1. Ejercicio de equilibrio estático.

Para cierto sistema masa resorte se deben satisfacer las siguientes ecuaciones de equilibrio estático.

$(k_1 + k_2 + k_3 + k_5) \cdot x_1 - k_3 \cdot x_2 - k_5 \cdot x_3 = m_1 \cdot g$

$-k_3 \cdot x_1 + (k_3 + k_4) \cdot x_2 - k_4 \cdot x_3 = m_2 \cdot g$

$-k_5 \cdot x_1 - k_4 \cdot x_2 + (k_4 + k_5) \cdot x_3 = m_3 \cdot g$

siendo $k_1 = k_3 = k_4 = k,\ k_2 = k_5 = 5k,\ m_1 = m_3 = 2m,\ m_2 = m$

- ***Código de Octave***

```
function [x]=desplazamientoElas(m,k)

%% La funcion me da como resultado los valores de los

%% desplazamientos elasticos.

%% - k es la constante del resorte.

%% - m es la masa del resorte.

%% Se resuelve el sistema mediante el metodo iterativo de Gauss-Seidel.

%%Entrada

k=input("Ingrese el valos de k:");

m=input("Ingrese el valor de m:");

%%Proceso

if k,m>0

A=[12*k,-k,-5*k;-k,2*k,-k;-5*k,-k,6*k];

b=[19.62*m;9.81*m;19.62*m];

x0=[0;0;0];

tol=1E-3;

x=gs(A,b,x0,tol); (*)

%%Salida

disp("Los valores de los desplazamientos elasticos son:");
```

```
else
disp("Los valores de k y m no son mayores a cero")
endif
endfunction
```

(*)Programa ocupado para resolver el problema anterior.

```
function [x]=gs(A,b,x0,tol)
%% El siguiente programa resuelve el SEL
%% por el método iterativo de Gauss-Seidel.
[n,n]=size(A);
x=zeros(n,1);
erro=999;
%% Obtengo la matriz diagonal de A.
for i=1:n
D(i,i)=A(i,i);
endfor
%% Obtengo la matriz triangular estrictamente inferior.
Tinf=tril(A,-1);
%% Obtengo la matriz triangular estrictamente superior.
Tsup=triu(A,1);
An=(D+Tinf);
while erro>tol
bn=b-Tsup*x0;
%% Aplicamos sustitucion progresiva.
x(1)=bn(1)/An(1,1);
for k=2:n
x(k)=(bn(k)-An(k,1:k-1)*x(1:k-1))/An(k,k);
```

