o comando **q = poly(r)**
Exercício: Ache as raízes dos seguintes polinômios:
a) $\mathbf{x^2 + 2x + 4}$ **b)** $\mathbf{3x^3 - 2x^2 + 5x - 20}$ **c)** $\mathbf{4x^4 - 2x^2 + 10}$ **d)** $\mathbf{8x^3 - 12x + 2}$

2.2.4- Operações Aritméticas

2.2.4.1- Soma e Subtração

Para somar ou subtrair polinômios basta somar ou subtrair seus respectivos coeficientes. O MATLAB não apresenta um comando específico para somar polinômios. A soma ou subtração padrão funciona se ambos os vetores polinomiais forem do mesmo tamanho. Somemos os polinômios a seguir:
$g(x) = x^4 - 3x^2 - x + 2.4$ $h(x) = 4x^3 - 2x^2 + 5x - 16$
$som(x) = g(x) + h(x)$ $sub(x) = g(x) - h(x)$

2.2.4.2- Produto

O produto dos polinômios é a convolução de seus coeficientes.O produto dos polinômios a(s) e b(s) pode ser obtido com o comando c = conv(a,b):
Exemplo: a = [1 0 -20.6] ; b = [1 19.6 151.2];   c=conv(a,b)

2.2.4.3- Derivada e Integral

O comando **polyder** serve para derivar o polinômio, enquanto que o comando **polyint** serve para integrar o polinômio.
Exemplo: p= [4 2 2 1 3];

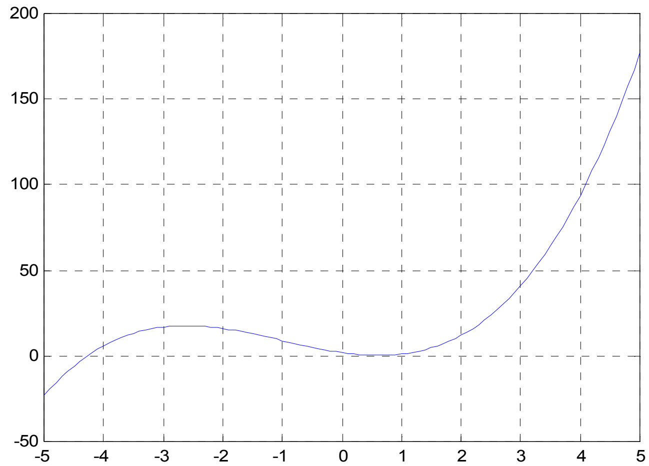
pd = polyder(p)	**polyint(p)**
pd = 20 8 6 2 3	**ans = 4 2 2 1 3 0**

Exercício: Derivem e integrem os seguintes polinômios: **$g(x) = x^4 - 3x^2 - x + 2.4$ $h(x) = 4x^3 - 2x^2 + 5x - 16$**

2.2.5- Avaliação de Polinômios

Para avaliar um polinômio utiliza-se o comando **polyval ,** exemplo: para calcular o polinômio **$x^3 + 3x^2 -5x + 2$**, entre **[-5,5]** utiliza-se os comandos abaixo:

```
pn = [ 1 3 -5 2];
xn = [ -5:0.1:5];
yn=polyval(pn,xn);
plot(xn,yn)
```

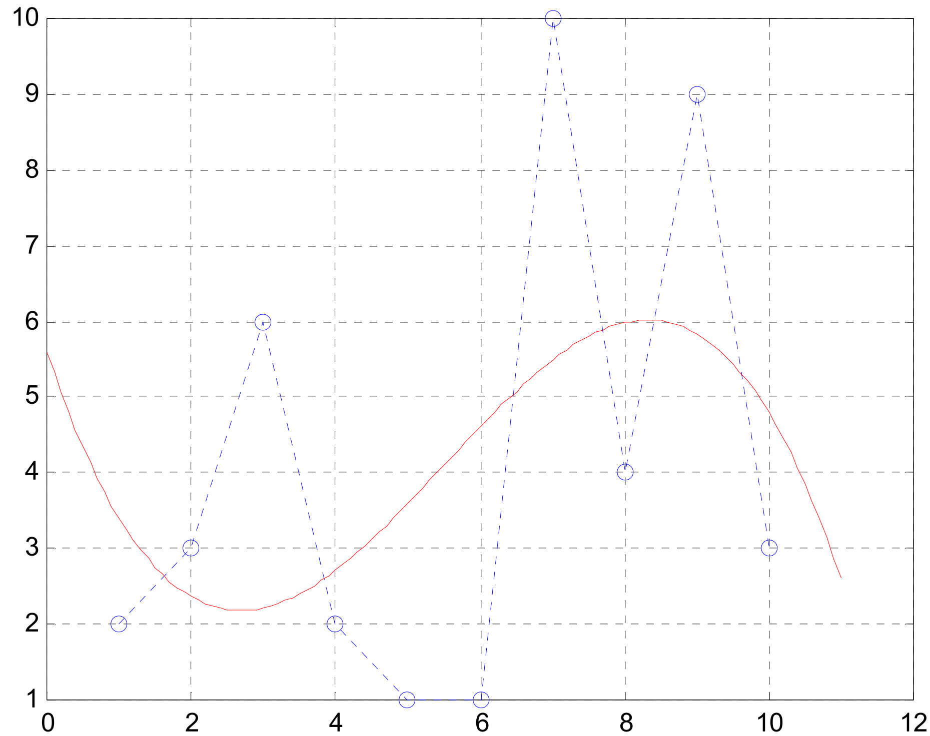


2.2.6 – Ajuste por polinômio (comando polyfit)

Dadas duas séries X e Y, e grau de polinômio N, retorna coeficiente de polinômio de grau N que melhor aproxima (método mínimos quadrados) a curva gerada por (X,Y).

Exemplo:

```
x=1:10;
y=[ 2 3 6 2 1 1 10 4 9 3];
p3=polyfit(x,y,3)  % calcula polinômio de grau 3
xn = 0:0.1:11;
yn=polyval(p3,xn);
plot(x,y,’bo:’,xn,yn,’r’)
```


10
9
8
7
6
5
4
3
2
1
0
2
4
6
8
10
12