

INTRODUCCION a FreeMat / MATLAB / Octave

Este tema se dictará en dos clases, según la siguiente distribución de contenidos en cada una de ellas:

En CLASE 1:

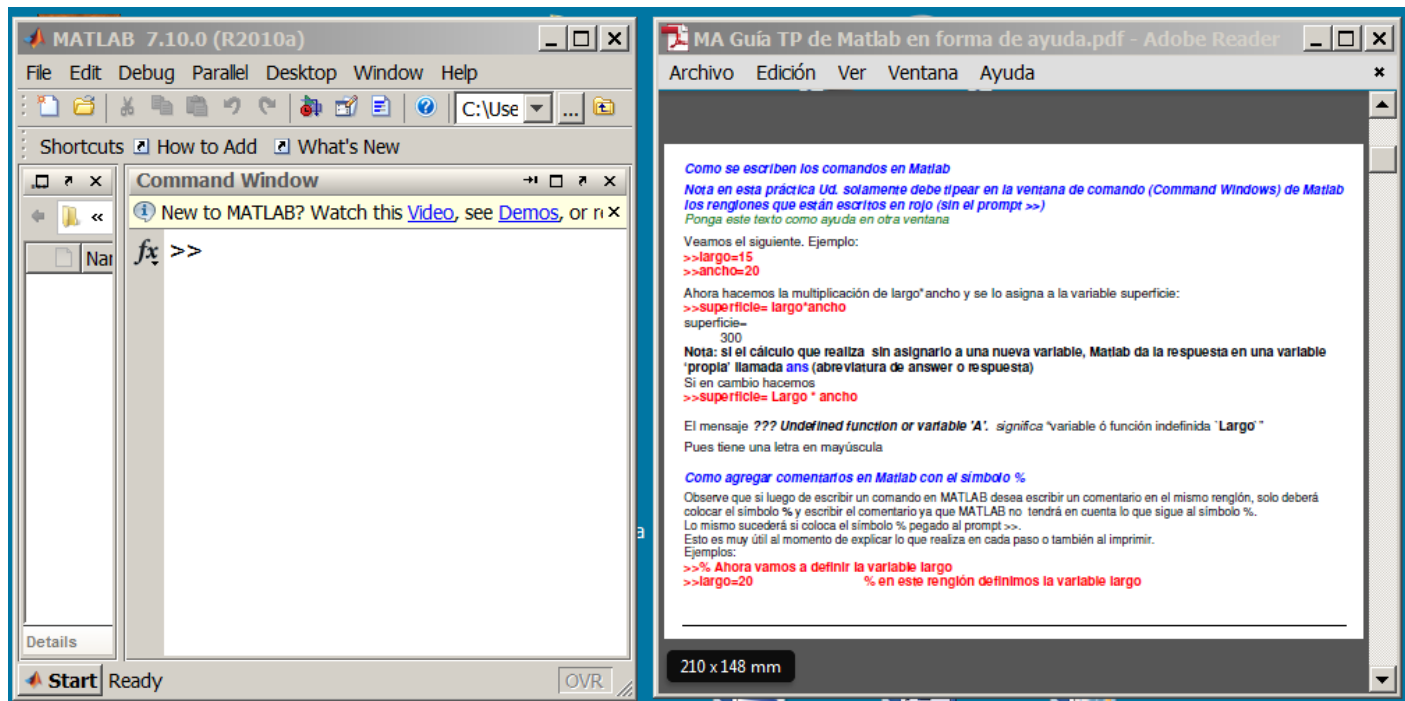
- **INTRODUCCION**
 - **POLINOMIOS**
 - **DEFINIR**
 - **EVALUAR**
 - En un valor de x
 - En varios valores de x (TABLA)
 - **HALLAR SUS RAICES O CEROS**
 - **GRAFICAR un polinomio**
 - OTROS comandos para gráficos.
 - Graficar rectas verticales y horizontales.
 - **DEFINIR UN POLINOMIO a partir de sus raíces**
-

En CLASE 2:

- **FUNCIONES NO POLINOMICAS**
 - **INTRINSECAS: SENO, COSENO, LOG**
 - **OTRAS**
 - **DEFINIRLAS**
 - **HALLAR CEROS O RAICES**
 - **GRAFICAR**
- **RESOLUCION SISTEMAS ECUACIONES LINEALES (SEL)**
 - **MATRICES Y VECTORES**
 - **RESOLUCION GRAFICA**

Antes de comenzar a recorrer esta ayuda (este archivo pdf) arranque el programa FreeMat.

