Computación 1
- 2021 -

Polinomios en Octave

Polinomios

Elementos básicos

$$f(x) = a_0 x^{N} + a_1 x^{N-1} + a_3 x^{N-3} + a_2 x^{N-2} + \ldots + a_{N-2} x^{2} + a_{N-1} x + a_N$$

Variable: x

Coeficientes: a_i , $i = 0 .. N$

Grado: N

Polinomios

Reglas de representación en Octave

- Los coeficientes ordenados en forma decreciente por su grado
- Completitud: deben estar TODOS los coeficientes, aún si su valor = 0

(estructura *posicional*)

» a = [1 1 1] % representa: $x^2 + x + 1$

» a = [2 0 1 3] % $2x^3 + 0x^2 + x + 3$

Polinomios

Operaciones: **Suma y resta**

» [1 1 1 1] + [3 2 1 0]

ans =

4 3 2 1 % $= 4x^3 + 3x^2 + 2x + 1$

¡Ambas representaciones deben ser de igual largo (cantidad de elementos)!

» [0 0 1 1] + [3 2 1 0] %$(x+1)(3x^3+2x^2+x)$

ans =

3 2 2 1 % la resta es análoga

Polinomios

Operaciones: **Producto**

Polinomio **x** *escalar*

» [3 2 1 0] * 3   % $(3x^3 + 2x^2 + x)\ 3$

ans =

    9   6   3   0   % $9x^3 + 6x^2 + 3x$

Polinomio **x** *polinomio*   Ej: $(x + 1)(3x^3 + 2x^2 + x)$

» [0 0 1 1] * [3 2 1 0]‘

ans =

    1

% ¡Este resultado no es correcto!

Polinomios

Operaciones: **Producto**

» conv([0 0 1 1], [3 2 1 0])

ans =

0 0 3 5 3 1 0

No es necesario que sean de igual largo:

» conv([1 1], [3 2 1 0])

ans =

3 5 3 1 0

Objetivo:

$(x + 1)(3x^3 + 2x^2 + x)$

$$
\begin{array}{lllllllll}
3x^4 & + & 2x^3 & + & x^2 & & & & \\
 & & 3x^3 & + & 2x^2 & + & x & & \\
\hline
3x^4 & + & 5x^3 & + & 3x^2 & + & x & + & 0
\end{array}
$$

Polinomios

Operaciones: **Cociente**

```
» [c, r] = deconv( [3 5 3 1 0], [ 3 2 1 0 ] )
c =
    1     1
r =
    0     0     0     0     0

» [c, r] = deconv( [3 5 3 1 0], [ 1 1 ] )
c =
    3     2     1     0
r =
    0     0     0     0     0
```

% El resultado se devuelve en 2 vectores (cociente y resto)

% vectores completos

% Si se recibe el resultado en un vector sólo obtenemos el cociente

Polinomios
Operaciones: **Raíces**

Octave provee una función que halla las raíces de polinomios con una precisión determinada; puede no ser la que le sirve al usuario.

Éste deberá verificar si precisión y tiempo de cálculo se adecuan a su problema

(en Métodos Numéricos se verán algunas alternativas)

```
» roots( [ 4 2 1 ] )
ans =
   -0.2500 + 0.4330i
   -0.2500 - 0.4330i

» roots( [ 8 4 2 1 ] )
ans =
   -0.5000 + 0.0000i
    0.0000 + 0.5000i
    0.0000 - 0.5000i
```

Polinomios

Operaciones: poly() **Construir un polinomio**

La función inversa a hallar las raíces es construir un polinomio que tenga raíces dadas:

```
» r1 = roots( [ 4 2 1 ] )
r1 =
  -0.2500 + 0.4330i
  -0.2500 - 0.4330i

» poly( r1 )
ans =
  1.0000 0.5000 0.2500
```

```
» roots( [ 1 0.5 0.25 ] )
ans =
  -0.2500 + 0.4330i
  -0.2500 - 0.4330i
```

Polinomios

Operaciones: polyval() **Evaluar un polinomio**

Argumentos: polinomio y un escalar

```
» polyval( [ 4 2 1 ],2 )
ans =
     21
```

Argumentos: polinomio y una matriz

```
» polyvalm([3,2,1],[1,0;0,1])
ans =
         6      0
         0      6
```

Polinomios

```
Operaciones: polyder() Deriva un polinomio
» polyder( [ 4 2 1 ])
ans =
     8     2
Operaciones: polyint()Integra un polinomio
» polyint([4,2])
ans =
     2     2     0
```

Polinomios
Operaciones: **Resumen**

- Se representan usando vectores
- En algunos casos las operaciones de vectores resuelven correctamente las operaciones con polinomios
  - Suma (y resta)
  - Producto de un polinomio por un escalar

Polinomios
Operaciones: **Resumen**

- En otros casos hay funciones específicas:
  - Producto (y cociente) entre polinomios
  - Raíces (construcción de polinomio)
  - Evaluar polinomios
  - Derivar
  - Integrar

Polinomios

para el curso

?

Polinomios ….

para el curso

Una estructura de datos sencilla
(e intuitiva) para realizar diferentes
algoritmos !!!!