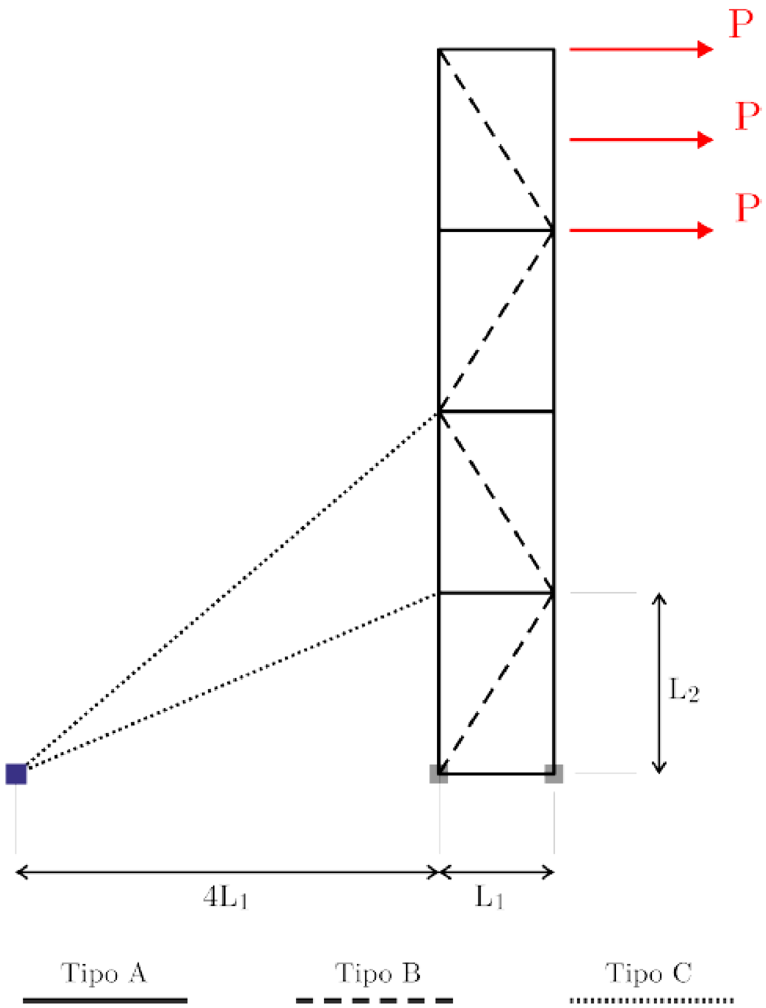
SIMULACIÓN DE UNA ESTRUCTURA DE BARRAS

Trabajo de Simulación de Materiales y Estructuras

Universidad Loyola Andalucía
Facultad de Ingeniería
Grado Ingeniería Informática y Tecnologías Virtuales



Autora

Ferrer Haba, Laura

Curso Académico **2021-22**

ÍNDICE DE CONTENIDO

ÍNDICE DE FIGURAS

1. INTRODUCCION Y OBEJTIVOS

El modelo de la Figura 1 representa una estructura de barras sometidas a un sistema de fuerzas puntuales.

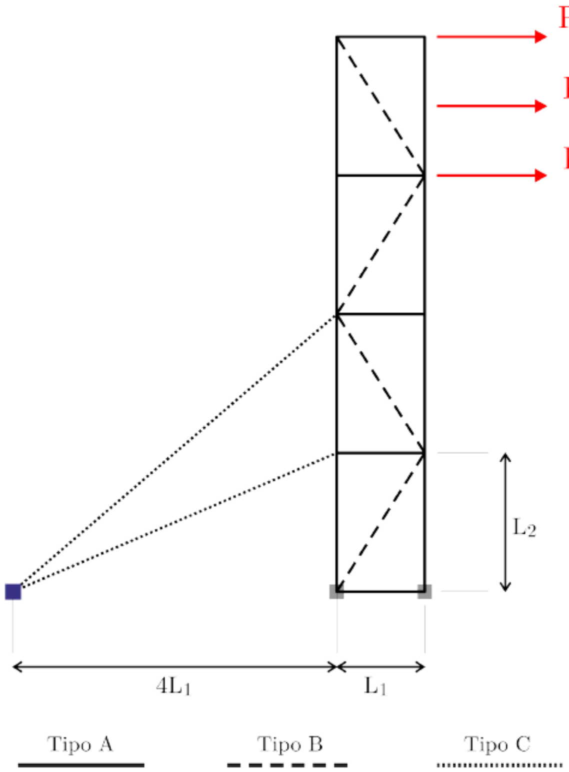


Figura 1:Figura 2D

La estructura está compuesta por barras de acero (rigidez 210-GPa, límite elástico 235-MPa), cuyo peso propio puede despreciarse. Tome L_1 = 2-m y L_2= 3-m. Cada carga puntual tiene un valor P = 500 N. Los nodos grises representan empotramientos perfectos. Considere además que en el nodo resaltado azul se impone un desplazamiento vertical y hacia arriba de 1-cm.

