

Análisis Matricial de Estructuras

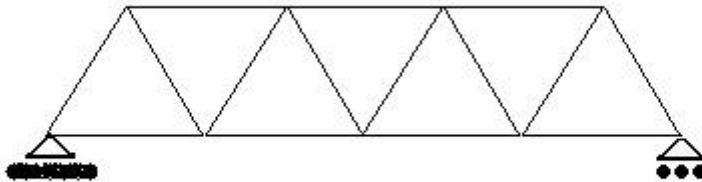
Carlos Armando De Castro Payares

Diseño de Marcos: Problema propuesto por René Meziat, Universidad de los Andes.

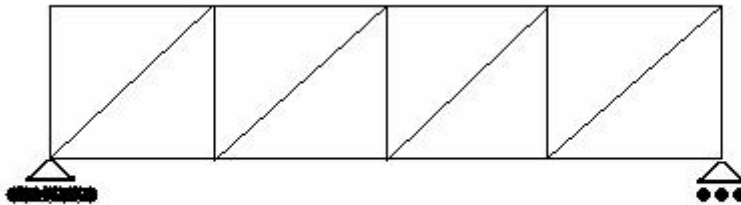
Vamos a definir un marco como una estructura plana compuesta por barras y nodos que carecen de masa y de peso en comparación a las cargas aplicadas que debe soportar la estructura. La estructura puede tener apoyos fijos y apoyos tipo “patín” que solo las sostienen en una dirección. El problema general es encontrar las fuerzas en cada barra como función de las cargas externas que sostiene la estructura las cuales sólo pueden actuar sobre los nodos.

Como ejemplo consideraremos dos tipos de marcos:

Tipo 1



Tipo 2



Note que los tipos de estructuras dados son modulares, es decir que su forma y diseño se repite regularmente, para el primero en forma de triángulos y para el segundo en forma de cuadrados cruzados por la diagonal. Desarrollar un programa que funcione para cualquiera de los dos tipos de estructuras y realice las siguientes operaciones.

- 1) Reciba del usuario el tipo de estructura elegida: Tipo 1 o Tipo 2.
- 2) Reciba del usuario el tamaño de la estructura, ya sea de Tipo 1 o de Tipo 2.
- 3) Reciba del usuario las cargas aplicadas en cada uno de los nodos de la estructura seleccionada en cuanto a tipo y tamaño.
- 4) Calcule las fuerzas transmitidas en cada uno de los elementos.
- 5) Mostrar al usuario las fuerzas en cada elemento.
- 6) Mostrar al usuario las reacciones en los apoyos.
- 7) Advertir de las máximas tensiones y máximas compresiones calculadas.
- 8) Pedir al usuario el Esfuerzo de Fluencia y el área de la sección transversal de un material y decidir si el material especificado es conveniente o no para las cargas dadas.
- 9) Pedir al usuario el módulo de Young del material de las barras para determinar la elongación en cada una de ellas.

Solución: El algoritmo elaborado en Matlab es el siguiente:

```

function [Fuerzas, Reacciones, Esfuerzos, Deformaciones,
Max_Compresion, Max_Tension]=cerchas

%Elaborado por Carlos Armando De Castro Payares.
%Curso de Análisis Numérico.
%Universidad de los Andes.

seleccion=input('Tipo de cercha (1 o 2):');

switch seleccion;

    case 1;

        %Número de triángulos:
        %-----

        n=input('Número de triángulos: ');

        %Número de nodos:
        %-----

        n=n+2;

        %Matriz topológica:
        %-----

        %Nodos generales:

        for i=1:n-2;
            topologica(i,i+1)=1;
            topologica(i,i+2)=1;
        end

        topologica(n-1,n)=1;

        for i=1:n;
            for j=1:n;
                topologica(j,i)=topologica(i,j);
            end
        end

        %Nodos con apoyo:

        topologica(1,n+1)=3;
        topologica(n,n+1)=1;

        for i=2:n-1;
            topologica(i,n+1)=0;
        end

        %Matriz de ángulos:
        %-----

        for i=1:2:n-2;

```

```

        angulos(i,i+1)=pi/3;
        angulos(i,i+2)=0;
end

for i=2:2:n-2;
    angulos(i,i+1)=-pi/3;
    angulos(i,i+2)=0;
end

angulos(n-1,n)=-pi/3;

for i=1:n;
    for j=i+1:n;
        angulos(j,i)=angulos(i,j)+pi;
    end
end

%*****
%*****
%Llenado de la matriz de coeficientes A:
%*****
%*****

%Barras de la estructura. Diagonal superior:
%-----

k=1;
m=1;
for i=1:n;
    for j=i+1:n;
        if topologica(i,j)==1
            A(2*k-1,m)=cos(angulos(i,j));
            A(2*k,m)=sin(angulos(i,j));
            m=m+1;
        end
    end
    k=k+1;
end

%Diagonal inferior:
%-----

k=1;
m=1;
for i=1:n;
    p=0;
    for j=i+1:n;
        if topologica(i,j)==1
            p=p+1;
            A(2*k-1+2*p,m)=-A(2*k-1,m);
            A(2*k+2*p,m)=-A(2*k,m);
            m=m+1;
        end
    end
    k=k+1;
end
end

