

**UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA**

**FACULTAD DE INGENIERÍA**

**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

---



**DESARROLLO DE SOFTWARE EDUCATIVO PARA EL ANÁLISIS DE ARMADURAS ESPACIALES  
USANDO SCILAB**

**POR:**

**MANUEL DE JESÚS RINCÓN CAMACHO**

**TESIS PRESENTADO COMO REQUISITO PARA OBTENER EL GRADO DE**

**MAESTRO EN ESTRUCTURAS**

**CHIHUAHUA, CHIH., MÉXICO**

**OCTUBRE 2022**



Desarrollo de Software Educativo para el Análisis de Armaduras Espaciales usando Scilab; Tesis presentada por Manuel de Jesús Rincón Camacho como requisito parcial para obtener el grado de Maestro en Estructuras, ha sido aprobado y aceptado por:

---

**M.I. Fabián Vinicio Fernández Hernández.**  
Director de la Facultad de Ingeniería

---

**Dr. Fernando Martínez Reyes.**  
Secretario de Investigación y Posgrado

---

**M.A. María de Lourdes Flores Portillo.**  
Coordinador(a) Académico

---

**Dr. Mario Alberto Madrid Pérez.**  
Director(a) de Tesis

---

Octubre 2022

**COMITÉ**  
**Dr. Mario Alberto Madrid Pérez.**  
**M.S.C Rubén Castañeda Balderas.**  
**M.E. Mirna Teresita Armendáriz Portillo.**  
**M.E. Oscar Adrián Montes Hernández.**

Derechos Reservados  
**Manuel de Jesús Rincón Camacho**  
Circuito Universitario No. 1, Campus  
Universitario 2 C.P. 31125  
Chihuahua, Chih. México  
Octubre 2022



UNIVERSIDAD AUTÓNOMA DE  
**CHIHUAHUA**

20 de octubre de 2022.

**M.I. MANUEL DE JESÚS RINCÓN CAMACHO**  
**Presente**

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestro en Estructuras, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Mario Alberto Madrid Pérez** para que lo desarrolle como tesis, con el título: **"DESARROLLO DE SOFTWARE EDUCATIVO PARA EL ANÁLISIS DE ARMADURAS ESPACIALES USANDO SCILAB"**.

### **Índice de Contenido**

- Agradecimientos
- Índice de contenido
- Índice de figuras
- Capítulo 1. Introducción
- Capítulo 2. Marco teórico
  - 2.1 Armaduras espaciales
  - 2.2 Método de rigidez
- Capítulo 3. Metodología
  - 3.1 Código motor
  - 3.2 Interfaz gráfica
- Capítulo 4. Validación del código motor
- Capítulo 5. Conclusiones y trabajos futuros
- Referencias
- Apéndice
- Currículum vitae

**FACULTAD DE INGENIERÍA**  
Circuito No.1, Campus Universitario 2  
Chihuahua, Chih., México. C.P. 31125  
Tel. (614) 442-95-00  
[www.fing.uach.mx](http://www.fing.uach.mx)



UNIVERSIDAD AUTÓNOMA DE  
CHIHUAHUA

Solicitamos a Usted tomar nota de que el título del trabajo se imprima en lugar visible de los ejemplares del documento terminal.

**ATENTAMENTE**

*"Naturam subiecit aliis"*

EL DIRECTOR

M.I. FABIÁN VINICIO HERNÁNDEZ  
MARTÍNEZ

FACULTAD DE  
INGENIERÍA  
U.A.CH.



DIRECCIÓN

EL SECRETARIO DE INVESTIGACIÓN

Y POSGRADO

DR. FERNANDO MARTÍNEZ REYES

FACULTAD DE INGENIERÍA  
Circuito No.1, Campus Universitario 2  
Chihuahua, Chih., México. C.P. 31125  
Tel. (614) 442-95-00  
[www.fing.uach.mx](http://www.fing.uach.mx)

## **Agradecimientos**

A Dios por permitirme lograr este sueño de culminar mi maestría en estructuras y poder dedicarme a esta hermosa área de la ingeniería.

