| | | | | | |
|---|---|---|---|---|---|
| **Código del Curso: 10295** | | **Semestre: I** | | **No. De Guía: 1** | |
| **Nombre del Curso: ÁLGEBRA LINEAL** | | | | **Duración (Horas) 2** | |
| **Relación con el micro currículo (tema de clase): Matrices, Eliminación de Gauss Jordan, matrices inversas.** | | | | | |
| **Elementos de seguridad industrial: Cuáles** | | | | **SI** ☐ | **NO X** |
| **Overol** ☐ | **Botas** ☐ | **Gafas** ☐ | **Tapa oídos** ☐ | **Tapa Bocas** ☐ | |
| **OTROS** | ☐ | | | | |
| **Espacio solicitado para la práctica.** | | **Laboratorio X** | | **Campus** ☐ | **Externo** ☐ |

## <u>PRÁCTICA No 1.</u>

## SOLUCIÓN A SISTEMAS DE ECUACIONES

**INTRODUCCIÓN:**

Investigaciones recientes, muestran la necesidad de incorporar la tecnología en el aula de clase, para promover en los estudiantes un aprendizaje significativo y autónomo de los temas propios de la matemática, y para aportar en el desarrollo de competencias propias de la tecnología (Badia, 2016). En concordancia con lo anterior, el modelo pedagógico de la Universidad de San Buenaventura promueve la formación integral del estudiante, en el que la práctica pedagógica “se centra fundamentalmente en el proceso de aprendizaje del estudiante, desarrollando estrategias específicas para el aprendizaje autónomo y significativo” (USB, 2010, p.62).

Desde esta perspectiva, para apoyar los procesos de enseñanza aprendizaje de las Ciencias Básicas, se brinda a los estudiantes la oportunidad de trabajar con el Software Matlab. Este programa está diseñado para resolver problemas de ingeniería y su lenguaje está basado en matrices, que es la forma natural de expresión de las matemáticas computacionales. Una de sus grandes ventajas es la facilidad de visualizar los datos gracias a su interfaz gráfica, lo que nos permite deducir, analizar e interpretar la información obtenida.[1]

La educación matemática no sólo debe centrarse en los contenidos, sino en el desarrollo de procesos tales como: representar, argumentar, demostrar, clasificar, analizar, resolver, conjeturar, razonar, visualizar, plantear, explicar, reconocer,

[1] Recuperado: https://es.mathworks.com/products/matlab/index.html?s_tid=gn_loc_drop

relacionar, describir, transformar, interpretar, entre otros. Por tanto, el desarrollo de esta guía le permitirá al estudiante ahondar en temas propios del currículo desarrollando los procesos anteriormente descritos, pero fundamentalmente se diseñó este laboratorio con el fin de profundizar en la representación tanto gráfica como algorítmica de los conceptos estudiados y en el planteamiento y resolución de problemas.
En esta primera guía se abordarán varios métodos de solución de sistemas de ecuaciones, como método gráfico, Gauss Jordan, regla de Cramer y por la matriz Inversa. En varios de estos métodos se realizan las operaciones elementales de reducción de renglones en una matriz mediante tratamiento de la matriz en MATLAB.

**OBJETIVOS**

- Utilizar los comandos del software Matlab que permitan solucionar ejercicios propuestos en clase.
- Identificar la representación gráfica como una forma de solución alternativa a la representación algebraica.
- Resolver problemas contextualizados que requieren los conceptos vistos en clase y las funciones propias de Matlab.

| **EQUIPOS Y HARDWARE NECESARIOS**<br>**SI** ☒ **NO** ☐ | **MATERIALES CONSUMIBLES**<br>No aplica |
|---|---|
| **SOFTWARE : SI** ☒ **NO** ☐<br><br>Matlab | **MATERIALES APORTADOS POR EL ESTUDIANTE**<br><br>No aplica |

**PROCEDIMIENTO**:

## PARTE 1. COMANDOS EN MATLAB PARA EL MANEJO DE VECTORES Y MATRICES

En Matlab se puede trabajar desde el command window que es donde se ejecutan las operaciones y comando al dar ENTER. Sin embargo, no podrá guardar ninguno de ellos bajo un archivo. Para guardar los comandos que usted escriba y pueda modificarlos posteriormente o correrlos en otro computador debe escribirlos en un script o editor para esto usted debe abrir un nuevo script con el botón que encuentra en la parte superior izquierda de la pantalla
(Ver Fig. 1) abriendo el menú y luego guardando el archivo con una sola palabra en una carpeta que matlab pueda acceder directamente o existe la posibilidad de crear una nueva ruta a esta carpeta.

