TAREA problema aplicado autovalores

Carolina Camargo

April 30, 2021

Descripción del problema:

La carga de compresión de colapso por pandeo de una columna se puede determinar resolviendo el siguiente problema de autovalores para los desplazamientos transversales $(u_1, u_2, ..., u_5)$

$$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = -\frac{L^2 p^2}{36} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} \quad , \quad p^2 = \frac{P}{EI}$$

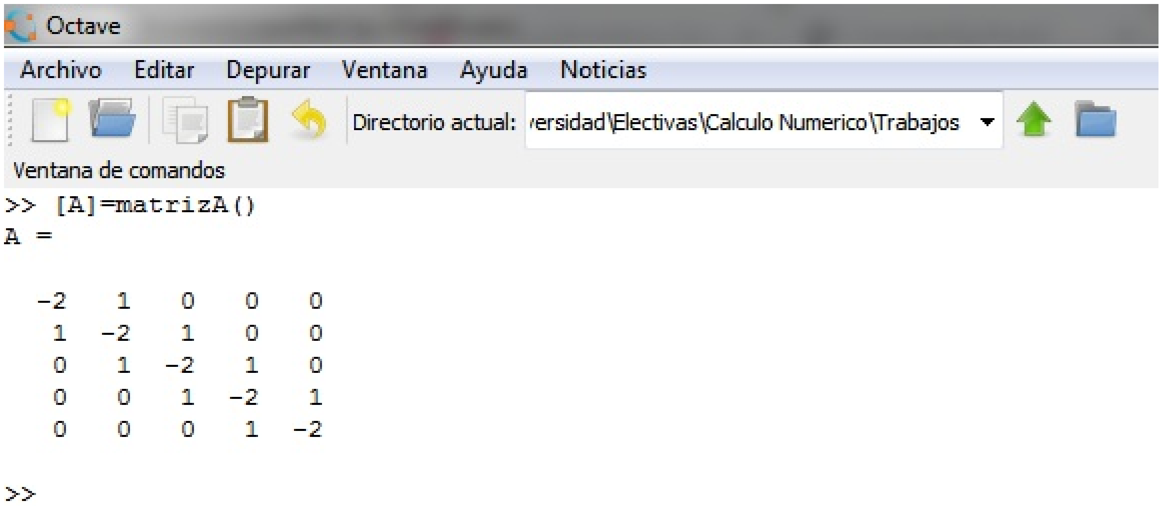
Desarrolle un algoritmo que aproxime el valor de la mínima carga P que logra el colapso, considerando E=10E9[Pa], I=1.25E-5[m4], L=3[m].

1. Creación de la matriz A

- ***Código de Octave***

```
function [A]=matrizA()
    %% A es una matriz NxN
    %% Donde la diagonal esta formada por -2
    %% y 1 en las columnas (i,i+1) y en las fijas (i+1,i).
    A=-2*eye(5);
    for i=1:4
    A(i,1+i)=A(1+i,i)=1;
    endfor
    endfunction
```

- ***Captura***



2. Cálculo de la norma de un vector

- ***Código de Octave***

```
function [norma1]=norm1(x)
    %% calculo de la norma 1 de un vector.
    sum1=abs(x(1));
    for i=2:length(x)
    sum1=sum1+abs(x(i));
    endfor
    norma1=sum1;
    endfunction
```

3. Solucion del SEL

- ***Código de Octave***

```
function [X]=tSSR(A,B)
    %% A matriz N x N no singular
    %% B es una matriz N x 1
    %% X es matriz N x 1 con la solución de AX=B.
    %% C es una matriz de almacenaje temporal.
    %% Entrada
    [N N]=size(A); X=zeros(N,1); C=zeros(1,N+1);
    %% Crea 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
```