Coloque las ventanas en simultáneo, en MOSAICO HORIZONTAL como observa en la figura, de modo de poder leer en la derecha y realizar los comandos (escribirlos) de esta lección.



Ventana de comandos (Command Windows) ... QUE ES EL PROMPT?

Aparece así: >> Indica que el programa (FreeMat) está listo para recibir órdenes (comandos) para ejecutarlos, siempre que su sintaxis esté correcta.

INTRODUCCION

MATLAB que significa MATrix LABoratory es un software aplicado de matemática que nos facilitará, debido a sus innumerables funciones y comandos, el trabajo en ingeniería.

MATLab es un programa en el que todos los datos son matrices, o sea **los vectores y escalares** se comportan como casos particulares de matrices.

Las versiones más recientes de MATLAB poseen un entorno para escribir los comandos y otras ventanas para el historial, la edición y la ayuda.

Los nombres de las variables en MATLAB / Freemat

Para realizar operaciones debemos definir variables

En **MATLAB** los nombres de las variables a utilizar deben cumplir ciertos requisitos:

- comienzan con letras y no admiten espacios ni caracteres raros (/&%\$?)
- se diferencia entre nombres con **mayúsculas y minúsculas**
- como norma se escriben en:

minúsculas (los vectores y escalares)

MAYUSCULAS (las matrices)

NOTA: En esta guía los comandos que escribiremos para hacer los ejercicios están escritos en ROJO

Ud. solamente deberá tipear en la ventana de comandos (Command Windows) lo que está en rojo (sin el prompt >>)

Ejemplo 1: definimos las variables escalares **largo y ancho**

```
>>largo=15
```

```
>>ancho=20
```

Ahora hacemos la multiplicación $\text{largo} \times \text{ancho}$ y asignamos el resultado a la variable **superficie**:

```
>>superficie= largo*ancho
```

```
superficie=
```

```
300
```

Las tres son variables escalares !

Nota: Se observa la respuesta del Sw. Si el cálculo que realiza lo asignamos a una nueva variable, en este ejemplo superficie nos lo informa, SI NO asignamos el resultado a una variable dará la respuesta en una variable 'propia' llamada **ans**, abreviatura de answer (respuesta)

Ejemplo 2: si hacemos ahora

>>superficie= Largo * ancho

El mensaje **??? Undefined function or variable 'A'**. Este mensaje significa “variable ó función indefinida **“Largo”**”

Pues tiene una letra en mayúscula, la L, esa variable no está definida, sí existe en memoria la variable largo.

Cómo agregar comentarios en Matlab: con el símbolo %

Si luego de escribir un comando se desea escribir **un comentario en el mismo renglón**, solo se deberá colocar el símbolo % y escribir a continuación el comentario.

El Sw no tendrá en cuenta lo que sigue al símbolo %.

Lo mismo sucederá si coloca el símbolo % a continuación del prompt >>

Esto es muy útil si se desea explicar algo, agregar un comentario o recordar lo que se realiza en cada paso.

Ejemplo 3: de este modo largo “pierde” el valor 15 y pasa a guardar el valor 20

>>%Ahora vamos a definir la variable largo

>>largo=20 % en este renglón re definimos la variable largo

POLINOMIOS

Definir un polinomio:

Un polinomio se define con **un vector**, cuyos elementos son los **coeficientes del polinomio**.

Para definir un vector se le asigna un nombre (generalmente en minúsculas) y entre corchetes se dan los valores de las componentes de ese vector. Más adelante se ven más detalles del tema: vectores y matrices.