INTRODUCCIÓN A VECTORES Y MATRICES

La matriz es el tipo fundamental de dato en MATLAB con números organizados en m filas y n columnas. Un escalar que es un caso particular de matriz de dimensión 1×1. Los vectores son las matrices más simples: un vector fila de m elementos es una matriz de dimensión m×1 y un vector columna de n elementos es una matriz de dimensión 1×n. En la figura 2 se observa una relación conceptual entre escalares, vectores y matrices.

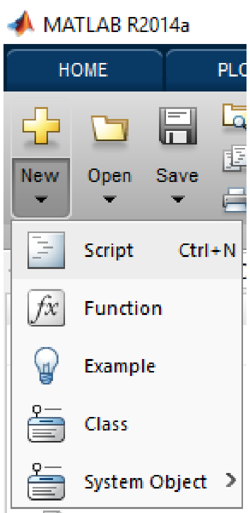



Fig. 1 Menú desplegable para crear Archivos tipo Script o Function

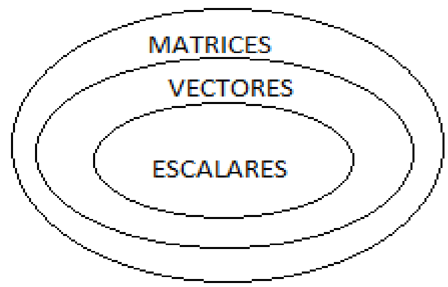



Fig. 2. Relación entre escalares, vectores y matrices.

**Creación de Vectores y matrices en Matlab**. Para crear un vector fila se escribe sus elementos unos a continuación de los otros separados por espacios o comas, y entre paréntesis cuadrados. Para crear un vector columna se escribe los elementos unos a continuación de los otros separados por puntos y comas o bien, en forma columna tal como se indica en el recuadro:

```
>>p=[1 2 3];  %Se crea el vector p con los números 1,2,3
p= 1  2  3
```

También se puede escribir con la estructura: *valor_inicial:incremento:valor_final*, (*xi:Δx:xf*), ejemplo el vector q con los números 2, 4, 6, 8,10:

```
>>q=2:2:10
q =    2    4    6    8    10
```

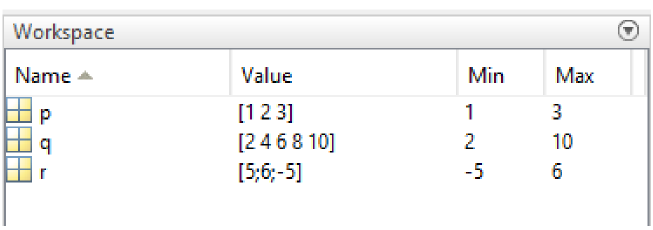
Y también con el comando linspace, el cual genera vectores con sus valores linealmente espaciados, entre un valor inicial *a* y uno final *b* con *n* valores en total: *linspace*(*a,b,n*). Si se omite *n*, por defecto toma 100 elementos.

```
>> r=linspace(1,10,5) %Crea un vector con 5 elementos igualmente espaciados entre 1 y 10
r =     1.0000    3.2500    5.5000    7.7500   10.0000
```

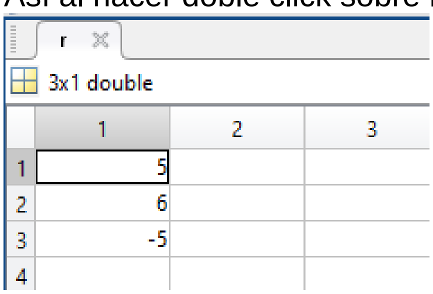
Para crear un vector columna se escribe los elementos unos a continuación de los otros separados por puntos y comas o bien, en forma columna tal como se indica en el cuadro de la derecha:

| | |
|---|---|
| >>R=[5;6;-5] %Dé Enter y observe:<br><br>R =<br>5<br>6<br>-5 | >> R=[5<br>6<br>-5] %Dé Enter y observe<br><br>R =<br>5<br>6<br>-5 |