Al amor de mi vida, Nancy Graciela Favela Rodríguez que siempre ha estado a mi lado en las buenas y en las malas y con su amor y apoyo he logrado culminar, a mi hijo Iker Eduardo Rincón Favela por su amor y cariño y esperando que algún día pueda ser un gran y exitoso profesionalista.

A mis padres Julio César Rincón Rincón y María Elena Camacho de la Cruz quienes me dieron la vida y siempre me enseñaron que el esfuerzo y el estudio son la base del éxito y aunque los tengo lejos de mí siempre me dieron su apoyo incondicional en los proyectos emprendidos. A mi hermano Julio Cesar Rincón Camacho quien siempre me ha dado su apoyo y aunque no está conmigo en Chihuahua me manda sus buenas vibras y el gusto de verlo siempre exitoso.

A mis suegros Concepción Favela Balderrama y María Graciela Rodríguez Pacheco quienes han sido para mí como unos segundos padres, pues desde que me casé con su hija me adoptaron como un hijo más y siempre me han dado su apoyo y cariño.

A mi amigo y compañero Mario Alberto Madrid Pérez quien siempre ha sido un gran apoyo para mí y fue mi director de tesis, esperando a futuro podamos seguir trabajando e investigando.

A Rubén Castañeda Balderas quien fue parte fundamental para el desarrollo de esta tesis, por su tiempo para apoyarnos en la programación y su revisión.

A Mirna Teresita Armendáriz Hernández quien a parte de haber sido mi maestra, me ha dado su apoyo y amistad, gracias por compartirme tus conocimientos y ser tan profesional en tu trabajo.

A Oscar Adrián Montes Hernández quien fue mi alumno y ahora tengo el gran honor de que sea mi revisor y sinodal, gracias por su dedicación y profesionalismo.

A María de Lourdes Flores Portillo además de ser mi coordinadora siempre desde que ingrese a la facultad ha sido una gran amiga y siempre me ha dado su apoyo para lograr mis proyectos.

A mis compañeros de secretaria académica, con quien durante 9 años compartimos alegrías y tristezas, dándome su apoyo para salir adelante y lograr este éxito, los quiero y extraño.



## Índice de Contenido

Agradecimientos .....	III
Índice de Contenido.....	V
Capítulo 1: Introducción .....	VI
Capítulo 2: Marco Teórico .....	XII
2.1 Armaduras Espaciales.....	XII
2.2 Método de Rigidez .....	XIII
2.2.1 Sistemas de Coordenadas .....	XIV
2.2.2 Matriz de rigidez coordenadas locales .....	XV
2.2.3 Matrices de Transformación.....	XVIII
2.2.4 Matriz de Rigidez en Coordenadas Globales.....	XXII
2.2.5 SCILAB.....	XXV
Capitulo 3: Metodología .....	XXVII
3.1 Código Motor.....	XXVII
3.2 Interfaz gráfica .....	XXXII
Capitulo 4. Validación del código motor.....	XXXVIII
Capitulo 5. Conclusiones y trabajos futuros .....	LV
Referencias.....	LVIII
Apéndice.....	LX
Curriculum Vitae.....	LXVII



## Capítulo 1: Introducción

Una de las áreas de interés para la Ingeniería Civil es el estudio y el comportamiento de las estructuras que pueden ser definidas como un conjunto de elementos unidos con la cualidad de soportar sin deformaciones considerables a lo largo del tiempo las acciones de carga y agentes externos a los que está sometida.

La estructura soporta las cargas y las transporta a través de fuerzas internas en los diferentes elementos que la componen, así mismo se asocia a cada efecto interno un estado de esfuerzos que es absorbido por el material que la constituye<sup>1</sup>.

Para entender el estudio de una estructura se recurre al análisis estructural que se define como la predicción del desempeño de una estructura ante las cargas o solicitaciones y agentes externos tales como movimientos en los apoyos (asentamientos) y cambios de temperatura (deformación térmica).

En la actualidad sabemos que existen muchas teorías y métodos para el análisis estructural, por lo que es necesario hablar un poco históricamente para entender desde donde surgieron y como fueron evolucionando con el tiempo.

Para situarnos en un inicio de la historia comenzamos con los griegos en el año 500 A.C. donde sus obras se basaban en el uso de la piedra como elemento principal y se pudieron ver grandes obras como el templo de Hera y el túnel de Eupalinos. Durante ese mismo periodo de tiempo, Aristóteles y Arquímedes establecían los principios de la estática.