endfor

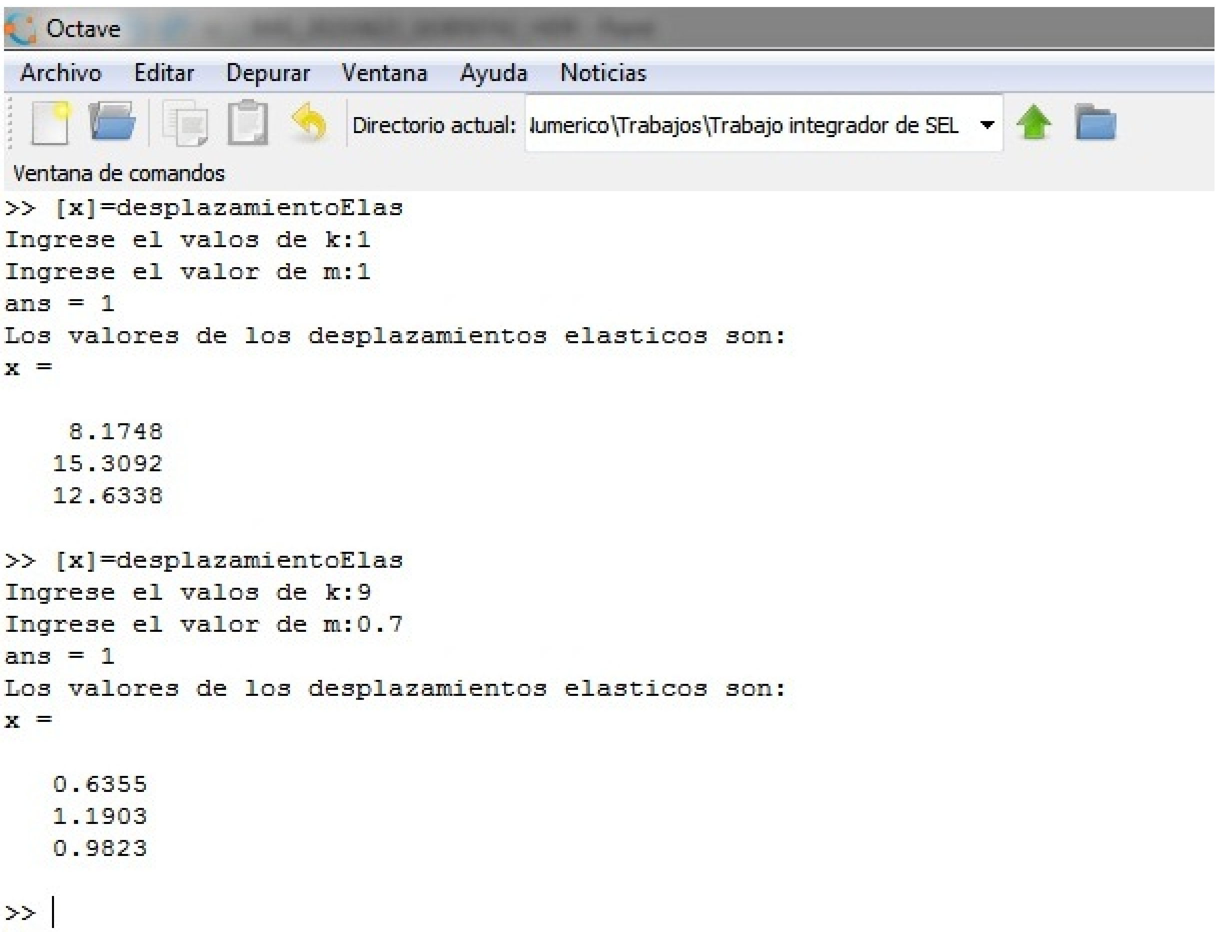
erro=norm((x-x0),1);

x0=x;

endwhile

endfunction

- ***Captura del programa en ejecución***



- El método que se utiliza para resolver el problema es el método iterativo de Gauss-Seidel

2. Ejercicio de equilibrio estático.

Los desplazamientos de un cierto reticulado plano se pueden determinar mediante el conjunto de ecuaciones de equilibrio estático que se presentan a continuación.

$27,58E6u_1 + 7,004E6u_2 - 7,0044E6u_3 = 0$

$7,004E6u_1 + 29,57E6u_2 - 5,253E6u_3 - 24,32E6u_5 = 0$

$-7,004E6u_1 - 5,253E6u_2 + 29,57E6u_3 = 0$

$27,58E6u_4 - 7,004E6u_5 = 0$

$-24,31E6u_2 - 7,004E6u_4 + 29,57E6u_5 = -47000[N]$

donde los coeficientes numéricos que multiplican a los desplazamientos expresan rigidez en (N/m).

