Matlab

Agustin Teliz

September 2018

1 Introducción

En este trabajo se pretende mostrar cómo utilizar el programa "Torres" desarrollado en el software Matlab con cierto enfoque a torres de alta tensión. Dicho programa tiene como objetivo el estudio de reticulados, estudiando así, esfuerzos máximos sobre la estructura en la que se trabaja, frecuencias naturales, pandeo localizado e inestabilidades globales bajo cargas determinadas por el usuario.

2 Programa

El programa “Torre” está constituido por diferentes partes en las que el usuario puede trabajar. Esta sección trata de cada una de ellas finalizando con un pequeño ejemplo. Comienza con la sección “Definición de la Estructura”, aquí se debe definir distancias, ubicación de los nodos, elementos, propiedades de los materiales, ejes de coordenadas y las cargas sobre la estructura entre otras cosas.

Se comienza con un análisis básico que consiste en lo siguiente: Partir del programa “Análisis Inicial” donde se definen ciertas variables, relacionadas con “definición de la estructura”, que se utilizaran en los diferentes procesos a posteriori. Se prosigue con el programa “Mount” que utiliza diferentes programas para crear la matriz de rigidez y vector de fuerzas externas y a partir de ellos el vector de desplazamientos nodales y vector de esfuerzos. Dentro de los programas básicos, por último se puede utilizar el programa “Plot” para plotear la estructura descargada y el programa “Plot_U” para plotear la estructura deformada.

Por otra parte, cuenta con una serie de programas que calculan si existen barras con pandeo localizado, las frecuencias naturales del sistema y modos de vibración e inestabilidad global para el tipo de cargas aplicadas.

Por último se puede utilizar un sistema de optimización de la torre. Este programa optimiza la torre de forma de disminuir su masa y con esto el costo de la misma.

2.1 Definición de la Estructura

Se presentan los pasos a seguir para definir como está constituida la torre.

Por lo general, en las siguientes matrices a definir, habrá dos tipos, una donde la variable puede ser modificada para el proceso de optimización y otra donde dicha variable es fija en todo caso.

2.1.1 Matrices de secciones

La matriz de secciones variables “MatVarSec” está compuesta de la siguiente manera:

$$MatVarSec = [Num\ Area\ Tmin\ Tmax\ Amin\ Amax\ GrD]$$

Donde “Num” identifica la sección con un número natural de 1 a n, comenzando por 1 y n es el número total de secciones variables. “Area” como dice la palabra, es el área de la barra en cuestión. En este caso puede existir más de un identificador con la misma área, ya que luego hay barras que pueden variar de diferente manera su sección. “Amin” y “Amax” representan el área mínima y máxima respectivamente que puede tomar la barra en el estudio de optimización. Los términos “Tmin” y “Tmax” toman los valores 1 o 0 y le dicen al programa si existe o no cotas superior o inferior respectivamente. Por ejemplo si $Tmin = 1$ cuando se procesa el programa de optimización, el área mínima que puede tomar esta sección será la que se muestre en Amin, de otra forma si $Tmin = 0$ el programa podrá tomar valores menores que el mostrado en Amin. Por último el termino GrD indica. . . .

Dentro de las matrices de sección también se encuentra la matriz “MatVarSecFix” que representa las secciones fijas. Se encuentra compuesta de la siguiente manera:

$$MatVarSecFix = [Num\ Area]$$

Nuevamente, “Num” identifica la sección con un número natural de n+1 a m, comenzando por n+1 y n-m es la cantidad de secciones fijas. Y el término “Area” es el valor de la sección asignada. Por el contrario de las secciones variables, en este caso no tiene sentido tener dos identificadores con la misma área ya que es irrelevante cuando la sección es fija.

2.1.2 Matrices de distancias

Las matrices de distancias también llamadas matrices de posiciones “MatVarPos” y “MatVarPosFix” solamente definen distancias, las cuales más adelante serán utilizadas para colocar los nodos en el espacio.