```

```

%Nodos con apoyo:
%-----

L=length(A(1,:));
k=1;
for i=1:n;
    if topologica(i,n+1)~=0
        switch topologica(i,n+1);
            case 1;
                A(2*i,L+k)=1;
                k=k+1;
            case 2;
                A(2*i-1,L+k)=1;
                k=k+1;
            case 3;
                A(2*i,L+k+1)=1;
                A(2*i-1,L+k)=1;
                k=k+2;
            end
        end
    end
end

        case 2;

%Número de cuadrados:
%-----

n=input('Número de cuadrados: ');

%Número de nodos:
%-----

n=4+2*(n-1);

%Matriz topológica:
%-----

%Nodos generales:

for i=1:n-2;
    topologica(i,i+1)=1;
    topologica(i,i+2)=1;
end

topologica(n-1,n)=1;

for i=1:n;
    for j=1:n;
        topologica(j,i)=topologica(i,j);
    end
end

%Nodos con apoyo:

topologica(1,n+1)=0;
topologica(2,n+1)=3;
topologica(n,n+1)=1;

```

```

for i=3:n-1;
    topologica(i,n+1)=0;
end

%Matriz de ángulos:
%-----

for i=1:2:n-3;
    angulos(i,i+1)=-pi/2;
    angulos(i,i+2)=0;
end

for i=2:2:n-2;
    angulos(i,i+1)=pi/4;
    angulos(i,i+2)=0;
end

angulos(n-1,n)=-pi/2;

for i=1:n;
    for j=i+1:n;
        angulos(j,i)=angulos(i,j)+pi;
    end
end

%*****
%*****
%Llenado de la matriz de coeficientes A:
%*****
%*****

%Barras de la estructura. Diagonal superior:
%-----

k=1;
m=1;
for i=1:n;
    for j=i+1:n;
        if topologica(i,j)==1
            A(2*k-1,m)=cos(angulos(i,j));
            A(2*k,m)=sin(angulos(i,j));
            m=m+1;
        end
    end
    k=k+1;
end

%Diagonal inferior:
%-----

k=1;
m=1;
for i=1:n;
    p=0;
    for j=i+1:n;
        if topologica(i,j)==1

```

```

        p=p+1;
        A(2*k-1+2*p,m)=-A(2*k-1,m);
        A(2*k+2*p,m)=-A(2*k,m);
        m=m+1;
    end
end
k=k+1;
end

%Nodos con apoyo:
%-----

L=length(A(1,:));
k=1;
for i=1:n;
    if topologica(i,n+1)~=0
        switch topologica(i,n+1);
            case 1;
                A(2*i,L+k)=1;
                k=k+1;
            case 2;
                A(2*i-1,L+k)=1;
                k=k+1;
            case 3;
                A(2*i,L+k+1)=1;
                A(2*i-1,L+k)=1;
                k=k+2;
            end
        end
    end
    otherwise

        'Selección no válida.';

end

%Lectura de las cargas:
%-----

['Cargas verticales positivas hacia abajo, cargas horizontales
positivas hacia la izquierda.'];
for i=1:n;
    if topologica(i,n+1)==0;
        Fx(i)=input(['Carga horizontal en el nodo ' num2str(i) ': ']);
        Fy(i)=input(['Carga vertical en el nodo ' num2str(i) ': ']);
    else
        Fx(i)=0;
        Fy(i)=0;
    end
end

for i=1:n;
    b(2*i-1,1)=Fx(i);
end

for i=1:n;
    b(2*i,1)=Fy(i);
end

```

end

```
%*****  
*****  
%Solución del sistema:  
%*****  
*****
```

X=A⁻¹*b;

```
for i=1:length(X)-3;  
    Fuerzas(i)=X(i);  
end  
for i=1:3;  
    Reacciones(i)=X(length(X)-(3-i));  
end
```

```
%*****  
*****  
%Lectura de las propiedades del material:  
%*****  
*****
```

```
E=input('Módulo de Young del material E= ');  
Ar=input('Área transversal de las barras Ar= ');  
Sy=input('Esfuerzo de fluencia del material Sy= ');
```

```
Esfuerzos=1/Ar*Fuerzas;  
Deformaciones=Esfuerzos/E;
```

```
%*****  
*****  
%Valores máximos de las fuerzas:  
%*****  
*****
```

```
Max_Compresion=Fuerzas(1);  
Max_Tension=Fuerzas(1);
```

```
for i=2:length(Fuerzas);  
    if abs(Fuerzas(i))>abs(Max_Compresion) && Fuerzas(i)<0  
        Max_Compresion=Fuerzas(i);  
    end  
    if abs(Fuerzas(i))>abs(Max_Tension) && Fuerzas(i)>=0  
        Max_Tension=Fuerzas(i);  
    end  
end
```

```
w=0;  
i=1;  
while i<=n && w==0;  
    if abs(Esfuerzos(i))<Sy  
        i=i+1;  
    else  
        w=1;  
    end  
end
```

```
if w==0
    ['El material es adecuado.']
else
    ['El material es inadecuado. El esfuerzo supera el límite
    elástico.']
end
```