**INTRODUCCIÓN AL MANEJO DE VECTORES EN MATLAB**

El propósito de Matlab es facilitar el trabajo a la hora de realizar operaciones con números, pero este programa lo ve todo como si fueran matrices, de ahí su nombre (Matrix Laboratory), Matlab clasifica el manejo de sus datos de la siguiente forma:



|  |  |
| --- | --- |
| **Arrays o arreglos:** son distribuciones de números en forma rectangular o cuadrada, se caracterizan porque pueden tener una o más dimensiones como en la figura que vemos al frente (3 dimensiones), cada dimensión es un arreglo de 4 columnas por 6 filas, (cada arreglo en cada dimensión debe ser de igual tamaño). Se utilizan, por ejemplo, para representar imágenes de alta calidad, guardar una gran cantidad de datos. |  |
| **Matrices o matrix**: se utiliza este término para representar sólo una dimensión del arreglo, se compone de filas o columnas, también se les conoce como arreglos bidimensionales.Se utilizan, por ejemplo, para representar imágenes de menor calidad, bases de datos, bits en el interior de un archivo, etc.  |  |
| **Vectores:** son la división de una matriz en una fila o columna, representan datos numéricos o tipo texto (por lo general los vectores son de un solo tipo, P.E un vector de datos numéricos no podrá contener datos de tipo texto o alfanuméricos). |  |
| **Escalares:** Así se le llaman a los números en términos de matrices. |  |

**ARREGLOS UNIDIMENSIONALES (VECTORES)**

Es una estructura que se utiliza para almacenar y manipular gran cantidad de datos y se encuentran dispuestos en filas y columnas, por lo general se denotan con letras minúsculas.

|  |  |
| --- | --- |
|  |  |

**Ingresar vectores**

Los elementos del vector se ingresan entre corchetes [ ], si se separan por espacios será vector fila y con ; serán vector columna.

**Transponer vectores**

Es cambiar los vectores fila por columna y se hacen con una comilla al final de la variable del vector. P.E

|  |  |
| --- | --- |
| El cambio solo se hace de forma visual. | Se puede asignar a otra variable si se quiere hacer de forma definitiva. |

**Transponer un vector que no se encuentra en una variable:**

Se desea transponer un vector utilizando el comando de Matlab, pero al aplicar el comando no ocurre nada. P.E:



La explicación es que se está transponiendo el último elemento, y la transpuesta de un escalar no tiene modificación alguna, por lo que se aconseja llevar el vector a una variable o escribirlo entre paréntesis. P.E:



**Acceder a los elementos de un vector**

Para acceder a los elementos individuales de un vector se utilizan subíndices, x(n) y si se desea acceder al último elemento del vector será con x(end).

|  |  |
| --- | --- |
| El vector en la posición 3 tiene el elemento 2 | En la posición final del vector se encuentra el elemento 6 |

**Para cambiar un elemento al interior del vector:**

En el caso del vector del ejemplo anterior el último elemento está en la posición 5 y es el número 6, para cambiar un elemento se escribe la variable del vector con la posición entre paréntesis, luego el número por el cual se va a reemplazar P.E:

|  |  |
| --- | --- |
|  | *Se observa que la última posición del vector se ha reemplazado por el número 100.* |

**Redimensionar**

Se puede redimensionar un vector al agregarle elementos, P.E: el vector **a** tiene 3 elementos, pero al asignarle a la posición 7 el número 5, pondrá las posiciones restantes en cero.



**Concatenar**

Se pueden concatenar o juntar dos vectores ya creados:



**Acceso a elementos internos**

Se puede acceder a elementos en posiciones consecutivas del vector mediante el signo dos puntos “ : “, P.E: nvector (2:5), la posición inicial será 2 y la posición final es 5, los : significa hasta.

|  |  |
| --- | --- |
|  | Crear un nuevo vector |

**Eliminar elementos**

También se pueden eliminar elementos de un vector asignándole vacío [ ] al elemento o elemento que se van a eliminar.

|  |  |
| --- | --- |
|  |  |

**Actividad 1**

1. Con el vector de prueba completo:



¿Qué efecto tendrán estos comandos?  , explicar

1. Para el manejo de vectores Matlab tiene unos comandos especiales, dos de ellos son length y size, consulta la ayuda de Matlab para definirlos y encontrar la diferencia entre ellos.