La matriz “MatVarPos” está compuesta de la siguiente manera:

$$MatVarPos = [\,Num\ Valor\ T_{min}\ T_{max}\ P_{min}\ P_{max}\ ]$$

“Num” identifica al valor con un número natural de 1 a n, comenzando por 1 y n es el número total de valores de distancias variables que habrá que utilizar. Las distancias que se querrán utilizar para colocar los nodos vienen dadas por el término “Valor”. Éste puede estar repetido en más de un identificador si a la hora de la optimización se desea que dos nodos comiencen con una misma distancia pero luego se “muevan” con independencia entre ellos. Los términos “Tmin” y “Tmax” toman los valores 1 o 0 y le dicen al programa si existe o no cota superior o inferior respectivamente. “Pmin” y “Pmax” son las cotas inferiores y superiores que puede tomar la variable en el proceso de optimización.

La matriz “MatVarPosFix” que representa las distancias fijas se encuentra compuesta de la siguiente manera:

$$MatVarPosFix = [\,Num\ Valor]$$

Nuevamente, “Num” identifica los valores con un número natural de $n+1$ a m, comenzando por $n+1$ y $m-n$ es la cantidad de secciones fijas y el término “valor” es la distancia asignada. Por el contrario de las distancias variables, en este caso no tiene sentido tener dos identificadores con el mismo valor ya que es irrelevante cuando la sección es fija.

2.1.3 Matriz de sistemas de coordenadas

La matriz de sistema de coordenadas, “MatSisCoord”, define con cuantos sistemas de coordenadas se trabajará para ubicar los nodos en el espacio. De esta forma se obtiene de forma rápida simetría con respecto a los planos principales y todos los valores en la matriz de distancias pueden ser definidos positivos.

La matriz está compuesta como se esboza a continuación:

$$MatSisCoord = [\,Num\ X\ Y\ Z]$$

“Num” es el indicador de cada sistema de coordenadas. En los lugares donde están X, Y y Z se debe colocar las coordenadas de cada sistema. Esta terna ira multiplicando la distancia asignada al nodo. Por ejemplo, si a cierto nodo le fue asignado un valores de distancias $(X_i\ Y_i\ Z_i) = (10\ 20\ 30)$ con un sistema de coordenadas $(1\ -1\ 1)$, la posición del nodo en el espacio real será $(10\ -20\ 30)$.

2.1.4 Matriz de Nodos

La matriz de nodos define la posición de cada uno de ellos. Para lograr esto lo realiza de la siguiente forma.

$$MatNodos = [\,Num\ C_x\ C_y\ C_z\ Sistema]$$

“Num” es el indicador de cada uno de los nodos. Luego Cx, Cy y Cz toman valores entre 1 y el valor máximo de los identificadores de la matriz de distancias. Por lo tanto, el nodo tomara la distancia del identificador que se coloque allí. Además se debe especificar en qué sistema de coordenadas se encuentra el nodo, ya que si se desea uno de estos valores negativos, basta con asignarle un sistema de coordenadas que se encargue de eso.

2.1.5 Matriz de Materiales

La matriz de materiales cumple la función de información cuantos materiales se utilizaran y las propiedades de los mismos. Se encuentra compuesta de la siguiente manera.

$$MatMateriales = [\,Num\ E\ P\ D\ C\ Tac\ Tat\ tac\ tat]$$

“Num” es el indicador de cada material. “E” representa el módulo de Young, “P” es el coeficiente de Poisson, “D” la densidad del material, “C” ..., “Tac” y “Tat” toman valores de 1 o 0 dependiendo si existe tensión máxima de compresión o tracción respectivamente y por último, “tac” y “tat” representan los valores de tensión máxima a la compresión y tracción respectivamente.

2.1.6 Matriz de Elementos

La matriz de elementos tiene por objetivo nombrar todos los elementos del reticulado, los cuales se encuentran definidos entre dos nodos, además de asignarle la sección y el tipo de material.

$$MatElementos = [Num\ N_1\ N_2\ Seccion\ Material]$$

“Num” es el indicador de cada elemento. “N1” y “N2” representan los nodos por los que estará formado el elemento con los identificadores de la matriz de nodos. Es importante destacar que el programa está diseñado con cierto cuidado en este punto, el usuario debe respetar que el identificador del nodo 1 sea menor al identificador del nodo 2 ($N1 < N2$). Luego, en los lugares de “Seccion” y “Material”, se le debe asignar la sección de la barra y el material con los indicadores de la matriz de secciones y matriz de materiales respectivamente.