Dado el polinomio: $P(x) = x^3 - 2x^2 + 5x - 4$

Para definirlo primero debemos darle un nombre CUALQUIERA, NEMOTECNICO, por ejemplo **pol3 ó poli3**.

Nota: En general recomendamos llamar **polN** a los polinomios, donde **N** es **el grado del polinomio**, pero también podría llamarse **fpoli** o cualquier nombre permitido de Matlab/Freemat, por ejemplo: **poli**

Los valores de los coeficientes del polinomio se asumen en el vector desde el de mayor exponente hasta **x^0** , **por eso el polinomio debe estar ORDENADO**.

Así el polinomio dado $P(x)$ se puede representar en Matlab:

$$\begin{array}{cccc} x^3 & - & 2x^2 & + & 5x & - & 4 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \end{array}$$

>>pol3 = [1 -2 5 -4] -4 es el término independiente que corresponde a **x^0**

Ejemplo 4:

Dado el polinomio: $T(x) = x^2 - 4$ definirlo con Freemat.

Es de grado dos, INCOMPLETO. Se lo debe COMPLETAR. De este modo tiene los coeficientes:

$$\begin{array}{ccc} 1 & 0 & -4 \\ & \downarrow & \downarrow & \downarrow \\ 1 \cdot x^2 + 0 \cdot x - 4 \end{array}$$

Definimos como **pol2** así:

>>pol2 = [1 0 -4] % Como se observa se completó el polinomio con un coeficiente 0 para x

Importante: El polinomio debe estar completo y ordenado antes de escribir sus coeficientes para definirlo

Ejemplo 5: Dado el siguiente polinomio $5x^2 - 2x^4 - 3 + 2x$

*Darle el nombre **pol4**.

*Ordenarlo (si corresponde)

*Completarlo (si corresponde)

* definirlo y escribirlo:

$$2x^4 + 0x^3 + 5x^2 + 2x - 3 \quad \text{Se ordena y completa.}$$

Se define el polinomio **pol4**

$$\begin{array}{ccccccccc} -2x^4 & + & 0x^3 & + & 5x^2 & + & 2x & - & 3 \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \end{array}$$

>>pol4 = [-2 0 5 2 -3]

Otros casos: Las rectas

Las rectas son polinomios de grado 1

Así la recta r1: $y = 2x - 1$ se define en Matlab o freemat así:

>> r = [2 -1] % el nombre usado aquí es r pero podría ser poli1 o recta

Ejemplo 6:

Escribir o definir la recta con ordenada al origen 0 y pendiente 1.5. La ecuación de dicha recta es:

y = 1.5 x Es un polinomio incompleto de grado 1 y se escribe o define así:

>> r1 = [1.5 0]

Ejemplo 7:

Definir la recta horizontal, paralela al eje de abscisas o con pendiente 0, que corta al eje y en 2.

O sea, la ecuación de esta recta es: $y=k$, donde $k=\text{constante}=2$, luego: **$y=2$** es la ecuación de la recta, es un polinomio incompleto. Se escribe o define en Matlab:

```
>> p1 = [0 2]
```

Evaluar un polinomio. Función polyval

Para evaluar un polinomio, por ejemplo: **pol2 = $x^2 - 4$** en $x=2$

Se usa la función **polyval**(*nom_polinomio, valor escalar o vector*)

Esta función necesita 2 argumentos:

- **el nombre del vector** que define el polinomio (nom_polinomio)
- **un valor de x o una serie de valores de x** (valor escalar o vector)

>> polyval(pol2,2) % da como respuesta 0 pues es $x^2 - 4$ será en $x=2$ $2^2 - 4 = 0$

NOTA: Recordemos que si el resultado es 0 significa que el valor 2 es una de las raíces de ese polinomio.

Cómo evaluar un polinomio para una serie de valores de x:

Primero debemos obtener **una serie de datos (vector x)** donde se desea evaluar un polinomio.