**Para obtener más de una posición específica**

Se pueden especificar las posiciones del vector en corchetes además de paréntesis.



**Crear vectores de números consecutivos**



**Crear vectores con intervalos**





**Preguntitas capciosas**: ¿Cuál será el último elemento del vector s = 1:5:10? Y ¿cuál será el último elemento de p = 10:-5:-4 ?

¿Qué pasará al crear este vector? w = 7:3:1 Aplicar el comando **size** y explicar.

**Tarea**

Consultar los comandos **linspace** y **rand**, realizar 2 ejemplos de cada uno.

Los comandos **ones** y **zeros** son comunes en matrices, pero también sirven para crear vectores, consultar su funcionamiento y crear 2 vectores con cada comando.

**Crear vectores con números aleatorios:**

Como ya sabrán, el comando **rand** crea un vector o matriz con números aleatorios, siempre de 0 a 1 y entregando 4 decimales:

|  |  |
| --- | --- |
|  |  |

Se ingresan como parámetros la cantidad de filas y columnas que tendrá la matriz o vector resultante.

Para crearlos con elementos en un rango diferente se puede hacer lo siguiente:



Se puede multiplicar o sumar, o cualquier otra operación, para obtener los resultados en el rango deseado.

Además, con el comando **fix** se puede obtener solamente la parte entera de los elementos.



**Actividad 2 (se aconseja el uso de variables para almacenar los vectores)**

1. Crear un vector consecutivo desde 1 hasta 7.
2. Crear un vector con valor inicial 3 hasta 21, con intervalo de 3.
3. Crear un vector desde 1 hasta 100 con intervalo de 0.5.
4. Al vector del punto 3, cambiar el elemento de la posición 50 por 5000.
5. ¿De cuánto es la longitud del vector del punto 3?
6. ¿De cuánto es el tamaño del vector del punto 2?
7. Crear un vector desde 50 hasta 1 con intervalo 1.
8. Crear un vector en el cual sus elementos estén distribuidos uniformemente entre sí, de 1 a 100, pero que tenga solamente 50 elementos.
9. Crear un nuevo vector a partir de vector anterior, pero que contenga solamente los elementos de las posiciones 25 a 40
10. Del vector resultante del punto anterior, eliminar las posiciones del 10 al 16.
11. Crear un vector **r** que contenga los números 1, 0.9, 0.8… 0.1, 0 en este orden.
12. Crear un vector **a** de números enteros de 1 al 100, de este mismo vector extraer los elementos pares y guardarlos en el vector **pares**, y en el vector **imp** guardar los impares.
13. Crear un vector de 25 elementos que estén uniformemente distribuidos entre -0.5 y 0.5.
14. Crear un vector de elementos aleatorios entre 1 y 10.

**Operaciones entre vectores y escalares**

Asumiendo x como vector y k como escalar

|  |  |
| --- | --- |
| **Expresión** | **Operación** |
| x + k | Suma a los elementos del vector x el escalar k |
| x – k | Resta a los elementos del vector x el escalar k |
| k\*x | Multiplicación los elementos del vector x por el escalar k |
| x / k | División los elementos del vector x por el escalar k |
| k .^ x | Potenciación del escalar k a cada uno de los elementos de x |
| x .^ k | Potenciación los elementos del vector x a la potencia escalar k |
| x + y | Suma de los vectores x e y con el mismo tamaño |
| x – y | Resta de los vectores x e y con el mismo tamaño |
| x .\* y | Multiplicación elemento a elemento |
| x ./ y | División elemento a elemento por la derecha (por encima) |
| x .\ y | División elemento a elemento por la izquierda (por debajo) |
| x .^ y | Potenciación elemento a elemento |
| x’ | Transpuesta. |
| cross (x,y) | producto cruz de los vectores x e y de dimensión 3 |
| dot (x,y) | producto punto de los vectores x e y |

Las operaciones de vectores con escalares se pueden realizar sin problemas y se hacen uno a uno, pero para operar vectores con otros vectores existen reglas, las mismas del álgebra lineal.

**Ejemplos operaciones con escalares: sea** el vectorp= [7 21 14 2]

|  |  |
| --- | --- |
| Suma | Resta |

La multiplicación y la división se hacen de la misma forma.

