


```

1 33tic
2 clc
3 clear all
4
5 %A=input('Introducir valores a:')
6 %B=input('Introducir valores b:')
7
8 %Introducir valores de la matriz
9
10 fi=input('Ingresa la cantidad de filas: '); %Se le pedira al usuario que in
11 co=input('Ingresa la cantidad de columnas: ');
12
13 for i=1:co %Guardamos la matriz de la misma manera que en el ejemplo de g
14     for j=1:fi
15         fprintf('Fila: %x\n', j)
16         fprintf('Columna: %x', i)
17         r= input ('Numero de fila y columna: ');
18         A(j,i)=r;
19         j=j+1;
20     end
21     i=i+1;
22 end
23 A

```

Command Window

Numero de fila y columna:
1

Numero de fila y columna:
4

Numero de fila y columna:
-1

Sistema a solucionar							b									
1	x	-1	y	1	z	=	1									
1	x	1	y	1	z	=	9	No. Iter	x	y	z	err x	err y	err z		
4	x	-2	y	1	z	=	0	0	0	0	0	x inicial				
Verificamos que la matriz sea								1	1	9	0					
diagonal dominante								2	10	8	14	0,900000000	0,125000000	1,000000000		
								3	-5	-15	-24	-3,000000000	-1,533333333	-1,583333333		
								4	10	38	-10	1,500000000	1,394736842	-1,400000000		
Fila 1	Valor inicial							1	5	49	9	36	0,795918367	3,222222222	1,277777778	
Fila 2	Valor inicial							1	6	-26	-76	-178	-2,884615385	-1,118421053	-1,202247191	
Fila 3	Valor inicial							1	7	103	213	-48	1,252427184	1,356807512	-2,708333333	
								8	262	-46	14	0,606870229	-5,630434783	4,428571429		
Fila 1	suma valores restantes							#ERROR!	9	-59	-267	-1140	-5,440677966	-0,827715356	-1,012280702	
Fila 2	suma valores restantes							#ERROR!	10	874	1208	-298	1,067505721	1,221026490	-2,825503356	
Fila 3	suma valores restantes							#ERROR!	11	1507	-567	-1080	0,420039814	-3,130511464	-0,724074074	
								COMP EC1	17	1507	-2	-567	-3	-1080	29993	
Situacion #ERROR!								COMP EC2	-5	1507	21	-567	-2	-1080	-17282	
								COMP EC3	-5	1507	-5	-567	22	-1080	-28460	

JewthonRap.m x +

```
1 clear, clc
2 f = input('f(x)= ', 's');%'x^2+ 64*x + 4'
3 sf = str2sym (f); %convierte un string en una funcion
4 tol = input('tolerancia del metodo = ');%tolerancia
5 x0 = input('valor inicial = ');%valor inicial, ejemplo: -3
6 v = symvar(sf);%extrae las variables de la funcion
7 f1 = diff(sf); %calcula la derivada de una funcion
8 sf
9 v
0 f1
1
2 y=subs (sf,v,x0)
3 z=subs(f1,v,x0)
4 y
5 z
6
7 sw = 0;
8 while (sw==0)
9     %subs es una funcion que reemplaza valores constantes en las variables
0     %de una funcion
1     x1 = x0 - (subs (sf, v, x0) / subs (f1, v, x0));
2     fprintf ('\t x0 %f \t x1 %f \t tol %f \n', x0, x1, tol)
3     if abs(x0 - x1) > tol %aun no se encuentra la raiz
4         x0 = x1;
5         sw=0;
6
7     else %raiz encontrada
8         sw =1;
9     end
0 end
1
2 vpa(x1)
```

```
%% Jacobi Method
%% Solution of x in Ax=b using Jacobi Method
% * Initialize 'A' 'b' & initial guess 'x'
%%
```

```
A=[ 1 -1 1 1; 1 1 1 9; 4 -2 1 0]
b=[-1 2 3 0.5]'
x=[0 0 0 0]'
```

```
% A=[ 17 -2 -3;
%     -5 21 -2;
%     -5 -5 22]
%b=[500 200 30]'
%x=[0 0 0]'
```

```
n=size(x,1);
normVal=Inf;
%%
% * Tolerance for method
tol=1e-5; itr=0;
%% Algorithm: Jacobi Method
%%
```

```
while normVal>tol
    xold=x;
```

```
    for i=1:n
        sigma=0;
```

```
        for j=1:n
```

A = 3x4

```
1   -1   1   1
1    1   1   9
4   -2   1   0
```

b = 4x1

```
-1.0000
2.0000
3.0000
0.5000
```

x = 4x1

```
0
0
0
0
```

ID WINDOW

50

```
while normVal>tol
    xold=x;
```

```
    for i=1:n
        sigma=0;
```

```
        for j=1:n
```

```
            if j~=i
                sigma=sigma+A(i,j)*x(j);
            end
```

```
        end
```

```
        x(i)=(1/A(i,i))*(b(i)-sigma);
    end
```

```
    itr=itr+1;
    normVal=abs(xold-x);
```

```
end
```

50

```
fprintf('Solution of the system is : \n%f\n%f\n%f\n%f in %d iterations',x,itr)
```

[illegible]