En la ventana Workspace se visualizan las variables creadas x debajo de *Name* y los valores que guarda, debajo *Value*. Seleccionado la variable x, se pueden cambiar los valores que guarda mediante el Variable Editor, que se abre pulsando el botón del menú Workspace denominado Open selection o haciendo doble-clic en el nombre de la variable.




Así al hacer doble click sobre la variable r, se tiene:




Introducción de matrices. Los elementos de un renglón se separan por espacios y/o comas, y las columnas se separan por ";"

Así para producir la matriz

$$A=\begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{pmatrix}$$

se utilizan cualquiera de las dos formas:

| | |
|---|---|
| >> A=[1 2 3;4 5 6;7 8 9]<br><br>A =<br>1 2 3<br>4 5 6<br>7 8 9 | >> A=[1 2 3;<br>4 5 6;<br>7 8 9]<br><br>A =<br>1 2 3<br>4 5 6<br>7 8 9 |

**Acceso a los elementos de un vector o matriz.** Cuando se crea un vector, por ejemplo **x**=[3,6,9,12,15,18]; la tabla siguiente muestra los índices del vector **x** y los valores que guardan los elementos del vector.

| Índice | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Valor | 3 | 6 | 9 | 12 | 15 | 18 |

De esta forma para necesitar el elemento 4 de este vector **x**, utilice x(4), como se observa en el código

```
>> x=[3,6,9,12,15,18];
>> x(4)
ans =
   12
```

**Notación para formar las submatrices y las matrices aumentadas**.
Dada una Matriz A definida como A=[1 2 3;4 5 6;7 8 9], analice el Script ejecutándolo en Matlab:

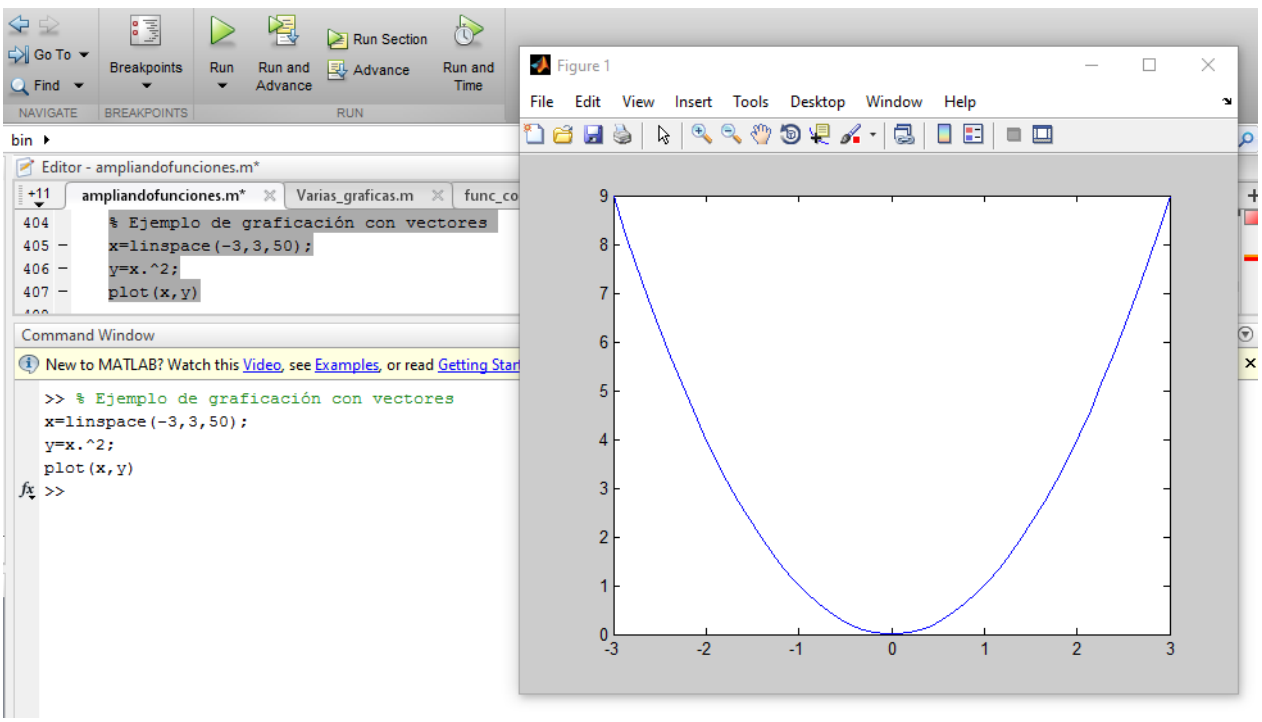
```
A=[1 2 3;4 5 6;7 8 9]
f= A(2,3) %f es el elemento en el segundo renglón, tercera columna de A.
d = A(3,:)  %d es el tercer renglón de A.
g= A(:,3)  % g es la tercera columna de A.
c =A([2 3],:) % c es la matriz de los renglones segundo y cuarto de A.
H = [A b] % Forma una matriz aumentada H = (A|b).
```