Las barras están distribuidas en tres grupos:

- **Tipo 1**: Perfil IPE80.
- **Tipo 2**: Perfil Cuadrado hueco 60.3.
- **Tipo 3**: Tensor de cable de acero redondo de 20 mm.

Se pide:

1. Calcular el nodo que sufre mayor desplazamiento y su valor.
2. Representar la estructura deformada sobre la indeformada.

El objetivo de esta práctica poder ser capaz de crear un solver numérico (**SolverBeam2D**) que permita la resolución paramétrica de estructuras de elementos (**barra biarticuladas**). Dicha función recibirá un único argumento, que será una estructura, con toda la información necesaria y devolverá una tabla con la solución, representando gráficamente la estructura deformada e indeformada. Todas las funciones que se consideren necesarias deberán ser indexadas al final del propio código o bien incluidas en archivos independientes.

2. EXPLICACIÓN DE LA IMPLEMENTACIÓN

En este apartado explicaré como he resuelto la práctica mediante la utilización de un script, en el cual se engloban dos funciones (**SolverBeam2D** y **PlotSol**)

2.1. DATOS

Los datos proporcionados para esta práctica son, los siguientes:

```
L1 = 2;         %m
L2 = 3;         %m
P = 500;        %N
A = 7.64e-4;    %Área de las barras de tipo A
B = 6.53e-4;    %Área de las barras de tipo B
C = 3.14e-4;    %Área de las barras de tipo C
IzA = 8.01e-7; %Inercia de flexión de la barra tipo A
IzB = 3.44e-7; %Inercia de flexión de la barra tipo B
IzC = 7.85e-9; %Inercia de flexión de la barra tipo C
E = 2.10e11;    %Pa
```

Figura 2: Datos Proporcionados

2.2. CREACIÓN DE MATRICES

Para la correcta implementación deberemos crear una Matriz de elementos y de nodos. La matriz de elementos contendrá los elementos de la figura 3 (que en mi caso son 20), el área de cada tipo de elemento, el módulo de Young y su correspondiente inercia de flexión. He enumerado los **elementos** (Color azul) y los **nodos** (Color rojo) de la siguiente forma:

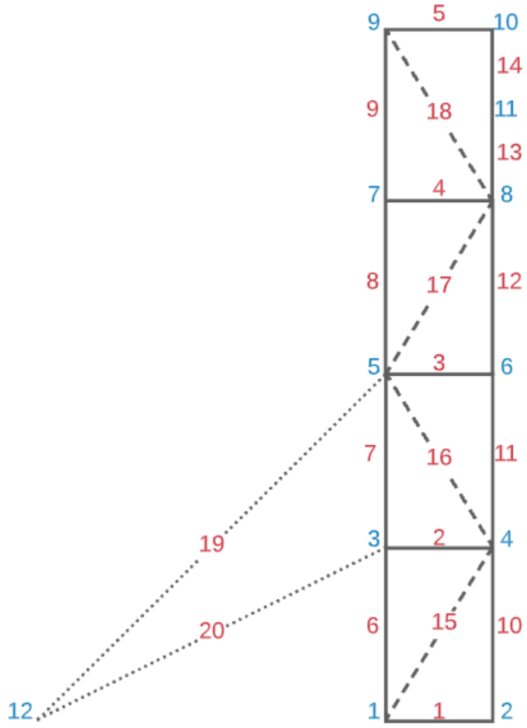


Figura 3: Enumeración de los Elementos

Siendo los nodos **1** y **2** los que no se mueven en ninguno de sus ejes (ni x ni y). En cambio los nodos **8**, **10** y **11** en su eje x si se mueve. Una vez tenido en cuenta esto, pasamos a crear las matrices de elementos y nodos.