2.1.7 Matriz de condiciones de borde

La matriz de condiciones de borde representa las fuerzas que serán aplicadas sobre los nodos y es la encargada de fijar los nodos en el espacio. Para esto se debe completar de la siguiente manera:

$$MatCondBorde = [Num\ T_x\ T_y\ T_z\ V_x\ V_y\ V_z]$$

Donde, nuevamente, “Num” representa el identificador de la condición de borde. Luego siguen los términos T_x, T_y y T_z, los mismos definen los grados de libertad del nodo. Si en uno de ellos aparece un 0, cuando se le asigne a un nodo esta condición de borde, el nodo no podrá desplazarse en este sentido. Por lo general siempre existe un fila de esta matriz formada por $T_x = T_y = T_z = 0$ y de esta forma cuando se le aplica a un nodo esta condición, el nodo queda fijo en el espacio. Por ultimo aparecen los términos V_x, V_y y V_z representa la fuerza y la dirección a aplicar sobre el nodo. Se debe destacar que no tiene sentido que si $T_i = 0$, V_i sea diferente de cero, ya que se le estaría aplicando una fuerza a un nodo en la dirección que no se moverá.

2.1.8 Matriz de condición de borde en nodos

La matriz de condición de borde en nodos tiene la función de asignarle a cada nodo que tipo de condición de borde se le aplica. La misma está compuesta de la siguiente forma.

$$MatCondBordeNodos = [Num\ Nodo\ Estado\_de\_cargamento\ Condicion\_de\_contorno]$$

“Num” representa el identificador de la matriz de condiciones de borde de nodos. En “Nodo” se refiere al nodo en el cual será aplicada la condición. En el “Estado$_de_cargamento”secolocaelidentificadorasociadoalam$

Por otra parte, vale aclarar que las condiciones para los nodos fijos se pueden aplicar en cualquier “Condicion_de_contorno” y valdrá para todas las simulaciones de carga.

2.1.9 Matriz de Restricción de colinealidad

Esta matriz permite mantener la colinealidad entre tres nodos a la hora de la optimización. Para aplicar esta restricción, la matriz debe estar formada de la siguiente manera.

$$MatResCol = [Num\ Nodo\_Cenrtal\ Nodo_1\ Nodo_2]$$

“Num” representa el identificador. Con “Nodo_Central”, como dice la palabra, se refiere al nodo que estará en entre medio de los nodos de los extremos (Nodo1 y Nodo2).

2.1.10 Matriz de Restricción de dislocamiento

Esta matriz tendera validez únicamente a la hora de la optimización donde se le puede decir al código el desplazamiento máximo para un nodo en determinado eje cuando se carga la estructura en cualquiera de sus simulaciones.

$$MatResDes = [Num\ Nodo\ Grado\_de\_libertad\ T_{min}\ T_{max}\ D_{min}\ D_{max}]$$

"Num" representa el identificador. En "Nodo" se debe colocar el nodo al cual se le aplica la restricción. "Grado_de_libertad" puede tomar valores de 1, 2 o 3, los cuales representan las variables x, y y z respectivamente. "T_min" y "T_max toman valores de 1 o 0 y le dan la orden de si existe o no un dislocamiento mínimo o máximo respectivamente. En caso de que exista dislocamiento mínimo o máximo, los valores de los mismos vienen dados por "D_min" y "D_max".

2.2 Montaje de matrices

Una vez definida la estructura y ciertas restricciones en la misma se procede de la siguiente manera. Siempre se debe correr el código "AnalisisInit" ya que este es el que organiza al programa global definiendo matrices y sus tamaños para ser utilizadas en los demás códigos. Dentro de este se corren los siguientes códigos: "MountVF", para crear el vector de fuerzas externas, "MountMK", para crear la matriz de rigidez, "MountVecU", para hallar, mediante un análisis lineal, los desplazamientos nodales y por ultimo "MountSigma", para hallar las tensiones en bielas del reticulado. Luego se puede utilizar los códigos "Plot" y "Plot_U" para representar gráficamente la torre y la torre deformada con cierta escala y el estado de carga que el usuario le asigne. Además, se pueden hallar las frecuencias naturales y ver los modos de la estructura utilizando el código "Plot_Modo" y también hallar inestabilidades estructurales para las fuerzas aplicadas.

3 Conclusion

References