Pruebe ejecutando el script, por ejemplo la matriz aumentada H queda:
H =

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 5 |
| 4 | 5 | 6 | 7 |
| 7 | 8 | 9 | 9 |

## PARTE 2: REPRESENTACIÓN GRÁFICA

El uso de vectores es de gran utilidad para representación gráfica. En la figura 2 se observa un script en que se definen un vector x con el comando linspace y con suficientes valores (n=50), y luego otro vector y para los valores de **x** elevados al cuadrado. Al seleccionar el código que interesa ejecutar y presionar F9 (o con el botón run en cuyo caso se ejecuta todo el código que esté en el script), se genera la gráfica. Tenga en cuenta que cuando realice alguna operación con vectores, debe colocar un punto, así en este caso ***y=x.^2***, con lo que se genera otro vector de 50 valores para cada elemento del vector ***x*** elevado al cuadrado.




## PARTE 3: RESOLUCIÓN DE PROBLEMAS

SISTEMAS DE ECUACIONES DE 2X2

Gráficamente se pueden representar un sistema de ecuaciones de 2x2 teniendo en cuenta que cada ecuación es una recta en el plano, luego la intersección representa la solución. En este ejemplo se trabajará de forma simbólica las ecuaciones:

Ejercicio: Los ingenieros Wilson y Camilo hacen sus cuentas de las ganancias obtenidas en un proyecto. Camilo gana dos veces lo que gana Wilson menos un millón de pesos. Para invertir en un segundo proyecto con un valor de $2.000.000, Camilo invierte todas sus ganancias y Wilson cuatro veces sus ganancias del proyecto anterior. Hallar cuánto ganó cada ingeniero en millones de pesos en ese primer proyecto.

Solución. Defina las variables como:
x: Total de ganancias en millones de pesos obtenidas por el ing. Wilson
y: Total de ganancias en millones de pesos obtenidas por el ing. Camilo

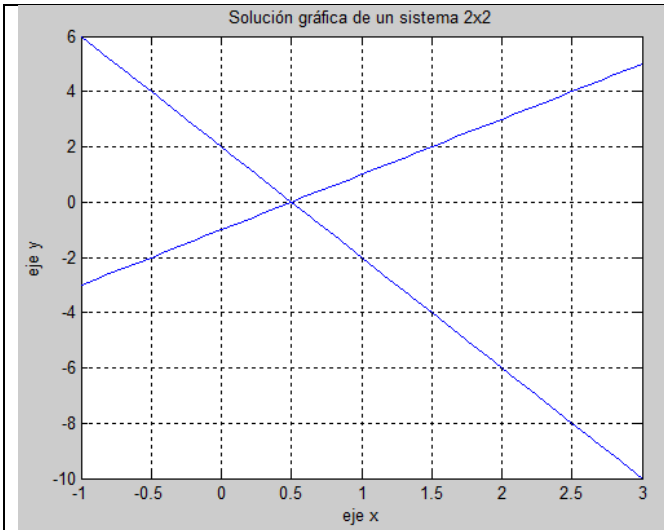
Plantee con la información dada, el sistema de ecuaciones. Luego despeje la variable y de cada ecuación, con lo cual se tiene:

