**INFORME – INTRODUCCIÓN A MATLAB - POLIMONIOS**

Ing. Samir Fernando Vergara Beltran.

Miguel Angel Gómez Alarcón

Juliana Castillo Araujo

Julio Cesar Junior Prada Hernandez

Diana Mayerly Sanchez Gonzalez

Mayo de 2024.

Universidad De Cundinamarca

Ubaté

Modelación.

**Polinomios En Matlab**

Un polinomio puede ser introducido en Matlab mediante un vector cuyos elementos son los coeficientes reales o complejos del polinomio completo y or- denado (los términos se escriben de mayor a menor grado y se encuentran todos los términos completando con cero si fuese necesario).

Ejemplos:

Definir el siguiente polinomio *P(x)*=4x3-2x2+6

>> p=[4 -2 0 6]

p =

4 -2 0 6

Definir el polinomio S(x)= *x*4 -2*ix*2 + *x*-4 + *i*

>> s=[1 0 -2i 1 -4+i]

s =

1. + 0.0000i 0.0000 + 0.0000i 0.0000 - 2.0000i 1.0000 + 0.0000i

-4.0000 + 1.0000i

**Evaluando en la variable**

* **Función polyval:** Evalúa un polinomio en un valor.

Sintaxis:

Polyval (p, x).

1. > polinomio
2. > valor para evaluar en el polinomio puede ser real o complejo Ejemplo:

>> s=polyval(p,3) s =

96

>> t=polyval(p,2-i) t =

8.0000 -36.0000i

# **Raíces del polinomio**

Tanto para los polinomios con coeficientes reales P(x) como también para polinomios con coeficientes complejos S(z), Matlab nos permite encontrar sus raíces con el uso del comando roots().

### **Función roots**

Sintaxis:

roots(p).

* 1. > polinomio con coeficientes reales P(x) o polinomios con coeficientes

complejos S(x)

Ejemplo:

>> p p =

4 -2 0 6

>> raices\_p=roots(p) raices\_p =

0.7500 + 0.9682i

0.7500 - 0.9682i

-1.0000 + 0.0000i

Raíces del polinomio p con números reales.

>> s= [1 0 -2i 1 -4+i]

s =

1.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 - 2.0000i 1.0000 + 0.0000i

-4.0000 + 1.0000i

>> raices\_p=roots(s) raices\_p =

-1.4841 - 0.2798i

-0.3216 - 1.3889i

0.5966 + 1.4362i

1.2092 + 0.2325i

Raíces del polinomio s con coeficientes complejos.

# **Producto polinomial**

* **Función conv:** multiplicación de polinomios

Sintaxis:

Conv (p, t)

1. > polinomio t->polinomio

Ejemplo:

Definir el polinomio p(x)=x3+3x2-4x+2 y el polinomio t(x)=2x2-3x+1

>> p= [1 3 -4 2]

p =

1 3 -4 2

>> t= [2 -3 1]

t =

2 -3 1

>> s=conv (p, t) s =

2 3 -16 19 -10 2

Matlab devuelve los coeficientes del polinomio resultante s(x)=2x5+3x4- 16x3+19x2-10x+2

# División polinomial

* **Función deconv:** para dividir dos polinomios P(x) y T(x) utilizamos el comando [q, r]= deconv (p, t) y nos devolverá el cociente q(x) y residuo r(x).

Sintaxis:

[q, r]= deconv (p, t)

1. >almacena el cociente de la división r-> almacena le residuo de la división p->polinomio

t->polinomio Ejemplo:

>> p p =

1 3 -4 2

>> t t =

2 -3 1

>> [q, r] = deconv(p, t) q =

0.5000 2.2500

r = 0 0 2.2500 -0.2500

Los polinomios resultantes de la división son: Cociente: q(x)=0.5x+2.25

Residuo: r(x)=2.25x-0.25

# **Derivada de un polinomio**

* **Función polyder (p):** calcula la derivada del polinomio Ejemplo:

Calcular la derivada del siguiente polinomio p(x)=x3-4x2+4

>> p=[1 -4 0 4]

p =

1 -4 0 4

>> d = polyder (p) d =

3 -8 0

El polinomio resultante es d(x)=3x2-8x

# **Integración de un polinomio**

**Función polyint:** calcula la integral de un polinomio.

Sintaxis:

Polyint (p)

p->polinomio

Ejemplo:

Calcular la integral del siguiente polinomio p(x)=□2x□^3+3x^2-2

>> p= [2 3 0 -2]

p =

2 3 0 -2

>> polyint(p) ans =

0.5000 1.0000 0 -2.0000 0

Nos da como resultado: ∫p(x) = 0.5x4+x3-2x

# **Polinomio interpolador**

**Función polyfit:** ajusta una curva polinómica a partir de unos puntos, el comando devuelve los coeficientes de un polinomio P(x) de grado n que tiene un mejor ajuste para los valores en (x, y).

Sintaxis:

Polyfit (x, y, n)

x, y ->puntos de ajuste n->grado del polinomio

Obtener el polinomio interpolar a partir de los siguientes puntos x=0 a 20 y=cos(x).