**Operaciones entre vectores:**

Para sumar o restar vectores se necesita que ambos tengan el mismo tamaño y sentido, o sea, que sean ambos vector fila o columna:

|  |  |
| --- | --- |
|  |  |
| En caso anterior ambos vectores son fila, pero no tienen igual tamaño. | Los vectores tienen diferentes orientaciones. |

El mensaje de error en rojo traduce: “Las dimensiones de las matrices deben coincidir”.

|  |  |
| --- | --- |
| **Suma** | **Resta** |

La suma es conmutativa, la resta no lo es.

Existen varios tipos de multiplicación en vectores:

|  |  |
| --- | --- |
| **Multiplicación vectorial** | **División vectorial** |
| El comando “**.\***” realiza la operación uno a uno entre los dos vectores. | El comando “**./**” divide cada elemento del vector **e** con su posición correspondiente del vector **p** |

**Producto escalar**

El primer vector debe ser un vector fila y el segundo un vector columna

|  |  |
| --- | --- |
|  |  |

**Actividad 3**

Dados los vectores a= [1 2 3] b= [-2 3 5]

1. Sume 3 a cada elemento del vector **a** y divida cada elemento del vector **b** entre 2.
2. Obtener el total entre **a** y **b**, calcular el producto vectorial y escalar entre **a** y **b**.
3. Crear un vector **ang** para cada uno de los siguientes valores

**ang** = [ 0 π/2 π/4 π/6 π 2π ]

Calcular f = cos(ang), g = sin(ang), h = tan(ang) para cada valor del vector **ang**:

Se deben mostrar los vectores resultantes f, g y h en la ventana de comando de Matlab.

1. Encuentre el área de un círculo con radio **r** y la fórmula Area = πr2 para los **r** =1, 2, 3, 4. Calcule mediante vectores.
2. Obtener un vector fila que contenga los números pares entre 2 y 1000.
3. Elevar a 5 el vector del punto anterior.
4. **Funciones para operar vectores**

Verificar el funcionamiento de cada una de las funciones que se muestran en el cuadro siguiente con el vector **x**, cambiar el formato a “**format bank**”.

Crear x = [6 3 9 2 5]

|  |  |
| --- | --- |
| **Expresión** | **Operación** |
| prod(x) | Obtiene el producto de los valores de x. |
| sum(x) | Calcula la suma de los valores de x. |
| cumprod(x) | Da como resultado un producto acumulado de los valores de x. |
| cumsum(x) | Calcula la suma acumulada de los valores de x. |
| sort(x) | Ordena los elementos del vector x ascendente. |
| sort(x,'descend') | Ordena de forma descendente. |

1. Se tiene un circuito resistivo de 10 elementos en serie:

Ordenarlos ascendentemente y obtener su resistencia total.

1. Calcular la resistencia total del siguiente circuito: (los valores de las resistencias deben ir en un vector y se aconseja utilizar algunas de las funciones vistas en el punto 7).

|  |  |
| --- | --- |
|  |  |

Se aconseja empezar por nombrar un vector: rt= [R1 R2 R3 R4] y a ese vector aplicar las operaciones y/o funciones.

Respuesta: rt =250Ω

1. Encontrar el valor de la resistencia equivalente del siguiente circuito:

(Si, utilizando vectores)



Se sugiere utilizar estas fórmulas:

R23= [R2 R3]

R234= [R23 R4]

Rt= [R1 R234]

Rt = 600 Ω

1. Calcular la capacitancia total para los siguientes circuitos

|  |  |
| --- | --- |
|  |  |
| Ct= 3µf  | Ct= 0.3µf |

**GRÁFICOS EN MATLAB**

Existen varias funciones que se pueden utilizar para graficar en Matlab, por ejemplo:





**Comando PLOT**

Crea gráficos tipo línea en 2D, para usar Plot nos debemos asegurar que los vectores o matrices que vamos a graficar tengan el mismo **tamaño y dimensión**. P.E: si se desea graficar el vector y= [3 9 27] versus tiempo, dicho vector también debe ser de tamaño y dimensión 1x3, t= [1 3 5]. plot(x,t)

**Ejemplo 0:**

Se pueden crear 2 vectores de forma manual y luego graficar, siempre y cuando tengan igual dimensión y cantidad de elementos:

|  |  |
| --- | --- |
|  |  |

Se observa que para cada posición del vector **x**, existe una posición correspondiente en el vector **y**.