En la era de los romanos la cual se distinguió por el uso de metales, madera y mampostería, se pudieron ver nuevas estructuras como el arco, el marco y las





bóvedas, pero todos sus hallazgos en construcción estaban basados en experiencias, prácticas y manejo de formas y no en conocimientos analíticos.

El uso del concreto en la construcción también se les atribuye a los romanos cuya principal obra fue la cúpula del Panteón de Agripa de Roma.

Durante la Edad Media (500-1500) la construcción de edificios como catedrales que se caracterizaban por arcos apuntados estabilizados horizontalmente por arbotantes voladores que fueron desarrollados por constructores góticos supuso en sus inicios un conocimiento, aunque rudimentario, de la estática.

Antes del Renacimiento no se tenían conocimientos aplicados sobre la teoría de las estructuras, se construían sin cálculos, pero apoyándose de un código de la buena práctica, código o manual cuyos contenidos fueron cimentando progresivamente el arte y la ciencia de construir estructuras: la "Ingeniería Estructural"<sup>1</sup>.

En el siglo XVII aparece el gran Galileo Galilei (1564-1642) quién estableció las primeras bases de la Mecánica y Resistencia de Materiales, tratando de determinar la resistencia a la falla de una ménsula empotrada con carga en un extremo Figura 1.1, intuyendo que la resistencia estaba relacionada al momento en el empotramiento, pero obtuvo resultados erróneos al suponer que la distribución de esfuerzos consistían en un par de fuerzas puntuales cuyo brazo era el espesor de la ménsula.

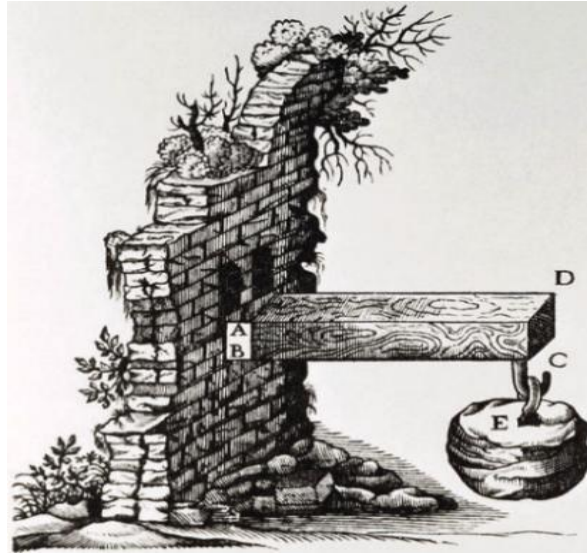


Figura 1.1 Experimento Galileo Galilei, viga en voladizo<sup>3</sup>.

Galileo no consideró la deformación de los materiales ante la flexión y es entonces Roberto Hooke (1635-1703) quien en 1660 emite la explicación más sencilla ante el fenómeno de la deformación con la conocida forma "ut tensio sic vis", es decir: como la extensión es la fuerza.

Fue Mariotte (1620-1684) quien aplicó la ley de Hooke al problema de Galileo llegando a la conclusión de que la resistencia a flexión procede de la extensión y contracción de las fibras de la ménsula (1680).

Johann Bernouilli, en 1705, estudia y concluye que el momento flector generado por la acción de las cargas en una viga prismática es proporcional a la curvatura de la misma, lo que dio lugar a estudios posteriores de Euler sobre la ecuación diferencial de la elástica y sobre la aplicación de métodos variacionales a problemas de flexión en vigas (1744) y pandeo de columnas (1757).

La gran mayoría de los métodos clásicos conocidos de análisis estructural se desarrollaron entre el siglo XIX y XX. Los principales colaboradores en este periodo incluyen a B.P. Clapeyron (1835-1884), quien formuló el teorema de los Tres Momentos, J.C. Maxwell (1831-1879), impulsor del método de la deformación constante y de la ley de las deflexiones recíprocas; Otto Mohr (1835-1918),



desarrollador del método de la viga conjugada y del Círculo de Mohr; Alberto Castigliano (1847-1884), quien formuló el teorema del trabajo mínimo; C. E. Green (1842-1925), creador del método de los teoremas de Mohr; H. Muller-Breslau (1851-1925), quien presentó el concepto de líneas de influencia; G.A. Maney (1888-1947), autor del método de deflexión-pendiente, y a quien se le considera el precursor del método matricial de las rigideces; y Hardy Cross (1885-1947), quien desarrolló el método de la distribución de momentos en 1924<sup>2</sup>.