$$\begin{cases} y = 2x - 1 \\ y = -4x + 2 \end{cases}$$

para resolver por el método gráfico, se plantea el siguiente script:

```
%ejemplo de sistemas de ecuaciones de 2x2
x=-1:0.1:3;
y1=2.*x-1;
y2=-4.*x+2;
plot(x,y1)
hold on
plot(x,y2)
title('Solución gráfica de un sistema 2x2')
xlabel('eje x')
ylabel('eje y')
```

Y la gráfica es:




En la cual se observa que la solución es (0.5, 0), es decir Wilson ganó medio millón de pesos y Camilo no obtuvo ganancias.

Para verificar, se puede resolver este sistema de ecuaciones con el comando solve, pero para utilizar este se deben manejar las ecuaciones de forma simbólica (comando syms). Analice el código correspondiente:

```
% resolver el sistema de ecuaciones con el comando solve
syms x y
y1=2*x-1;
y2=-4*x+2;
xsol=solve('2*x-1=-4*x+2')
ysol=subs(y1,xsol)
```

Al ejecutar se observa en el Command Window

xsol =
1/2

ysol =
0

Ya se ha visto cómo resolver sistemas de ecuaciones de 2x2 de forma gráfica y con el comando solve. Ahora se abordarán sistemas de ecuaciones de 3x3 y de mayor tamaño y su resolución con los métodos vistos en clase, como son Gauss Jordan, regla de Cramer y por la Matriz Inversa.

1. Método de Gauss Jordan. En Matlab las operaciones entre renglones de una matriz A, se realizan como:

```
A(2,:) = 3*A(2,:)
A(2,:) = A(2,:)/4
A([2 3],:) = A([3 2],:)
A(3,:) = A(3,:) + 3*A(2,:)
```

$R_2 \to 3R_2$

$R_2 \to \frac{1}{4}R_2$

Intercambia los renglones 2 y 3

$R_3 \to R_3 + 3R_2$

Observación: Todos estos comandos cambian a la matriz A. Si se quiere conservar la matriz original y llamar a C a la matriz cambiada, haga

C = A

C(2,:)= 3*C(2,:)   Operación: R2 <- 3R2

C = rref(A)   C= forma escalonada reducida por renglones de A.

Ejercicio: resolver por Gauss Jordan, el sistema de ecuaciones:

$$\begin{cases} x + y + z = 6 \\ 2x - y + 2z = 6 \\ x + 2y - z = 2 \end{cases}$$

Observe el código y los resultados de cada paso a la derecha:

| | |
|---|---|
| %RESOLUCION DE UN SISTEMA DE ECUACIONES DE 3X3<br>%1. METODO DE GAUSS JORDAN<br>A=[1 1 1;2 -1 2;1 2 -1]  %Matriz A de coefic.<br>b=[6;6;2] %Vector b de términos independientes<br>C=[A b] %Matriz C Aumentada que representa el<br>%sistema de ecuaciones Lineales. | A =<br>1 1 1<br>2 -1 2<br>1 2 -1<br><br>b =<br>6<br>6<br>2<br><br>C =<br>1 1 1 6<br>2 -1 2 6<br>1 2 -1 2 |
| C(2,:)=C(2,:)-2*C(1,:) % R2<- R2-2R1 | C =<br>1 1 1 6<br>0 -3 0 -6<br>1 2 -1 2 |
| C(3,:)=C(3,:)-C(1,:)  % R3<- R3 - R1 | C =<br>1 1 1 6<br>0 -3 0 -6<br>0 1 -2 -4 |
| % Intercambiar dos renglones:R2 por R3<br>C([2 3],:)=C([3 2],:) | C =<br>1 1 1 6<br>0 1 -2 -4<br>0 -3 0 -6 |