>> x= linspace (0,20,15);

>> y= cos(x);

>> p= polyfit (x, y, 4)

p =

0.0003 -0.0103 0.1376 -0.6703 0.8154

El polinomio resultante es: p(x)=0.0003x4-0.0103x3+0.1376x2- 0.6703x+0.8154

# **Ejercicios resueltos**

1. Dado el polinomio P(x)=x4-2x2+4x+6

Evaluar p(π2)

Determinar las raíces del polinomio Calcular la derivada

>> p=[1 0 -2 4 6]

p =

1 0 -2 4 6

>> eval=polyval(p, pi^2)

eval =

9.3392e+03

>> raices=roots(p) raices =

1.3345 + 1.2370i

1.3345 - 1.2370i

-1.3345 + 0.1772i

-1.3345 - 0.1772i

>> derivada=polyder(p) derivada =

4 0 -4 4

1. Dado el polinomio q(x) = - x3+ix4+5x-2

Evaluar q (3-i) Determinar las raíces

Multiplicarlo con el polinomio s(x)=3x2-4

>> q=[i -1 0 5 -2]

q =

0.0000 + 1.0000i -1.0000 + 0.0000i 0.0000 + 0.0000i 5.0000 + 0.0000i

-2.0000 + 0.0000i

>> val=polyval(q,3-i) val =

91.0000 +49.0000i

>> raices=roots(q) raices =

-0.1008 - 2.1270i

-1.5595 + 0.5790i

1.2465 + 0.5546i

0.4138 - 0.0065i

>> s=[3 0 -4]

s =

3 0 -4

>> mult=conv(q,s) mult =

Columns 1 through 6

0.0000 + 3.0000i -3.0000 + 0.0000i 0.0000 - 4.0000i 19.0000 + 0.0000i

-6.0000 + 0.0000i -20.0000 + 0.0000i

Column 7

8.0000 + 0.0000i

1. Crear un polinomio p de grado cuatro, con coeficientes iguales a los primeros términos de la sucesión de Fibonacci, el último término es el término independiente; es decir, 1.

Crear un vector d, con elementos iguales a los del polinomio anterior pero en orden inverso; evaluar el polinomio p en cada uno de los términos del vector d.

Hallar las raíces de p.

Mostrar la derivada e integral del polinomio p.

>> p = [8 5 3 2 1]

p =

8 5 3 2 1

>> d=p(5:-1:1)

d =

1 2 3 5 8

>> eva\_p=polyval(p,d) eva\_p =

19 185 817 5711 35537

>> raices=roots(p) raices =

0.1720 + 0.5886i

0.1720 - 0.5886i

-0.4845 + 0.3125i

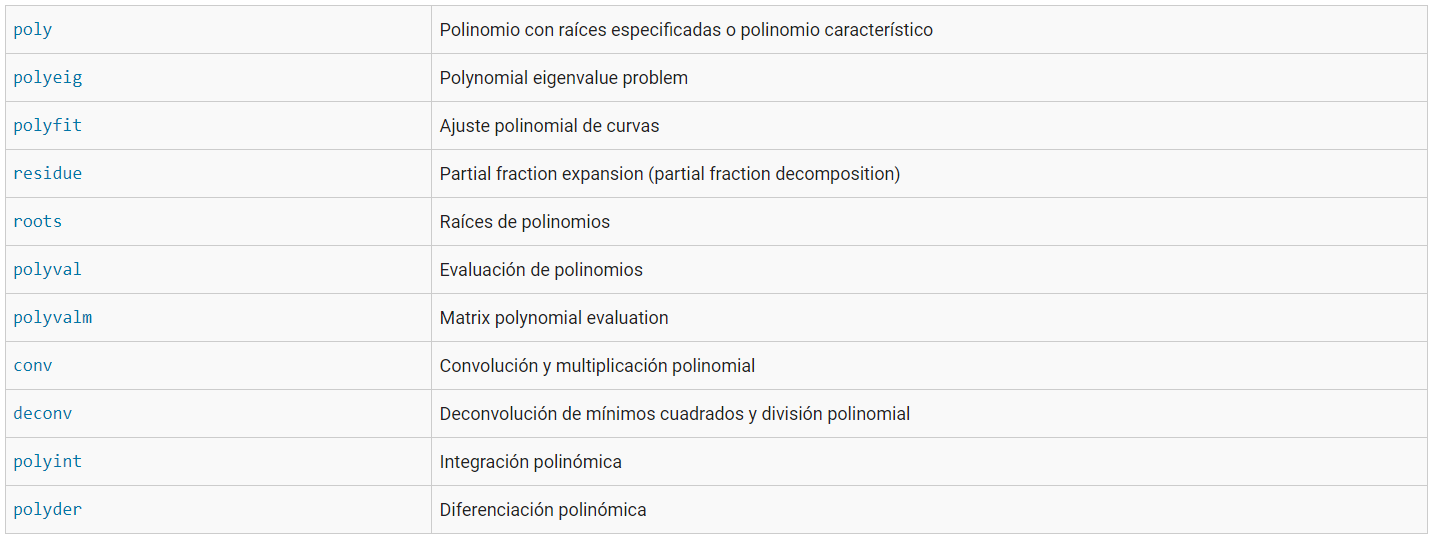
-0.4845 - 0.3125i

>> derivada = polyder (p) derivada = 32 15 6 2

>> integral = polyint (p) integral =

1.6000 1.2500 1.0000 1.0000 1.0000 0

**Nota: Funciones**

****