Con la evolución tecnológica llegó la era de las computadoras que representó un avance en el análisis pues se podrían plantear problemas mucho más complejos y la solución podría ser exacta, resolviendo sistemas de ecuaciones a gran escala, ganando su terreno el método de cálculo matricial (método de rigideces)<sup>3</sup>.

Así mismo con la revolución tecnológica se dio paso a la simulación con el desarrollo de software y con esto predecir el comportamiento de una estructura ante la acción de cargas sin necesidad de llevarla a la realidad para estudiarla.

El método de cálculo matricial es la esencia de los software de análisis ya que fundamenta el poder transformar una estructura sea viga, marco o armadura en un sistema de ecuaciones expresado en matriz y resolverlo para obtener reacciones en los apoyos y fuerzas internas en los elementos, mismas que serán utilizadas para el cálculo de esfuerzos y deformaciones.

Existen diversos programas para el análisis estructural cómo son Ram elements y SAP 2000 empleados para modelación y análisis de estructuras, que a su vez serán utilizados para validar los resultados de esta tesis.

El programa SAP 2000 es un software especializado que emplea el método de elementos finitos para el diseño y análisis, implementado dentro de una interfaz gráfica 3D orientada a objeto, Figura 1.2.



Una de sus desventajas radica en que no es un software gratuito, además de que al modelar se hace uso de líneas guía que complican el dibujo sobre todo en estructuras que tengan demasiados elementos, generando una posible equivocación. Una licencia de este programa en forma básica anda entre los \$6000 y \$6500 usd, lo cual no es accesible para un estudiante de licenciatura.

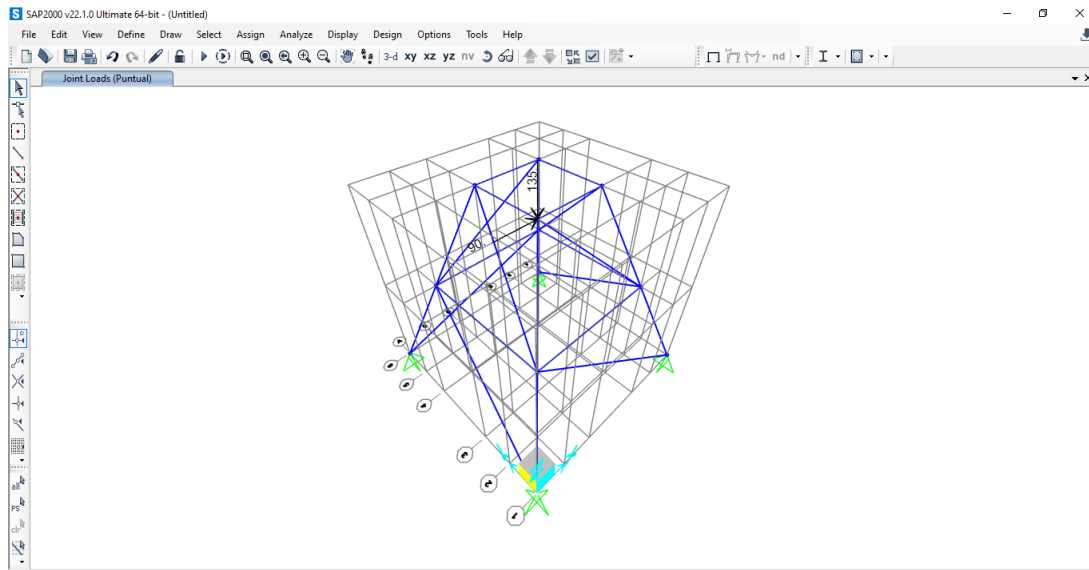


Figura 1.2 Modelado en SAP 2000.