El operador dos puntos (:)

Este comando permite generar una serie de valores (un vector) partiendo de un valor inicial con un incremento hasta un valor final, separados todos ellos por dos puntos (:)

Ejemplo 8: crear una serie de valores o datos de n desde 0 a 100 con salto (incremento) de a 2.

Se escribe: `>> n = 0 : 2 : 100`

Ejemplo 9: Crear la serie x (usando el operador dos puntos) para: xmin= -3 , xmax= 3 y dx= 0.1

`>>x = -3 : 0.1 : +3` *% se define un vector x = -3 -2.9 -2.8 -0.1 0.0 0.1 ... 2.9 3*

Ejemplo 10: Evaluar el polinomio **pol2** en cada uno de los valores de la serie x recién definida.
O sea se obtendrán los correspondientes valores de la variable dependiente y para cada x.

Recordar que el polinomio $y=T(x)=x^2 - 4$ se definió como `pol2 = [1 0 -4]` y los valores de x están definidos:

`>>y=polyval(pol2,x)` *% Se calcula el valor de pol2 para cada x, se guarda cada valor en el vector y*

Armar una tabla con los valores de x e y

Obtuvimos dos **vectores fila** (horizontales) llamados **x** e **y**.

Si queremos ver los valores x-y como una tabla vertical, conviene **transponer los valores** de x e y, pasándolos a vectores columnas y luego “pegarlos” en una matriz de 2 columnas que podemos llamar T

El operador 'apóstrofe' **transpone** filas a columnas o columnas a filas. Sirve para vectores y matrices.

El comando es el siguiente:

$\gg T = [x' y']$

Y obtenemos entonces una tabla que luce como se acostumbra al tabular funciones matemáticas:

```
-3.0000  5.0000
.....
-1.0000 -3.0000
.....
 0.0000 -4.0000
.....
 2.0000  0.0000
.....
 3.0000  5.0000
```

Operador punto y coma (;)

Nota: si queremos que no se impriman en la ventana de Matlab los 61 valores de x ó de y, se agrega un punto y coma al final del comando:

$\gg x = -3 : 0.1 : 3;$ % no imprime o muestra los valores en pantalla, pero SI ESTAN almacenados en memoria!

Esto puede hacerse para todos los comandos cuyos resultados NO QUEREMOS VER en pantalla.

GRAFICAR UN POLINOMIO

Ya obtuvimos los valores de (x, y) o sea la Tabla de los valores (y) del polinomio para los valores de x de -3 a $+3$.

Podemos representar gráficamente esa tabla de valores (o el polinomio) para ver la forma del mismo, sus ceros o raíces, el rango, entre otras cosas.

El comando **plot** permite graficar esa tabla de valores x, y (debemos tener pares x, y)

>>plot(x,y) % se grafican los valores de x e y

Se obtiene el gráfico en una ventana (Figure 1). Esta ventana permite copiar la figura (usar Edit)

Con el comando **grid on** se agrega una grilla que permite mejorar la visualización de la función

>>grid on

- ✓ **SE PUEDEN OBSERVAR LAS RAICES O CEROS del polinomio (lugar donde la función cruza al eje de abscisas, o sea donde $y=0$)**
 - ✓ **CONVIENE siempre graficar las funciones matemáticas para LUEGO hacer las determinaciones y los cálculos solicitados.**
-

Calcular las raíces del polinomio, comando roots

Para calcular las raíces del polinomio **pol2**:

```
>>roots(pol2)
```

```
ans =  
    2.000  
   -2.000
```

pol2 posee 2 raíces reales.