Siga el procedimiento hasta llegar a la matriz reducida:

C=

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 0 | 1 | 3 |

Puede probar este resultado con el comando **rref(A)**

De la cual se encuentra la solución al sistema como x=1, y=2, z=3, o (1, 2, 3)

**Matriz Inversa.** Para la matriz inversa de la matriz A, se creará una matriz D aumentada de la matriz A con la matriz identidad de 3x3, esta última se crea con el comando eye(3). Realice los mismos pasos para resolver el sistema de ecuaciones, cambiando la C por D. Ejecute cada línea de código y revise los resultados, luego complete los dos pasos que faltan para completar el procedimiento:

```
%Inversa de A
format rat
A=[1 1 1;2 -1 2;1 2 -1]  %Matriz A de coeficientes
b=[6;6;2] %Vector b de términos independientes
D=[A eye(3)] %Matriz aumentada con la matriz identidad de 3x3
D(2,:)=D(2,:)-2*D(1,:) % R2<- R2-2R1
D(3,:)=D(3,:)-D(1,:)  % R3<- R3 - R1
% Intercambiar dos renglones: R2 por R3
D([2 3],:)=D([3 2],:)
D(1,:)=D(1,:)-D(2,:) % R1<- R1-R2
D(3,:)=D(3,:)+3*D(2,:) % R3<- R3+3R2
D(3,:)=D(3,:)/(-6)  % R3<- R3/(-6)
```

Continúe con los 2 pasos que faltan y hallará la matriz inversa, su último resultado debe ser

```
1    0    0    -1/2    1/2    1/2
0    1    0     2/3   -1/3    0
0    0    1     5/6   -1/6   -1/2
```

Luego la matriz inversa (llámela invA) es:

```
invA =

   -1/2     1/2     1/2
    2/3    -1/3     0
    5/6    -1/6    -1/2
```

Una vez se tiene la inversa, se encuentra la solución al sistema de ecuaciones como $\mathbf{X}=\mathbf{A}^{-1}\mathbf{b}$, donde el vector **X**=[x; y; z], es decir, ejecute X=invA*b ya que invA es la matriz inversa de A, observe el resultado en el Command Window (parte derecha de la siguiente tabla).

| | |
|---|---|
| X=invA*b | X =<br>1.0000<br>2.0000<br>3.0000 |

**Procedimiento general para hallar la matriz inversa de 3x3 por Gauss Jordan con algo de programación ( Opcional)**. Como se puede observar en el procedimiento anterior, es necesario saber los resultados en cada paso para determinar qué hacer a continuación. En el siguiente script se muestra un script que plantea un procedimiento más general y sistemático en el cual se requieren elementos básicos de programación con la estructura condicional if – else- end. Puede probar con el mismo sistema de problema anterior y luego ingresando otro sistema de ecuaciones de 3x3. Advertencia: El código es una primera aproximación para la solución general de la matriz inversa de 3x3.

```
%Solución sist. 3x3 por reducción de renglones por medio de la matriz Inversa
%Creado por ing. Wilson Castro Z.
format rat
%Paso 0: Escriba la matriz del sistema y el vector b.
A=[1 1 1;2 -1 2;1 2 -1]  %Matriz A de coeficientes
b=[6;6;2]; %Vector b de términos independientes
D=[A eye(3)] %Matriz aumentada con la matriz identidad de 3x3
%Paso 1 divida primer renglón por elemento D(1,1), para tomar pivote
if (D(1,1)==0)
    % No requiere dividir
else
D=D/D(1,1)
end
%Paso 2. R2<- R2-D(2,1)*R1
D(2,:)=D(2,:)-D(2,1)*D(1,:)
%Paso 3. R3<- R3 - D(3,1)*R1
D(3,:)=D(3,:)-D(3,1)*D(1,:)
% Paso 4. Probar si ya hay un pivote (1) en los renglones 2 o 3
if (D(2,2)==1) %Toma pivote al renglón 2
elseif (D(3,2)==1);  %toma pivote al renglón 3
% Se deben intercambiar los renglones R2 por R3
D([2 3],:)=D([3 2],:)
else %No hay pivote, crearlo para el renglón 2
   if (D(2,2)==0)
   else
    D(2,:)=D(2,:)/D(2,2)
   end
end
% Paso 5. Realizar la reducción de renglones con el pivote en D(2,2)
D(1,:)=D(1,:)-D(1,2)*D(2,:) % R1<- R1-aR2
D(3,:)=D(3,:)-D(3,2)*D(2,:) % R3<- R3-bR2
% Paso 6. Pivote final en D(3,3)
if (D(3,3)==0)
else
D(3,:)=D(3,:)/D(3,3)  % R3<- R3/c
end
%Paso 7. Reducción de los renglones R1 y R2
D(1,:)=D(1,:)-D(1,3)*D(3,:) % R1<- R1-eR3
D(2,:)=D(2,:)-D(2,3)*D(3,:) % R2<- R2-fR3
%Paso 8. Captura de la matriz inversa
invA=D(:,4:6)
% ¨Paso 9. Solución del sistema de ecuaciones
x=invA*b
```