El programa Ram Elements tiene un sistema más amigable para el modelado ya que este trabaja con coordenadas para establecer los nodos de nuestra estructura, se puede trabajar con sistemas de 2 y 3 dimensiones y su código de análisis está basado en el método de rigideces o método matricial. Se puede ver su interfaz de trabajo en la figura 1.3.

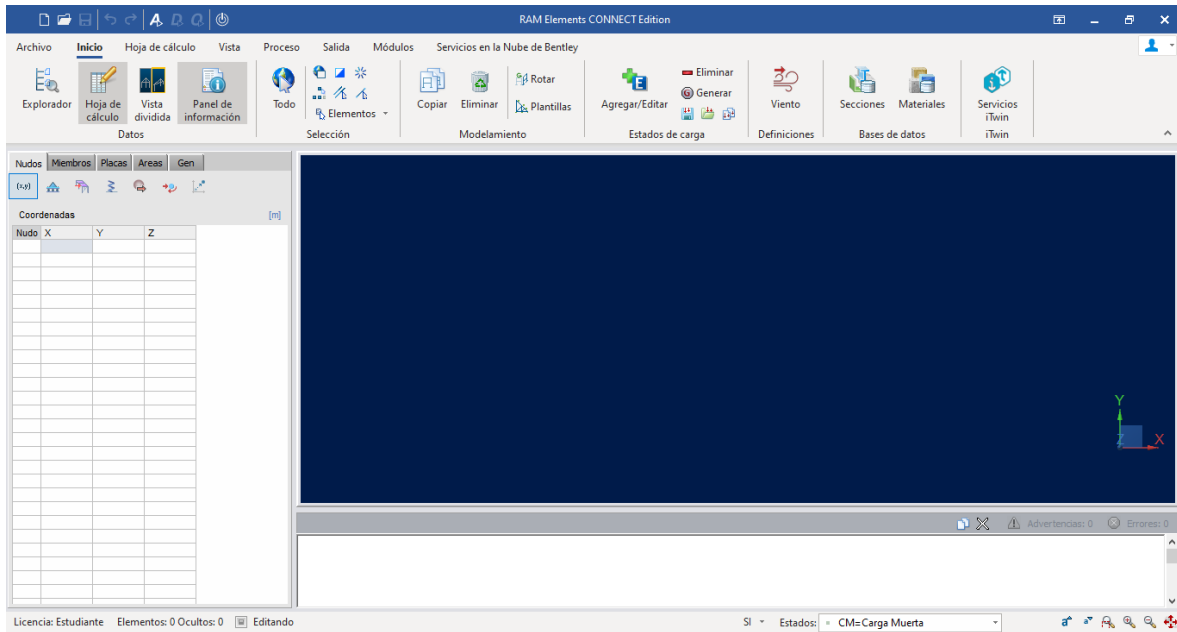


Figura 1.3 Interfaz de trabajo Ram Elements<sup>11</sup>.

La principal limitante de este software es que no es gratuito por lo que no es tan accesible, se cuenta con una licencia de tipo estudiantil con límite en el número de nodos y elementos a analizar. Una licencia cuesta aproximadamente \$ 3400 usd.

Finalmente podemos decir que el objetivo principal de esta tesis es el desarrollo de un código motor en Scilab empleando un análisis elástico del método de rigideces para el análisis de armaduras espaciales, así como desarrollar una interfaz gráfica que permita la interacción del usuario para el ingreso de nodos, elementos, secciones transversales y materiales, generando al final un archivo de salida que muestre las reacciones en apoyos, fuerzas y esfuerzos en los elementos y deformaciones.



## Capítulo 2: Marco Teórico

### 2.1 Armaduras Espaciales

Nuestro desarrollo está basado en el estudio y análisis de armaduras espaciales por lo que primeramente esta debe definirse.

Una armadura espacial (Figura 2.1) se define como un ensamble tridimensional de elementos prismáticos rectos conectados en sus extremos por nudos sin fricción, y sometido a cargas y reacciones que actúan sólo en los nudos. Al igual en las armaduras planas los miembros de las armaduras espaciales desarrollan solo fuerzas axiales<sup>4</sup>.