- ***Código de Octave***

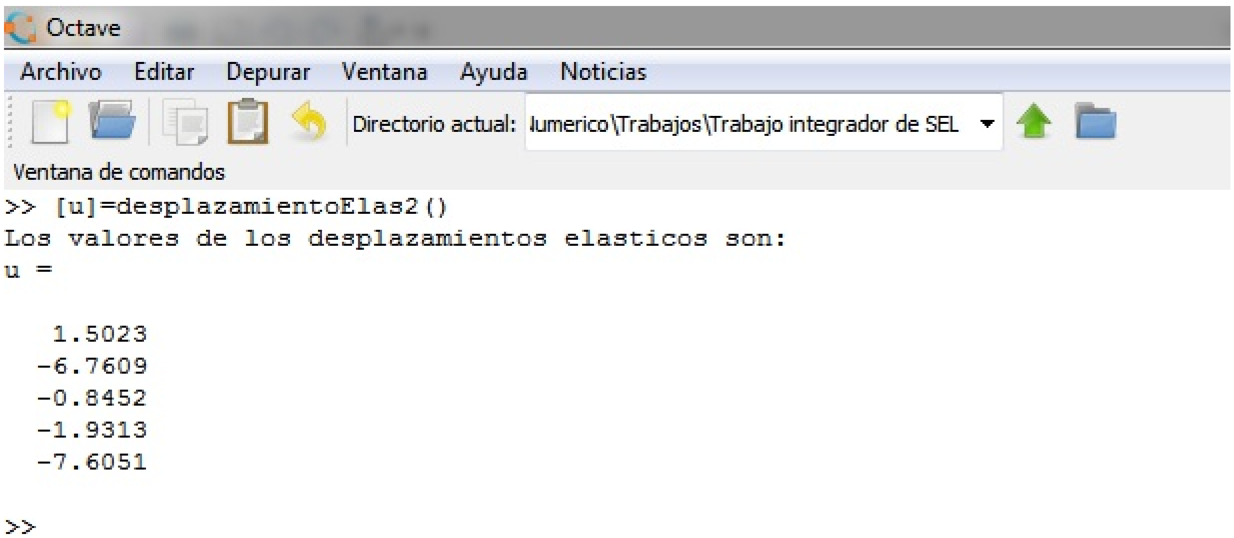
```
function [u]=desplazamientoElas2()

%% La funcion me da como resultado los valores de los

%% desplazamientos elasticos.

%% Se resuelve el sistema mediante triangulacion superior seguida

%% de una sustitucion regresiva.

%%Entrada

A=[27580,7004,-7004,0,0;7004,29570,-5253,0,-24320;-7004,-5253,29570,0,0;0,0,0,27580,-7004;0,-24310,0,-7004,29570];

B=[0;0;0;0;-47000];

%%Proceso

%%Resolvemos Au=B mediante triangulacion superior seguida de sustitucion regresiva.

u=tSSR(A,B); (*)

%%Salida

disp("Los valores de los desplazamientos elasticos son:");

endfunction
```

(*) Programa utilizado para resolver el problema anterior.

```
function [X]=tSSR(A,B)
%%Ingreso - A matriz N x N no singular
%% - B es una matriz N x 1
%%salida - X es matriz N x 1 con la solución de AX=B.
%%Iniciaalizar X y la matriz temporario de almacenaje C
[N N]=size(A);
X=zeros(N,1);
C=zeros(1,N+1);
%%Forma la matriz aumentada: Aug=[A|B]
Aug=[A B];
for p=1:N-1
%%Pivoteo parcial para columna p
[Y,j]=max(abs(Aug(p:N,p)));
%%Intercambio fila p y j
C=Aug(p,:);
Aug(p,:)=Aug(j+p-1,:);
Aug(j+p-1,:)=C;
if Aug(p,p)==0
'A es singular. No hay solución única'
break
endif
%%etapa de eliminación para columna p
for k=p+1:N
m=Aug(k,p)/Aug(p,p);
Aug(k,p:N+1)=Aug(k,p:N+1)-m*Aug(p,p:N+1);
```

```
endfor
endfor
%%sustituación hacia atrás en [U|Y]
X=sustRegre(Aug(1:N,1:N),Aug(1:N,N+1));
endfunction
```

- ***Captura del programa en ejecución***



- El método que se utilizo para resolver el problema es el método de triangulación superior seguda de sustitucion regresiva.

3. Ejercicio de circuitos eléctricos

En un circuito "puente de Wheatstone", las leyes de Kirchoff establecen que en cada lazo del circuito la suma de las caídas de tensión deben igualar a la suma de las fuentes de tensión. Para una configuración dada se obtiene el siguiente conjunto de ecuaciones.

$$(50 + R) \cdot i_1 - R \cdot i_2 - 30 \cdot i_3 = 0$$
$$-R \cdot i_1 + (65 + R) \cdot i_2 - 15 \cdot i_3 = 0$$
$$-30 \cdot i_1 - 15 \cdot i_2 + 45 \cdot i_3 = 120$$