Se le puede especificar a la función Plot, que solamente se necesita graficar los puntos, sin líneas entre ellos:

|  |  |
| --- | --- |
|  |  |

También se puede ajustar los límites para graficar agregando el comando axis a la siguiente línea:

|  |  |
| --- | --- |
|  |  |

Se puede complementar el código, mostrando los mismos puntos, pero esta vez con forma de barras:

|  |  |
| --- | --- |
|  |  |

**Ejemplo 0.5**:

1. Se tiene, en un vector, los resultados de las ventas de una tienda correspondientes a los primeros 6 meses del año, graficar dichas ventas mediante barras y gráfica de torta.



|  |  |
| --- | --- |
|  |  |

**Ejemplo 1, graficando una función:**

Abrir un archivo nuevo .m y digitar los comandos de Matlab que se exponen a continuación:

|  |  |
| --- | --- |
| * Para graficar se necesita crear un vector el cual tendrá los puntos a graficar:
* Y en otra variable se tiene la función a evaluar para cada punto del vector anterior.
* Se grafican con:
 | x= 0:2\*pi;y= sin(x);plot(x,y) |

Cada comando generará un vector





Para mejorar la resolución de la gráfica se deben aumentar los puntos del vector inicial, para ello colocaremos un intervalo:

x=0:0.5:2\*pi;



Se vuelve a calcular la línea: 

Y ya tenemos el doble de puntos para graficar.



Para verificar si la gráfica corresponde, podemos observar que los valores de los elementos del vector que creamos van en el eje **X** (se observa en la gráfica que va de 1 hasta 6) y los elementos calculados con la función Seno van en el eje **Y** (Se observa que van desde -1 hasta 1).

**Nota**: para crear el vector se pueden utilizar cualquiera de los comandos vistos al inicio de la guía.

P.E:



**Tarea:** ¿Cuantos puntos se tendrían para graficar si se genera x=0:0.05:2\*pi?

|  |  |
| --- | --- |
| En el archivo .m que estamos editando, vamos a agregar las líneas que tienen comentario: | De esta forma le pondrá cuadrícula a los gráficos: |

Se recomienda que siempre se usen los comandos **clear all** y **close all** al hacer el programa de un gráfico.

**Insertar texto en las gráficas:**

Verificar la función de este código:



**Colores, tipos de líneas y marcadores con Plot**

Plot(x,y,’ estilo de línea marcas de línea color’) los parámetros adicionales no se separan, van juntos entre comillas sencillas.

No se necesitan tener los 3 elementos en una sola gráfica.

P.E: plot(x,y,'-.or')



|  |  |
| --- | --- |
|  |  |

**Ejemplo 2**

Se necesita graficar la función Coseno en un intervalo de 0 a 2π, con 50 puntos distribuidos uniformemente, además, en la misma gráfica, se necesita graficar la función Seno, en el mismo rango, pero no se especifica un intervalo entre dicho rango (al gusto del programador).

La gráfica debe tener título en la parte superior y en sus laterales, las líneas se deben diferenciar entre sí mediante colores y marcadores.

|  |  |
| --- | --- |
|  |  |

Según lo visto en el gráfico, escribir la definición de los siguientes comandos (se puede utilizar la ayuda de Matlab):

**Hold on:**

**Title:**

**Xlabel:**

**Ylabel:**

**Legend:**

**Otra forma de graficar dos funciones o más:**

Vamos a graficar seno y coseno en una misma ventana, pero no tendrán igual rango, por lo que se crean dos vectores, uno para cada función:





**Ejemplo 3**

Digitar el siguiente código en un archivo .m nuevo:



**Graficar funciones en ventanas individuales:**

Se pueden hacer gráficos en ventanas separadas e independientes, son útiles cuando las funciones a graficar no poseen una escala o eje en común.

**Ejemplo 4**

Digitar y ejecutar el siguiente código:

|  |  |
| --- | --- |
|  | Para reflexionar:* ¿Cuántas gráficas esperaba?
* ¿Cuál gráfica es la que aparece?