Se puede VERIFICAR, que $y=0$ en $x=\text{raíz}$ con:

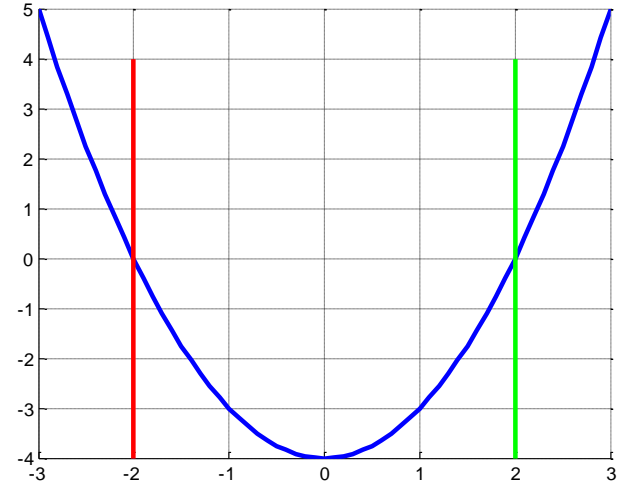
```
>>polyval(pol2,2) % se evalúa solamente para el valor x=2
```

```
ans=  
    0
```

```
>>polyval(pol2,-2) % se evalúa solamente para el valor x=-2
```

```
ans=  
    0
```

Comprobamos de esta manera que 2 es una de las raíces y -2 la otra. Observamos el rango: $l = [-4, 5]$



Marcar las raíces con una línea Vertical

Uso del comando `plot`

Una recta paralela al eje y se define como $x=k$, con $k=\text{constante}$.

En este ejemplo anterior serían las rectas: $x=2$ y $x=-2$

Si para $x=2$ se definen dos pares ordenados de modo que la vertical “corte” al eje x, por ejemplo **(2,-4) y (2, 4)**, vemos que para **$x=2$: $y=-4$ e $y=4$** . Igual se puede proceder para $x=-2$

Con el comando **plot** se traza una línea vertical (o paralela al eje y) usando dos pares ordenados:

```
>> plot([xinicial xfinal] , [yinicial yfinal])
```

Para el ejemplo x_{inicial} es -2 , x_{final} es -2 , y_{inicial} es -4 e y_{final} +4. Para la raíz positiva es similar:

```
>>plot([-2 -2] , [-4 4])
```

```
>>plot([ 2 2] , [-4 4])
```

En la figura anterior se observan las verticales en rojo y verde, además de mayor espesor. Los comandos usados fueron:

```
>>plot(x,y,'LineWidth',3)
```

```
>>plot([-2 -2],[ -4 4],'r', 'LineWidth',3)
```

```
>>plot([2 2],[ -4 4],'g', 'LineWidth',3)
```

Comandos a utilizar en plot para los gráficos

'LineWidth', 3 para obtener líneas más gruesas, más espesor

'r' es para dar color rojo

'g' para dar color verde

OTROS comandos para mejorar la presentación y el aspecto de los gráficos:

Ejes, títulos y grilla.

Los siguientes comandos son complementos para mejorar los gráficos:

hold on	Mantiene activo el gráfico para agregarle ejes, títulos u otro gráfico si no se indica hold on, se borra el gráfico al realizar otro
hold off	Desactiva el hold on
xlabel('Título eje x')	Agrega los nombres del eje X
ylabel('Título eje y')	Agrega los nombres del eje Y
grid on/off	Muestra la grilla x/y o la oculta
title ('Polinomio')	Agrega un título

`axis([xmin xmax ymin ymax])`

`clf`

Fija las escalas de los ejes x/y

Borra la figura

Generación de un polinomio a partir de sus raíces

Comando poly

Se desea generar un **polinomio** cuyas raíces son : - 2 y +2

Matemáticamente: $y = (x-2)(x+2)$ donde el primer factor considera la raíz $x=2$ y el segundo $x=-2$

Con Freemat, se debe aplicar la función **poly** indicando sus dos raíces:

```
>> polin2 = poly([- 2 2] )
```

Se obtiene el vector polin2= [1 0 -4]

Se generó el polinomio incompleto: $y = x^2 - 4$

Generación de un polinomio de mayor grado

Si se desea generar un polinomio de grado 3 con las raíces $x = -3$, $x = 1$ y $x = +3$ se escribe así:

```
>> cubica = poly([-3 1 3 ])
```

Se obtiene el vector cubica= [1 -1 -9 9]

Que representa en polinomio: $y = x^3 - x^2 - 9x + 9$