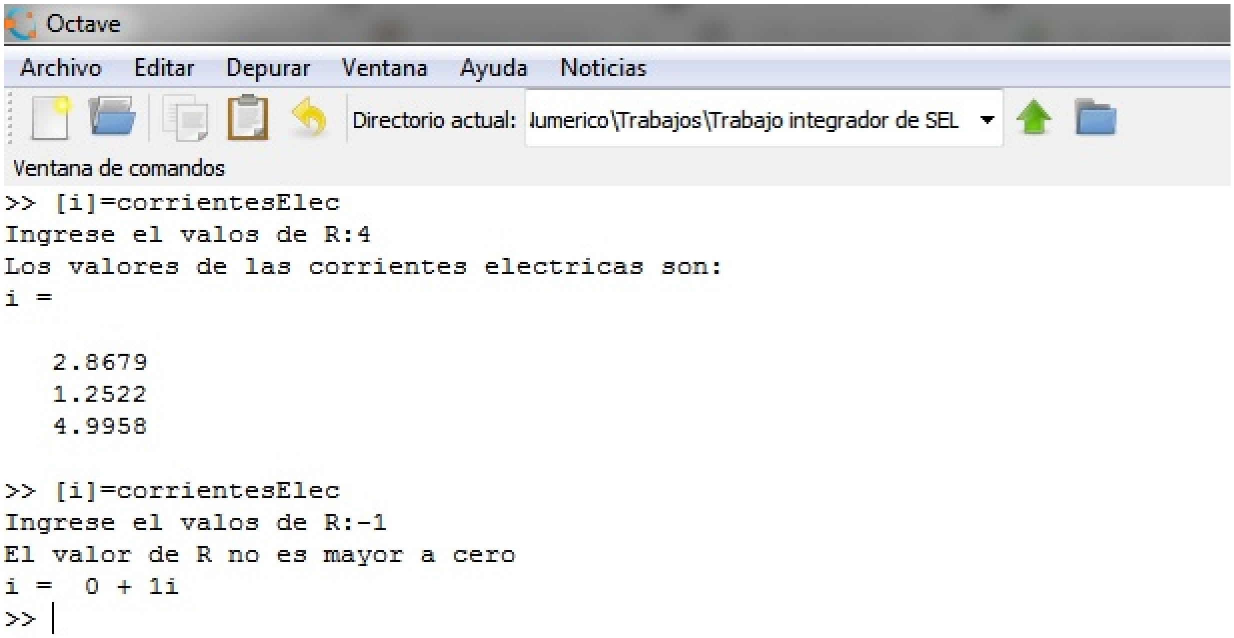
- ***Código de Octave***

```
function [i]=corrientesElec(R)

%% La funcion me da como resultado los valores de las

%% corrientes electricas.

%% - R es la resistencia.

%% Se resuelve el sistema mediante el metodo iterativo de Jacobi.

%%Entrada

R=input("Ingrese el valos de R:");

%%Proceso

if R>0

A=[(50+R),-R,-30;-R,(65+R),-15;-30,-15,45];

b=[0;0;120];

x0=[0;0;0];

tol=1E-3;

i=jacobi(A,b,x0,tol); (*)

%%Salida

disp("Los valores de las corrientes electricas son:");

else

disp("El valor de R no es mayor a cero")

endif

endfunction
```

(*) Programa utilizado para resolver el problema anterior.

```
function [x]=jacobi(A,b,x0,tol)
%% El siguiente programa resuelve el SEL por el método iterativo de Jacobi.
%%Entrada
[N,N]=size(A);
for i=1:N;
D(i,i)=A(i,i);
Dinv(i,i)=1/A(i,i);
endfor
R=A-D;
c=Dinv*b;
T=-Dinv*R;
errort=999;
while errort>tol
x=c+T*x0;
errort=norm((x-x0),1);
x0=x;
endwhile
endfunction
```

- ***Captura del programa en ejecución***



- El método que se ultilizo para resolver el problema es el método iterativo de Jacobi.

4. Ejercicio de mecánica de fluidos

Para un conjunto de tanques interconectados, donde se bombea una solución de agua y un químico determinado, el principio de conservación de la masa establece que, para cada tanque la suma de los caudales másicos de entrada debe ser igual a la suma de los caudales másicos de salida, es decir, $\sum \dot{m}_i^e = \sum \dot{m}_i^s,\ m_i \doteq q_i.c_i$ Donde q_i es el caudal volumétrico y c_i es la concentración del químico. Si se expresa el principio para cada tanque se obtiene el siguiente conjunto de ecuaciones.