La matriz de elementos: será una **20x5**, debido a que **hay 20 elementos**. Es una matriz 20x5 debido a que el primer elemento es el nodo, el segundo y el tercero la conexión entre los nodos, el cuarto el módulo de Young, el quinto el área y el sexto es el ángulo de torsión en el eje x. Para ello creamos cuatro bucles for. El **primer for** irá del elemento **1** hasta el **5**. Los elementos se unen entre **a** y **a+1**, siendo .**a = 1** y sumándole **2** a **a** al terminar cada iteración. El **segundo for** irá del elemento **6** hasta el **9**. Los elementos se unen entre **b** y **b+2**, siendo .**b = 1** y sumándole **2** a **b** al terminar cada iteración. El **tercer for** irá del elemento **10** hasta el **12**. Los elementos se unen entre **c** y **c+2**, siendo .**c = 2** y sumándole **2** a **c** al terminar cada iteración. El **cuarto for** irá del elemento **15** hasta el **18**. Este for constará de un **if** el cual sí el elemento es un número par se unirán entre **d+1** y **d**, siendo **d = 4** y sumándole **4** a **d** al terminar cada iteración. Mientras que si es un número impar se unirán entre **e** y **e+3**, siendo **e = 1** y sumándole **4** a **e** al terminar cada iteración. Los elementos **13**, **14**, **19** y **20** han sido introducidos a mano por no tener un patrón.

La matriz de nodos: será una **12x3**, debido a que **hay 12 nodos**. En ella guardaremos la coordenadas que existen entre los nodos, siendo estas los elementos dos y tres respectivamente de la matriz. Para ello elegiremos el centro de coordenadas en el nodo **1**. Hemos de tener en cuenta **L1** y **L2**.

2.3. CONDICIONES DE DESPLAZAMIENTOS

Las condiciones de desplazamiento son los desplazamientos que hay que tener en cuenta. Estos serán las componentes **x**, **y**, y **theta** de los nodos. Estos indican que nodos no se mueven (sujeción), estos nodos son el **1** y **2**, y que nodos si se mueven, por lo que el resto de los nodos se mueven salvo el nodo **12** que solo se mueve en la componente vertical, es decir, en el eje **y**.

2.4. CONDICIONES DE FUERZA

Las condiciones de fuerza son las fuerzas que hay que tener en cuenta. Estas serán las fuerzas que se les aplique a un nodo, los nodos a los que se les aplica fuerza son **8**, **10** y **11**, siendo esta fuerza de **500N**.

2.5. FUNCIÓN SolverBeam2D

2.5.1. DATOS

Lo primero que hay que hacer es sacar el tamaño del problema. Para ello sacamos el tamaño de Mnod (Matriz de nodos), estos serán los grados de libertad por **3** dado que cada nodo tiene **tres componentes**(**x**, **y**, **theta**). Con los grados de libertad construimos la matriz K, esta será la matriz de rigidez local. Y por último sacamos el tamaño de la matriz de elementos. Este tamaño servirá para iterar en cada elemento.

Los nodos i y j representan las columnas 2 y 3 de la matriz de elementos, siendo estos los nodos que une.

2.5.2. PROPIEDADES DE LA VIGA

Sacamos las propiedades de la viga una vez dentro del for. En mi caso la **Matriz de Elementos** tiene como parámetros el área (**A**, **B**, **C**) de los tipos de barras que hay, el módulo de Young (**E**) y la inercia de flexión de la barra de cada tipo (**IzA**, **IzB**, **IzC**). La L podemos calcularla con la fórmula: **sqrt((x2 - x1)^2 + (y2 - y1)^2)** puesto que conocemos las componentes **x** e **y** de los nodos. Asimismo podemos calcular **theta** debido a que conocemos el cateto opuesto y su hipotenusa (será la componente y de los nodos) por lo que podemos aplicar el arcoseno para hallar el ángulo.

2.5.3. MATRIZ DE RIGIDEZ LOCAL

La matriz de rigidez local se calcula con la fórmula **k = T^Tk'T,** siendo:

$$\mathbf{T} = \begin{bmatrix} l & m & 0 & & & \\ -m & l & 0 & & 0 & \\ 0 & 0 & -1 & & & \\ & & & l & m & 0 \\ & 0 & & -m & l & 0 \\ & & & 0 & 0 & -1 \end{bmatrix} \begin{cases} m = \sin\sigma \\ l = \cos\sigma \end{cases} \quad \mathbf{k}' = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & \frac{-EA}{L} & 0 & 0 \\ 0 & \frac{12EIz}{L^3} & \frac{6EI_z}{L^2} & 0 & \frac{-12EIz}{L^3} & \frac{6EI_z}{L^2} \\ 0 & \frac{6EIz}{L^2} & \frac{4EI_z}{L} & 0 & \frac{-6EIz}{L^2} & \frac{4EI_z}{L} \\ \frac{-EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & \frac{-12EIz}{L^3} & \frac{-6EI_z}{L^2} & 0 & \frac{12EIz}{L^3} & \frac{-6EI_z}{L^2} \\ 0 & \frac{6EIz}{L^2} & \frac{4EI_z}{L} & 0 & \frac{-6EIz}{L^2} & \frac{4EI_z}{L} \end{bmatrix} \text{y}$$

k' = EA/L. . Por otro lado, creamos **6** nuevos índices para establecer **i** y **j** puesto que al entrar en el bucle se superponen. Sabemos que se cumple la condición de que el **nodo 1** es **1**, **2** y **3** por lo que la componente **x** de **i** será **(i*3)-2**, su componente **y** será **(i*3)-1** y la componente **theta** será **(i*3)**, pasará lo mismo para **j**.

2.5.4. REDUCIMOS EL SISTEMA

Reducimos el sistema para quedarnos con la parte que nos interesa de él, es decir, quitamos los desplazamientos que sean cero. En este caso como tenemos incógnitas en la **U** y en la **F** deberemos de usar la fórmula: **Uf = Kff\(Ff-Kfs*Us).**

2.5.5. RESUELVO EL SISTEMA REDUCIDO

Resolvemos el sistema reducido, eliminando los elementos correspondientes a desplazamientos nulos y obtengo la matriz de contorno completa.

2.6. FUNCIÓN PlotSol

2.6.1. INDEFORNADO

Sacamos los elementos de las columnas 2 y 3 de la matriz de elementos. Después obtenemos la componente x e y de cada elemento (nodo). A cada tipo de barra se le atribuye un color, en mi caso la barra1 será de color azul, mientras que la barra 2 será verde y la barra 3 será roja.

2.6.2. DEFORMADO

Sacamos los elementos de las columnas 2 y 3 de la matriz de elementos. Después obtenemos la componente x e y de cada elemento (nodo). A cada tipo de barra se le atribuye un color, en mi caso la barra1 será de color azul, mientras que la barra 2 será verde y la barra 3 será roja. Por otro lado, crearemos **4** nuevos índices que serán las posiciones de **i** y **j** en la matriz de rigidez local. Usamos la función **linspace** para sacar puntos y la función **spline** para sacar la pendiente de dichos puntos. Los juntaremos en una matriz y la aplicaremos el giro correspondiente. Por último, le sumaremos la posición en la cual deben de estar.

2.6.3. NUMERACIÓN DE LOS PUNTOS

Para enumerar solo habrá que convertir el nodo en número ya que lo toma como un string y usar la función text.

3. NODO QUE SUFRE MAYOR DESPLAZAMIENTO Y SU VALOR

Para calcular el desplazamiento máximo calcularemos el módulo de cada nodo para obtener cuanto se mueven. Para ello crearemos una nueva función a la cual le pasaremos la U calculada previamente y haremos un bucle que recorra todos los elementos del vector U de dos en dos y calcule el módulo. Una vez calculado el módulo sacaremos el máximo usando la función max de Matlab.

```
function [Umax, U_nu, nod]  = desMax(U)
    %Leo el tamaño del problema
    [Nel, ~] = size(U);
    U_nu = zeros(Nel/3, 1);
    a = 1;
    Umax = 0;

    for spring = 1:3:Nel
        %Identifico los nodso i-j del vector U
        i = U(spring, 1);
        j = U(spring+1, 1);
        k = U(spring+2, 1);
        U_nu(a) =  sqrt(i^2+j^2+k^2);

        if U_nu(a) > Umax
            %Sacamos el desplazamiento maximo absoluto
            Umax = U_nu(a);
            nod = a;
        end
        a = a + 1;
    end
end
```