Puede agregar el comando **figure** en las líneas 6 y 12.Esto hará que Matlab cree una ventana para cada función a graficar.Puede verificar que en la parte superior izquierda de cada ventana aparece figure 1, 2 y 3 respectivamente. |



**Para concluir los conceptos de los comandos básicos de gráficos de Matlab:**

**Hold on**: grafica una o más funciones en el mismo eje.

**Subplot**: grafica las funciones en una misma ventana, pero de forma individual.

**Figure:** crea una ventana nueva para graficar una función.

**Ejemplo 5:**

|  |  |
| --- | --- |
| Graficar voltaje vs corriente para el siguiente circuito: | C:\Users\Alejita\AppData\Local\Temp\SNAGHTML4df660.PNG |

Los valores de voltaje variarán en el tiempo, inicialmente serán 5V y 50Ω, los valores finales serán 15V y 50 Ω.

La fórmula para hallar corriente (I) en Rs es IRs=V1/Rs.

Se crea un vector para los valores del voltaje, la cantidad de elementos depende de la resolución deseada para la gráfica.

|  |  |
| --- | --- |
|  |  |

**Tarea**: Graficar voltaje vs corriente, corriente vs resistencia y voltaje vs resistencia, la resistencia variará desde 50Ω a 500Ω.

Verificar cuidadosamente las respuestas ya que deben estar acordes con la ley de Ohm.

(Utilizar el comando subplot para visualizar las 3 gráficas en una sola ventana, no olvidar títulos de los ejes y de los gráficos).



**Gráficos discretos:**

Son aquellos que muestran una función como líneas que se extienden de una línea base, son prácticamente iguales que las gráficas de funciones continuas, solamente que estas se utilizan para representar datos que serán digitalizados, ya que pueden ser cuantificados.

La matemática discreta se utiliza para representar valores finitos y cuantificables, por lo tanto, las gráficas de sus funciones tendrán una cantidad de elementos o puntos finitos.

Dicha matemática es muy importante para los sistemas de cómputo, ya que es la base para digitalizar señales que el computador pueda procesar.

**Ejemplos de funciones graficadas de forma continua y discreta (stem):**



|  |  |  |
| --- | --- | --- |
|  |  |  |
| Función continua | Función discreta | Función discreta 3D |

**Actividad rapidita:**

1. ¿De cuántos puntos se compone la gráfica discreta?
2. Graficar la misma función, pero transponiendo el vector x.

**2.**



|  |  |
| --- | --- |
|  |  |

El comando **stem** puede utilizar uno o dos parámetros para graficar, en los ejemplos anteriores se observa que el vector se guarda en la variable **x** y la función en **z**, al graficar, se puede digitar **stem(x,z)** o **stem(z),** el resultado será el mismo, esto también es aplicable al comando **plot**.

1. **Graficando dos funciones usando una matriz:**

En ejemplos anteriores se ha visto la función **hold on**, que sirve para graficar dos funciones en una sola ventana o figure.

También se puede realizar esta tarea llevando las funciones a graficar a una matriz:



|  |  |
| --- | --- |
|  |  |
| stem | plot |

**Actividad 4** (Se aconseja crear el código o script en archivos .m)

1. Realizar los gráficos para las funciones que se muestran a continuación:

|  |  |  |  |
| --- | --- | --- | --- |
| **a.** | $y=x^{2}$ -2<=x<=2 | **e.** | $y=3\*2^{-x^{2}}$ -3<=x<=3 |
| **b.** | $y=e^{x}$ 0<=x<=2π | **f.** | $y=\frac{x(x-2)}{(x+1)(x-2)}$ -10≤x≤10Analizar qué pasa en la gráfica cuando x=-1 |
| **c.** | $y=^{1}/\_{(1+x^{2})}$ 0<=x<=2π | **g.** | $$y=\tan(\left(x\right)) -3\leq x\leq 3$$ |
| **d.** | $y=e^{-0.5\*x}\*sen(5x)$ -4≤x≤4 | **h.** | $y=sen\left(x\right).e^{\left(-0.4\*x\right)} 0\leq x\leq 10 $  |