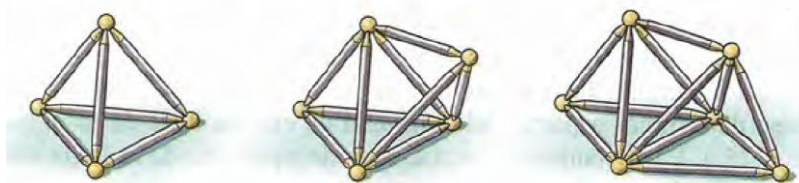


Figura 2.1 Armaduras Espaciales<sup>5</sup>.

El elemento más simple de una armadura tridimensional es un tetraedro el cual se forma al conectar seis elementos barra entre sí (Figura 2.2 a). Una armadura espacial simple puede construirse agregando tres elementos barra a la configuración básica de tetraedro (elementos AE, BE Y CE ) y uniéndolos a los tres nudos existentes y conectándolos en un nuevo nudo (Figura 2.2 b)

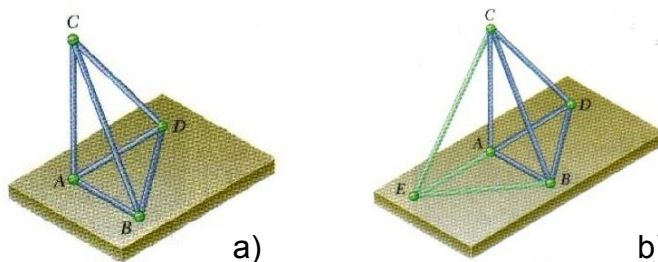


Figura 2.2. Tetraedro formado por elementos barra<sup>5</sup>.



Para el análisis de una estructura las características de interés en el desempeño de la misma son:

- Esfuerzos o fuerzas internas, tales como fuerzas axiales, fuerzas cortantes, momentos de flexión y momentos de torsión.
- Deflexiones.
- Reacciones en los apoyos<sup>2</sup>.

## 2.2 Método de Rigidez

El método de rigidez es un método general de análisis de estructuras, que puede aplicarse para resolver tanto estructuras isostáticas como hiperestáticas.

Se puede entender como una estructura isostática aquella que puede ser analizada y resuelta empleando únicamente las ecuaciones de equilibrio de la estática, mientras que una estructura hiperestática se considera que cuenta con más incógnitas comparadas con el número de ecuaciones de equilibrio y por lo tanto se requiere de métodos de análisis específicos para realizar el estudio del comportamiento de la estructura.

El método consiste en identificar el número de grados de libertad que determinan la deformación de la estructura (giros o desplazamientos), satisfaciendo las condiciones de compatibilidad.

El número de incógnitas del problema es, pues, igual al grado de indeterminación cinemática del problema. En principio, el número de éstas es:

$$k = gl \times nn - ca \quad (1)$$

donde "  $gl$  " es el número de grados de libertad a considerar por nodo, "  $nn$  " es el número de nudos de la estructura y "  $ca$  " es el número de grados de libertad prescritos por las condiciones de apoyo.



El método de rigidez, se basa en los tres principios fundamentales de la mecánica de estructuras:

I. Compatibilidad. La deformación es una función continua y tiene un valor único en cada punto. En consecuencia, los desplazamientos también lo son y, en particular, los desplazamientos en los extremos que concurren en un mismo nodo son idénticos para todos los elementos.

II. Equilibrio. Tanto la estructura está en equilibrio estático, bajo la acción de las fuerzas exteriores como los esfuerzos internos.

III. Linealidad. La estructura se comporta linealmente tanto a nivel local (relación esfuerzo-deformación) como a nivel global (relaciones desplazamiento-deformación y fuerzas-esfuerzos).

### **2.2.1 Sistemas de Coordenadas**

Tanto para la estructura como para los elementos que la forman se utilizan sistemas de coordenadas ortogonales y cartesianas, es preciso distinguir entre el sistema de coordenadas generales, globales o de la estructura y el sistema de coordenadas particulares, locales o del elemento, por lo que es necesario primeramente definirlos para distinguir su diferencia.

- **Coordenadas globales o de la estructura**

Este sistema se denomina así porque la posición de los nudos, las cargas aplicadas, desplazamientos y las reacciones en apoyos referentes a la estructura en su conjunto, están referidos a un único sistema de ejes  $x$ ,  $y$  y  $z$ , Figura 2.3.