**Regla de Cramer.** En este método se requieren las matrices Ax, Ay y Az que son en cada caso las matrices de A en que se sustituye la columna x, y, z respectivamente por el vector b. Luego la solución de cada variable es el resultado de la división entre el determinante de cada una de estas matrices y el determinante del sistema:

$$x = \frac{|Ax|}{|A|}, y = \frac{|Ay|}{|A|}, z = \frac{|Az|}{|A|}$$

Ejercicio: Resolver por la regla de Cramer el sistema:

$$\begin{cases} x + y + z = 6 \\ 2x - y + 2z = 6 \\ x + 2y - z = 2 \end{cases}$$

Analice el código siguiente y observe los resultados que se presentan en cada parte, a la derecha.

| | |
|---|---|
| `%Método de Regla de Cramer`<br>`%Código desarrollado por Ing. Wilson Castro Z.`<br>`A=[1 1 1;2 -1 2;1 2 -1];  %Matriz A de coef.`<br>`b=[6;6;2]; %Vector b de términos independientes`<br>`Ax=[b A(:,2:3)]` | Ax =<br>6 1 1<br>6 -1 2<br>2 2 -1 |
| `Ay=[A(:,1) b A(:,3)]` | Ay =<br>1 6 1<br>2 6 2<br>1 2 -1 |
| `Az=[A(:,1:2) b]` | Az =<br>1 1 6<br>2 -1 6<br>1 2 2 |
| `%Vector solución X=[x,y,z], es:`<br>`X=[det(Ax)/det(A),det(Ay)/det(A),det(Az)/det(A)]` | X =<br>1.0000 2.0000 3.0000 |

**Método de la Matriz Inversa a través de la Matriz Adjunta**. En este caso se debe hallar la matriz **B** de cofactores. Para orientarse sobre el correcto desarrollo de los cofactores, escriba a lápiz y papel cada una de las matrices menores, luego digite el código correspondiente para el cofactor, podría revisar la matriz menor seleccionando del código esta parte y ejecutándola con F9. Así, por ejemplo, el código para el primer cofactor y la ejecución de la matriz menor correspondiente es:

```
C(1,1)=(-1)^(1+1)*det([A(2:3,2:3)])
```

```
>> [A(2:3,2:3)]

ans =

    -1     2
     2    -1
```

**Nota**: Si se representa una falla en el proceso y Matlab se queda esperando un comando y no responde, presiones los botones Ctrl + C para suspender la ejecución.

En este método se halla primero la matriz B de cofactores, y la matriz inversa se encuentra con la fórmula:

$$A^{-1} = \frac{1}{|A|} B'$$

Donde B' es la traspuesta de B.
Observe el código siguiente para los primeros cofactores:

```
%Método de los cofactores
% Código desarrollado por Ing. Wilson Castro Z.
A=[1 1 1;2 -1 2;1 2 -1];  %Matriz A de coeficientes
C(1,1)=(-1)^(1+1)*det([A(2:3,2:3)]);
C(1,2)=(-1)^(1+2)*det([A(2:3,1) A(2:3,3)]);
C(1,3)=(-1)^(1+3)*det([A(2:3,1:2)]);
C(2,1)=(-1)^(2+1)*det([A(1,2:3);A(3,2:3)]);
C(2,2)=(-1)^(2+2)*det([A(1,1) A(1,3);A(3,1) A(3,3)]);
%Continúe con los demás cofactores hasta
C(3,3)=(-1)^(3+3)*det([A(1:2,1:2)]);
%Luego la Matriz B de Cofactores es:
B=[C(1,1) C(1,2) C(1,3);C(2,1) C(2,2) C(2,3);C(3,1) C(3,2) C(3,3)]
%Y la matriz Inversa de A por medio de la Matriz Adjunta, es
invA=(1/det(A))*B'
```

Si halló bien los cofactores, la matriz **B** es de la forma:

```
>>B =
   -3    4    5
    3   -2   -1
    3    0   -3
```

Y la Matriz Inversa ($A^{-1}$) da:

```
invA =

      -1/2           1/2           1/2
       2/3          -1/3           0
       5/6          -1/6          -1/2
```

Puede verificar el resultado directamente con el comando **inv(A).**

## PARTE 4: EJERCICIOS PROPUESTOS

1. Resuelva los sistemas de ecuaciones por el método gráfico y con solve (despeje la *y* en cada caso):

   a) $\begin{cases} x - y = 2 \\ x - 4y = -4 \end{cases}$   b) $\begin{cases} x + 4y = 16 \\ x - y = 1 \end{cases}$

2. Resuelva los sistemas de ecuaciones de 3x3 utilizando los métodos estudiados,

con al menos un método por problema (aplique Gauss Jordan y luego la matriz inversa en el mismo ejercicio):

a) $\begin{cases} 2x+4y+6z=18 \\ 4x+5y+6z=24 \\ 3x+\ y-2z=\ \ 4 \end{cases}$ $b)\begin{cases} x-2y+3z=11 \\ 4x+\ y-\ z=\ \ 4 \\ 2x-y+3z=10 \end{cases}$ $c)\begin{cases} x_1+\ x_2-\ x_3=\ 7 \\ 4x_1-\ x_2+5x_3=4 \\ 2x_1+2x_2-3x_3=0 \end{cases}$

3. Para alguno de los sistemas resueltos anteriormente, halle la matriz inversa por medio de la adjunta.

**ANÁLISIS DE DATOS Y RESULTADOS**:

- Escriba las respuestas de cada una de las preguntas propuestas.

Como conclusión de la práctica considere lo siguiente:
- ¿Qué dificultades se le presentaron en el desarrollo de la guía?
- Escriba en qué parte de la temática, la práctica le permitió aclarar las dudas que se le presentaron en el salón de clase.

**BIBLIOGRAFÍA**:

www.mathworks.com, mayo a junio 2017.

Badia, A. (2016). La percepción de la utilidad de la tecnología conforma su uso para enseñar y aprender. *Revista Electrónica de Investigación Educativa*, *18*, 95–105.

David Báez López, *MATLAB con Aplicaciones a la Ingeniería, Física y Finanzas*; 1ra. Edición, ed. Alfaomega grupo editor, 2006.

Stanley Grossman1997). Álgebra lineal (5 Edición). Mc Graw Hill.

Vega, Héctor Manuel (2006). *Lógica y algoritmos de programación en Matlab aplicada a la ingeniería*. Editorial Bonaventuriana.

USB. (2010). *Modelo Pedagógico - Referentes conceptuales, lineamientos curriculares y de flexibilidad*. *Editorial Bonaventuriana* (Vol. 1). http://doi.org/10.1017/CBO9781107415324.004

**INFORME DE PRÁCTICA DE LABORATORIO**

Para la entrega del informe de la práctica de laboratorio los estudiantes deben seguir los siguientes pasos:

- Diligenciar el formato (Word) del informe de laboratorio
- Guardar el archivo de Matlab (.m) de la práctica
- Ingresar al aula virtual de la universidad USBBOG
- Ingresar al curso o asignatura a la cual corresponde el laboratorio
- Ingresar al foro correspondiente a la práctica
- Publicar en el foro el informe de la práctica (Word) y el archivo de Matlab (.m) de esta.