Figura 4: Desplazamiento Máximo

4. REPRESENTAR LA ESTRUCTURA DEFORMADA SOBRE LA INDEFORMADA

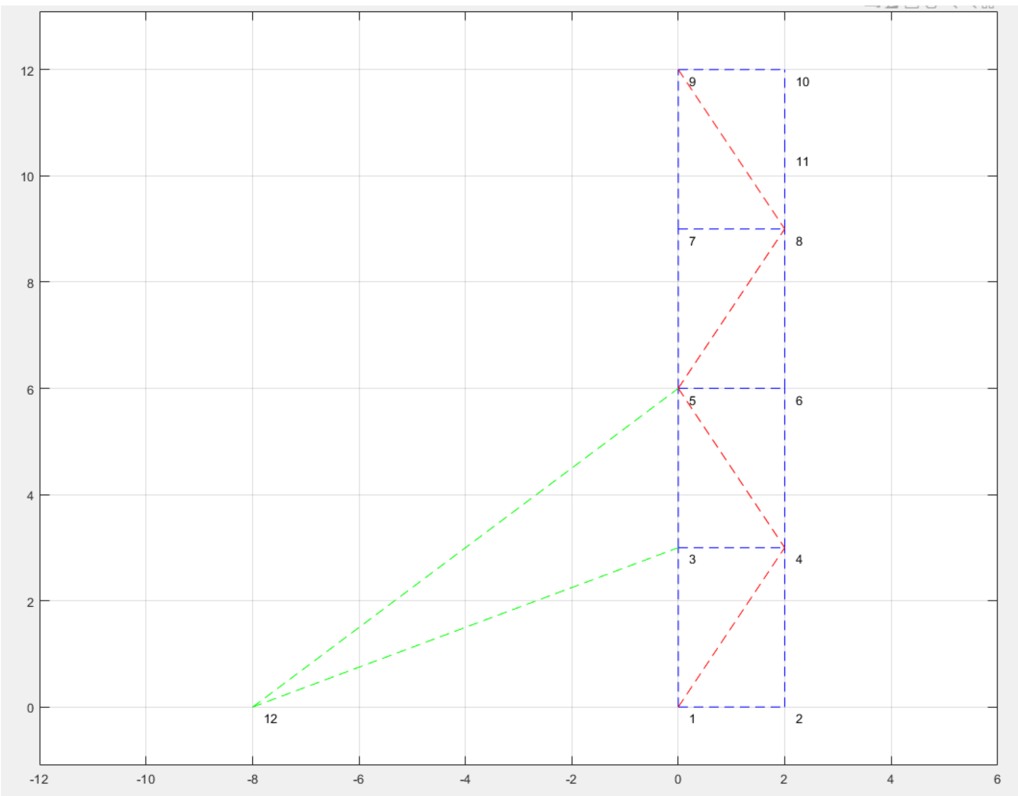


Figura 5:Ploteo Deformado

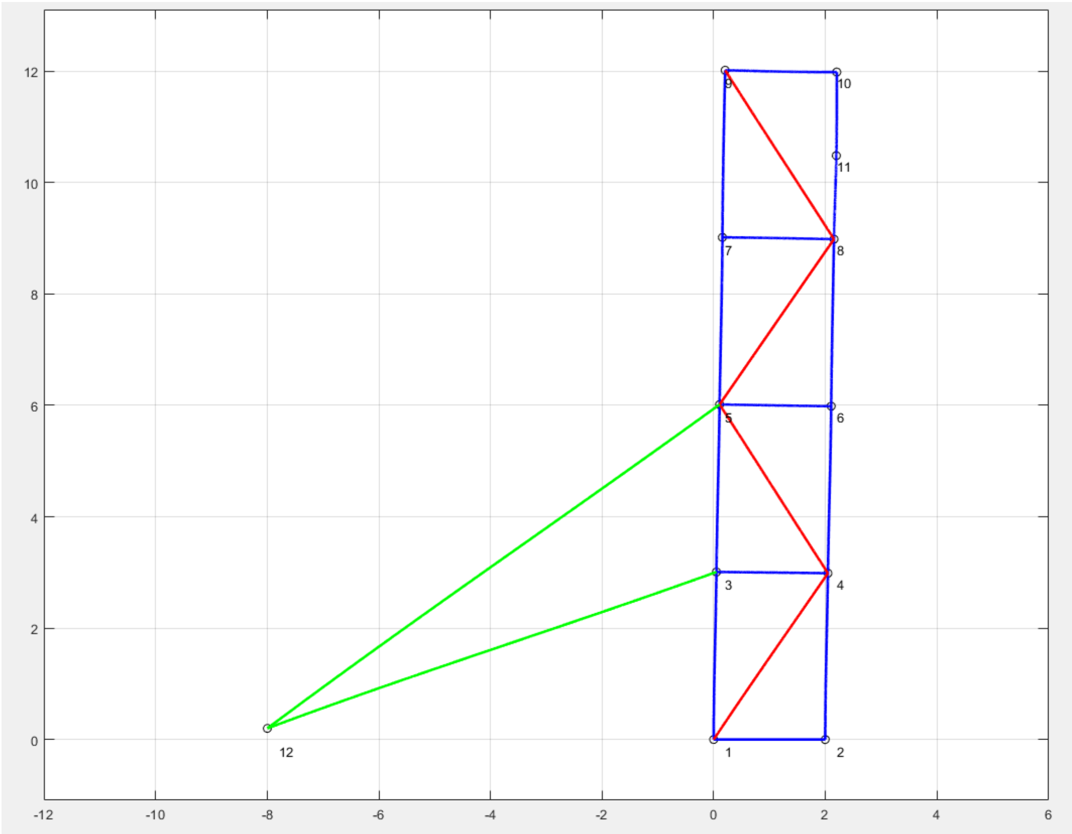


Figura 6: Ploteo Indeformado

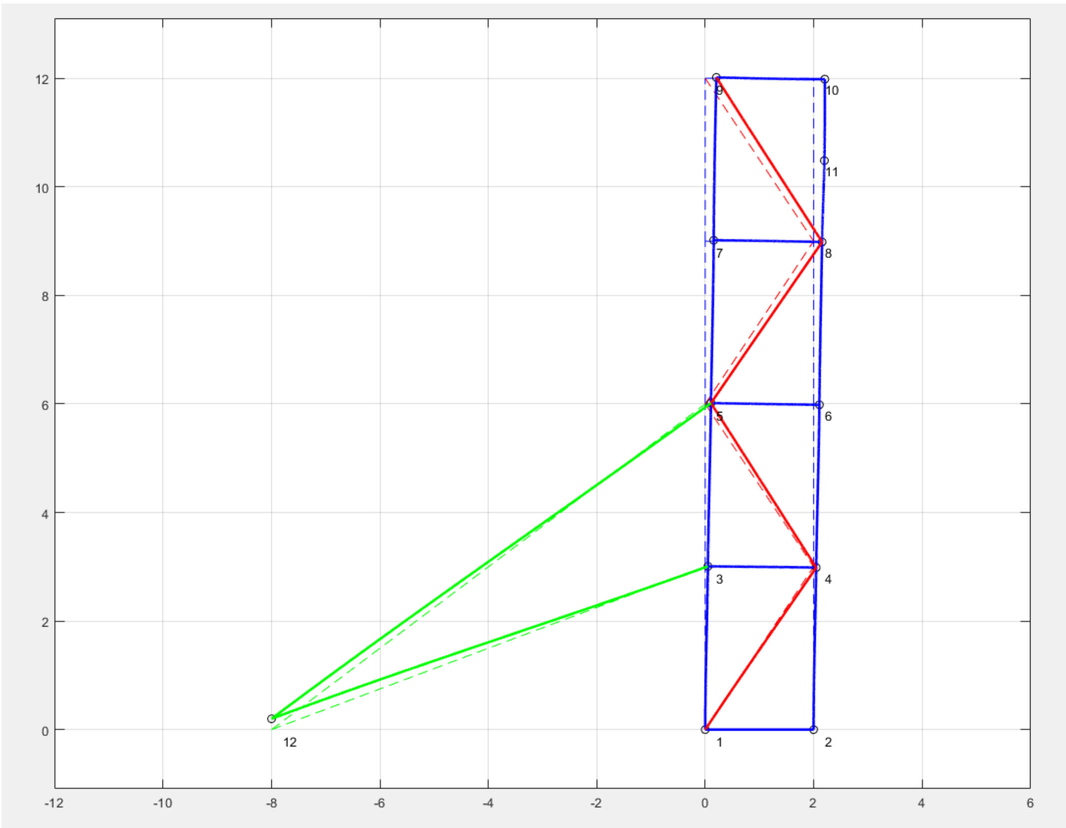


Figura 7: Ploteo Total

5. SOLUCIONES

```
----------------------Resolución Práctica 3----------------------
Los desplazamientos por nodos son:
         0
         0
    0.0026
    0.0026
    0.0054
    0.0054
    0.0079
    0.0079
    0.0104
    0.0103
    0.0100
    0.0100

Por lo que el desplazamiento máximo es:1.040335e-02 en el nodo:9
```

Figura 8: Resultados