$4 \cdot 20 + 4 \cdot c_2 = 8 \cdot c_1$
$8 \cdot c_1 + 2 \cdot c_3 = (4+6) \cdot c_2$
$6 \cdot c_2 + 5 \cdot c_4 = (2+3+6) \cdot c_3$
$3 \cdot c_3 + 4 \cdot c_5 = (2+5) c_4$
$2 \cdot 15 + 2 \cdot c_4 = 4 \cdot c_5$

- ***Código de Octave***

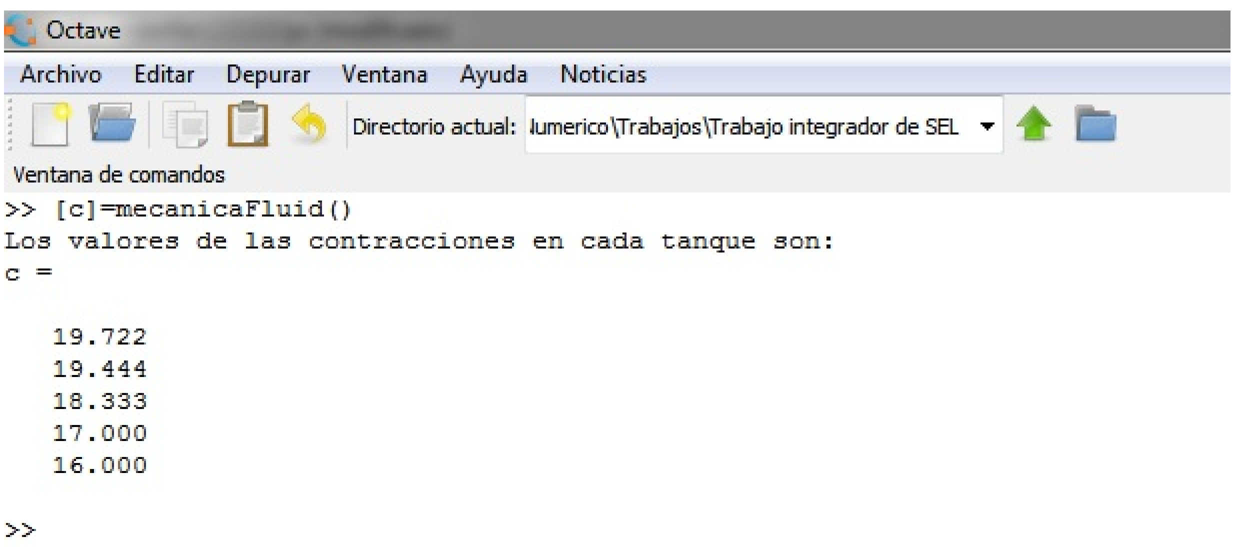
```
function [c]=mecanicaFluid()
    %% La funcion me da como resultado los valores de las
    %% contracciones en cada tanque.
    %% Se resuelve el sistema mediante triangulacion superior seguida
    %% de una sustitucion regresiva.
    %%Entrada
    A=[8,-4,0,0,0;8,-10,2,0,0;0,6,-11,5,0;0,0,3,-7,4;0,0,0,-2,4];
    B=[80;0;0;0;30];
    %%Proceso
    %%Resolvemos Ac=B mediante triangulacion superior seguida de sustitu-
cion regresiva.
    c=tSSR(A,B); (*)
    %%Salida
    disp("Los valores de las contracciones en cada tanque son:");
    endfunction
```

(*) Programa utilizado para resolver el problema anterior.

```
function [X]=tSSR(A,B)

%%Ingreso - A matriz N x N no singular

%% - B es una matriz N x 1

%%salida - X es matriz N x 1 con la solución de AX=B.

%%Iniciaalizar X y la matriz temporario de almacenaje C

[N N]=size(A);

X=zeros(N,1);

C=zeros(1,N+1);

%%Forma la matriz aumentada: Aug=[A|B]

Aug=[A B];

for p=1:N-1

%%Pivoteo parcial para columna p

[Y,j]=max(abs(Aug(p:N,p)));

%%Intercambio fila p y j

C=Aug(p,:);
```

```
Aug(p,:)=Aug(j+p-1,:);
Aug(j+p-1,:)=C;
if Aug(p,p)==0
'A es singular. No hay solución única'
break
endif
%%etapa de eliminación para columna p
for k=p+1:N
m=Aug(k,p)/Aug(p,p);
Aug(k,p:N+1)=Aug(k,p:N+1)-m*Aug(p,p:N+1);
endfor
endfor
%%sustituación hacia atrás en [U|Y]
X=sustRegre(Aug(1:N,1:N),Aug(1:N,N+1));
endfunction
```

- ***Captura del programa en ejecución***



- El método que se utilizo para resolver el problema es el método de triangulación superior seguda de sustitucion regresiva.