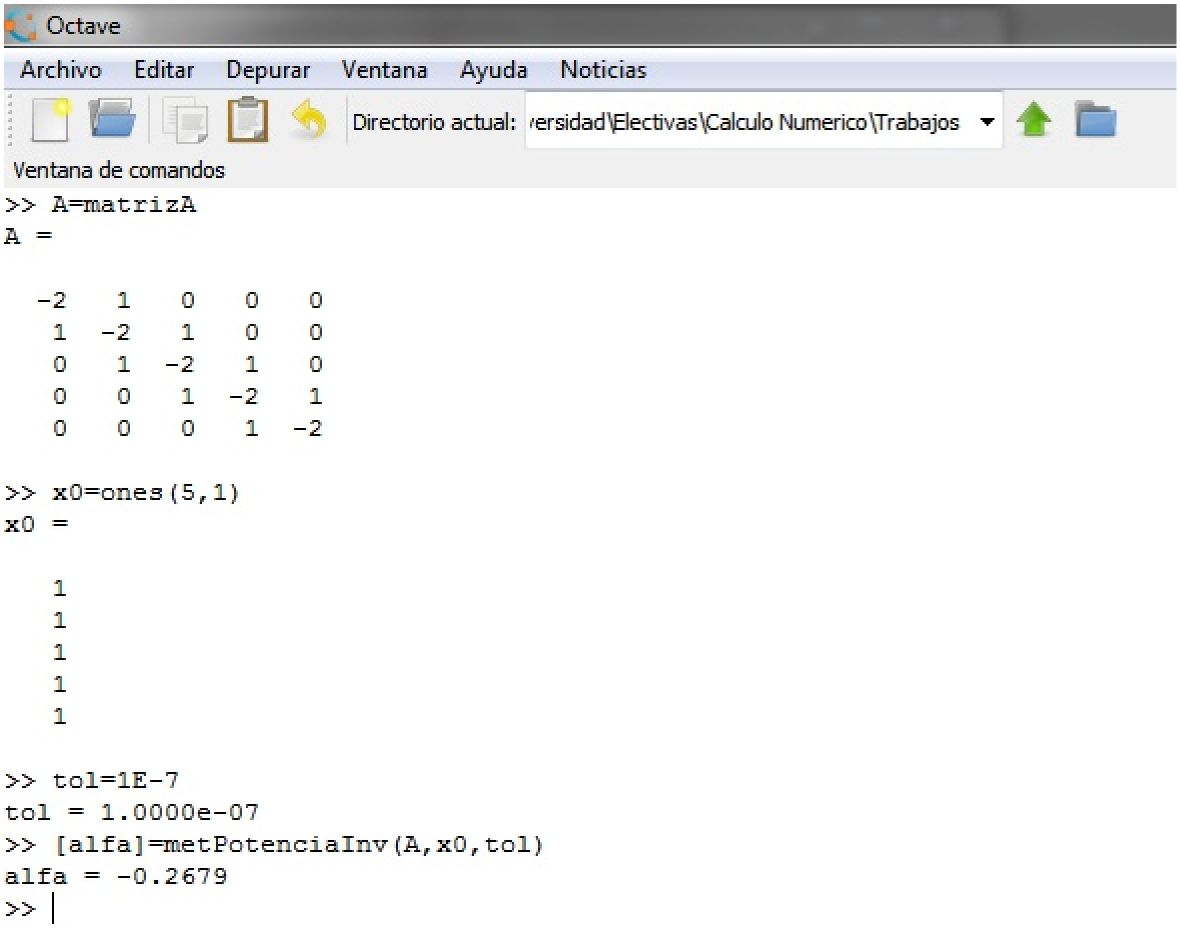
```
C=Aug(p,:); Aug(p,:)=Aug(j+p-1,:); Aug(j+p-1,:)=C;
if Aug(p,p)==0
'A es singular. No hay solución única'
break
endif
%%etapa de eliminación para columna p
for k=p+1:N
m=Aug(k,p)/Aug(p,p);
Aug(k,p:N+1)=Aug(k,p:N+1)-m*Aug(p,p:N+1);
endfor
endfor
%%Salida
%%sustituación hacia atrás en [U|Y]
X=sustRegre(Aug(1:N,1:N),Aug(1:N,N+1));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

4. Método de la potencia inversa

- ***Código de Octave***

```
function [alfa]=metPotenciaInv(A,x0,tol)
%% Funcion que encuentra los autovalores de un sistema
%% mediante el metodo de la potencia inversa utilizando
%% la funcion de triangulacion superior seguida de
%%sustitucion regresiva ocupada en SEL.
%% Entrada
alfa=0;
erro=1;
x0n=x0/norm1(x0,1);
x1=tSSR(A,x0n);
alfa=x1./x0n;
erro=max(abs(alfa))-min(abs(alfa));
%% Encontramos el valor adecuado del autovalor asociado
%% asociado al sistema.
while erro>tol
x0=x1;
x0n=x0/norm1(x0,1);
x1=tSSR(A,x0n);
alfa=x1./x0n;
erro=max(abs(alfa))-min(abs(alfa));
endwhile
%% Salida
alfa=1/(alfa(1));
endfunction
```

- ***Captura***




```
>> A=matrizA
A =

  -2   1   0   0   0
   1  -2   1   0   0
   0   1  -2   1   0
   0   0   1  -2   1
   0   0   0   1  -2

>> x0=ones(5,1)
x0 =

   1
   1
   1
   1
   1

>> tol=1E-7
tol = 1.0000e-07
>> [alfa]=metPotenciaInv(A,x0,tol)
alfa = -0.2679
>> |
```

5. *Solucion del problema para la carga mínima*

- ***Código de Octave***

```
function [P]=minCargaP()
    %% Esta funcion encuentra el valor minimo de carga P que logra el colapso.
    %% Se resuelve utilizando el metodo de la potencia inversa.
    %% Donde se considera un vector inicial y una tolerancia como datos.
    %%Entrada
    A=matrizA;
    x0=ones(5,1);
    tol=1E-10;
    [alfa]=metPotenciaInv(A,x0,tol);
    lanmda=alfa;
    %% Salida
    disp("El valor minimo de carga es:")
```

P=lanmda/(-2E-6);
endfunction

- ***Captura***