1. Consultar el uso del comando **subplot** y graficar las funciones anteriores utilizando dicho comando.
2. Graficar las funciones $sen\left(x\right)$ y $sen\left(x-π/4\right)$ en una sola gráfica, diferenciarlas entre sí mediante colores y marcadores y escribir una leyenda para identificarlas.

|  |  |
| --- | --- |
| 1. Generar esta misma gráfica (o lo más parecida posible, con las funciones$ sen\left(t\right),$

$sen(t-\frac{π}{2}$) y $sen(t-π$), no olvidar los marcadores, colores y tipos de línea. |  |

1. Graficar estos dos polinomios (consultar el comando **polyval o ezplot**)

f(x) = 3x4 + 2x3+ 7x2 + 2x + 9 (-5<=x<=5)

g(x) = 5x3 + 9x + 2 (-5<=x<=5)

1. Realizar el gráfico discreto de estas funciones:

|  |  |  |  |
| --- | --- | --- | --- |
| a. | $$\cos(\left(2x\right)) 0\leq x\leq 2π$$ | b. | $$0.5sen\left(2x\right) π\leq x\leq 3π)$$ |
| c. | $$e^{x}\*sen\left(x\right) 0\leq x\leq 2π$$ | d. | $$e^{0.3\*x}\*sen\left(3x\right) 0\leq x\leq 2π$$ |

1. Digitar el siguiente código en un archivo .m de Matlab:



**Tarea**: Con el siguiente circuito de fuente variable:

|  |  |
| --- | --- |
|  | 1. Hallar mediante Matlab y vectores, una resistencia equivalente **RS** para R1, R2 y R3 y otra **RP** para R4, R5 y R6, de tal forma que el circuito equivalente sea:
 |



1. En el circuito equivalente hallar los valores de voltaje en RS y RP utilizando la ley de Ohm, estos datos serán de utilidad para hallar la corriente en R6.
2. Graficar la corriente en R6 de acuerdo a las variaciones de voltaje de 5 a 15V de la fuente de voltaje.
3. Para verificar la gráfica, hallar la corriente en R6 para los valores 5V y 15V respectivamente.
4. Tener en cuenta los títulos en las gráficas, tanto el título general como para cada uno de los ejes.

**Actividad 4.5, Gráficos en 3D ->**

Ingresar el siguiente código en un archivo .m (un archivo para cada gráfico).

1. **Logo de Matlab**



1. **Comando Plot 3**

Se utiliza para graficar dos funciones, de tipo continuo, mediante un vector de referencia.

|  |  |
| --- | --- |
|  |  |
|  |  |
| Verificar lo que sucede con la gráfica al eliminar el elemento de cometario y ejecutarlo de nuevo |

1. **Comando stem3**

Crea gráficos discretos en 3D, y puede recibir varias funciones como parámetro.

|  |  |
| --- | --- |
|  |  |

**Comando Mesh**

Crea gráficos 3D en el cual el color se define en el eje z dependiendo de la altura.

Ejemplo 4: ingresar el siguiente código a un archivo .m de Matlab:

|  |  |
| --- | --- |
|  | Función equivalente |

La línea 2 define el rango a graficar, mismo rango para vectores **x** y **y.**

La función se define en las líneas 3 y 4.

**Actividad 5**

Utilizando como muestra el ejemplo anterior, graficar:

|  |  |  |  |
| --- | --- | --- | --- |
| **a.** | $z=xe^{y^{2}-x^{2}}$ -2<=x,y<=2 | **c.** | $z=cos(x)cos(2y)$ 0$\leq x\leq 2π$ $ 0\leq y\leq 2π$ |
| **b.** | $z=x^{2}-8x+y^{2}-6y-0.1xy+50$ -5<=x,y<=5 | **d.** | $z=\sqrt{x^{2}+y^{2}}$ -5<=x,y<=5 |

Complementar cada una de estas gráficas con los comandos Surface y Contour (se aplica igual que el Mesh), definir cada uno de ellos.

Pueden utilizar las ayudas de Matlab.

**Actividad 6**

Realizar las siguientes gráficas en Matlab utilizando el comando **subplot**.



Bibliografía:

Muestras de los principales comandos de gráficos de Matlab.

<http://www.mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html;jsessionid=41bd85a130680f2dacc935a422b1>

Gráficos discretos

<https://la.mathworks.com/help/matlab/ref/stem3.html>