

## PROGRAMA ELABORADO POR : ÑOL IVAN JUAN DE DIOS ROJAS (TJGO)

- Programa para la solución de sistemas de ecuaciones no lineales (SENL), por el metodo de Broyden.

```
clc;clear;close all;syms x;syms y;syms z;syms x1;syms x2;syms y1;syms y2;syms z1;syms z2;
% =====
text = {'Ingrese N° de ecuaciones: ', 'Cantidad de variable:', 'Valores iniciales, Entre espacios:'};
dlgtitle = 'S.E.N.O. - NEWTON RHAPSON';
dimensions = [1 18; 1 50; 1 40; 1 29; 1 60];
default_input = { '3', '3', '0.1 0.1 -0.1', '10^-8', '10' };
answer = inputdlg(text,dlgtitle,dimensions,default_input);
num_equ=str2num(answer{1}) ; num_var=str2num(answer{3}) ;
% =====
% =====
eques={'1'};
% vari={'x'};
diff_eques=["-", "-"];
% =====
% =====
% =====
% for j=1:1:num_equ
% j_equ = num2str(j) ;
% text_var = {j_equ};
% dlgtitle_var = 'INGRESE LA VARIABLE N°';
% dimensions_var = [1 50];
% default_input_var = { '0' };
% answer_var = inputdlg(text_var,dlgtitle_var,dimensions_var,default_input_var);
% vari{j}=answer_var;
% str2sym(vari{j})
% end

% =====
% =====
% =====
symss={'y'; 'y'};
matrixx=str2sym(symss);

for i=1:1:num_equ
i_equ = num2str(i);
text_equ = {i_equ};
dlgtitle_eques = 'INGRESE LA ECUACIÓN N°';
dimensions_eques = [1 50];
default_input_equ = { '0' };
answer_eques = inputdlg(text_equ ,dlgtitle_eques,dimensions_eques,default_input_equ);
eques{i}=answer_eques;
diff_eques(i)=string(eques{i});
matrixx(i)=eval(diff_eques(i)');
end
```

```
matrixx % MATRIZ DEL SISTEMA DE ECUACIONES
```

```
matrixx =
```

$$\begin{pmatrix} 3x - \cos(yz) - \frac{1}{2} \\ \sin(z) - 81 \left(y + \frac{1}{10}\right)^2 + x^2 + \frac{53}{50} \\ 20z + e^{-xy} + \frac{5332248173269055}{562949953421312} \end{pmatrix}$$

```
MatrizJacobiana = jacobian(matrixx) % JACOBIANO DEL SISTEMA DE ECUACIONES
```

```
MatrizJacobiana =
```

$$\begin{pmatrix} 3 & z \sin(yz) & y \sin(yz) \\ 2x & -162y - \frac{81}{5} & \cos(z) \\ -ye^{-xy} & -xe^{-xy} & 20 \end{pmatrix}$$

```
x=num_var(1); %#ok<NASGU> % OBTENEMOS EL VALOR X INICIAL  
y=num_var(2); %#ok<NASGU> % OBTENEMOS EL VALOR Y INICIAL  
z=num_var(3); %#ok<NASGU> % OBTENEMOS EL VALOR Z INICIAL  
nn=eval(matrixx) % EVALUAMOS EL SISTEMA CON LOS VALORES INICIALES
```

```
nn = 3x1  
-1.2000  
-2.2698  
8.4620
```

```
mm=eval(MatrizJacobiana) % EVALUAMOS EL JACOBIANO DEL SISTEMA CON LOS VALORES INICIALES
```

```
mm = 3x3  
3.0000 0.0010 -0.0010  
0.2000 -32.4000 0.9950  
-0.0990 -0.0990 20.0000
```

```
jacobian_invers=eval((MatrizJacobiana)^-1) % HALLAMOS LA INVERSA DEL JACOBIANO
```

```
jacobian_invers = 3x3  
0.3333 0.0000 0.0000  
0.0021 -0.0309 0.0015  
0.0017 -0.0002 0.0500
```

```
kk=(mm^-1)*nn %
```

```
kk = 3x1  
-0.3999  
0.0805  
0.4215
```

```
hh=num_var'-kk % DETERMINAMOS LOS NUEVOS VALORES DE X-Y-Z
```

```
hh = 3x1  
0.4999  
0.0195  
-0.5215
```

$$A_{i+1} = (A_i)^{-1} + \frac{[\Delta x_i - A_{xi}^{-1} \cdot \Delta f_{(xi)}] \Delta x_i^T \cdot A_{xi}^{-1}}{\Delta x_i^T \cdot A_{xi}^{-1} \cdot \Delta f_{(xi)}}$$

$$\Delta f_{(xi)} = \begin{pmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \\ z_{i+1} - z_i \end{pmatrix} \quad A_{xi} = \text{Jacobiano} \quad \Delta x_i = \begin{pmatrix} x_{i+1} \\ y_{i+1} \\ z_{i+1} \end{pmatrix}$$

```

x=hh(1); %ASIGNAMOS EL NUEVO VALOR A X
y=hh(2); %ASIGNAMOS EL NUEVO VALOR A Y
z=hh(3); %ASIGNAMOS EL NUEVO VALOR A Z
k=1;
a=[1 2 3];
b=[1 2 3];
printt=[a;b];
while(k<=str2double(answer{5}))
    sistema_evaluado=eval(matrixx); %EVALUAMOS EL SISTEMA CON X-Y-Z
    delta_fxs=(sistema_evaluado-nn); % DIFERENCIA ENTRE LOS SISTEMAS EVALUADOS
    nn=sistema_evaluado; % ASIGNAMOS NUEVO SISTEMA EVALUADO
    jacobianinvert_x_delt=jacobian_invers*delta_fxs ;
    first_part=(hh-num_var')-jacobianinvert_x_delt;
    num_var=hh';
    final_delts_trans_x_jacoinv=hh'*jacobian_invers ;
    numerator=first_part*final_delts_trans_x_jacoinv;
    denominator=final_delts_trans_x_jacoinv*delta_fxs ;
    division=numerator./denominator;
    broydenn=jacobian_invers+division;
    kk=broydenn*sistema_evaluado;
    hh=hh-kk;
    jacobian_invers=broydenn;
    x=hh(1);
    y=hh(2);
    z=hh(3);
    printt(k,:)=(hh');
    Valores_x=printt(:,1);
    Valores_y=printt(:,2);
    Valores_z=printt(:,3);
    export_dates=[Valores_x,Valores_y,Valores_z];
    k=k+1;
end
dates_table=table(Valores_x,Valores_y,Valores_z)

```

dates\_table = 10x3 table

	Valores_x	Valores_y	Valores_z
1	0.5000	0.0088	-0.5232
2	0.5000	0.0037	-0.5234
3	0.5000	0.0005	-0.5236
4	0.5000	0.0001	-0.5236
5	0.5000	-0	-0.5236
6	0.5000	0	-0.5236
7	0.5000	0	-0.5236
8	0.5000	-0	-0.5236
9	0.5000	-0	-0.5236
10	0.5000	-0	-0.5236

```
xlswrite('RESULTADOS.xlsx',export_dates);
```