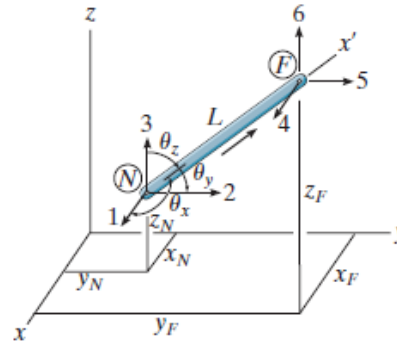


Figura 2.3 Coordenadas Globales y Locales<sup>8</sup>.

- **Coordenadas locales o de los elementos**

Todas las propiedades de los elementos así como las dimensiones, propiedades geométricas y fuerzas internas, deben referirse a un sistema particular de coordenadas, que es definido al asignarle una orientación al elemento, es decir, al indicar cuál es su nudo inicial y cual el final. Se supone entonces que el sentido positivo del eje  $x'$  local, va del nudo inicial al nudo final; los otros ejes locales quedan automáticamente definidos por la regla de la mano derecha.

En resumen, todo lo que se refiere a nudos deberá expresarse en coordenadas globales y todo lo relacionado con elementos deberá referirse a las coordenadas locales correspondientes. Lo mismo es aplicable a los diferentes resultados.

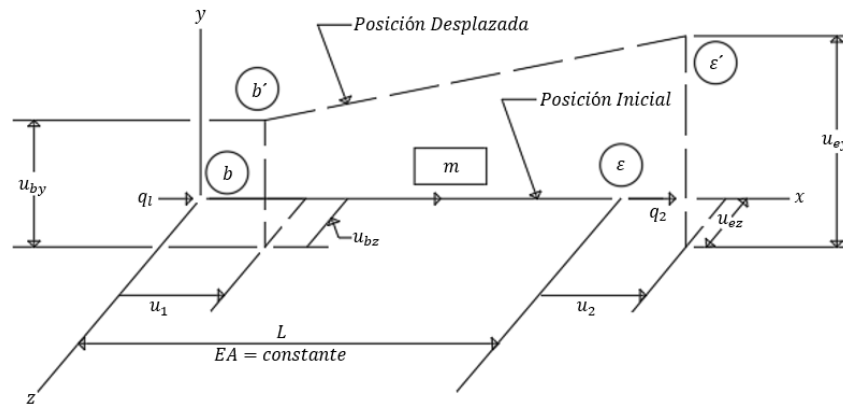
Para estructuras en el espacio como es el caso de armaduras espaciales la relación entre coordenadas globales y locales se refieren mediante los ángulos directores<sup>7</sup>, Figura 2.3.

## 2.2.2 Matriz de rigidez coordenadas locales

Para establecer las relaciones de rigidez local de los elementos debemos centrar nuestra atención en un elemento prismático arbitrario  $m$  de una armadura espacial. Cuando la armadura está sujeta a las cargas externas, el miembro  $m$  se



deforma y se inducen fuerzas axiales en sus extremos. Las posiciones inicial y deformada del miembro se muestran en la Figura 2.4 a.



(a) Fuerzas y Desplazamientos de los Miembros en el Sistema Local de Coordenadas

Figura 2.4a Fuerzas y Desplazamientos en coordenadas Locales<sup>4</sup>.

Como indica esta figura, tres desplazamientos: (traslaciones en las direcciones  $x$ ,  $y$  y  $z$ ) son necesarios para especificar completamente la posición desplazada de cada final del miembro. Así, el miembro tiene un total de seis grados de libertad o desplazamientos finales.

Sin embargo, pequeños desplazamientos en los extremos en las direcciones perpendiculares al eje centroidal de la armadura no causan ninguna fuerza en el miembro. Por lo tanto al final los desplazamientos  $u_{by}$ ,  $u_{bz}$ ,  $u_{ey}$  y  $u_{ez}$  en las direcciones de los ejes locales  $y$  y  $z$  no se evalúan en el análisis; para efectos analíticos, se considera que el miembro tiene solo dos grados de libertad,  $u_1$  y  $u_2$ , en su sistema de coordenadas local.

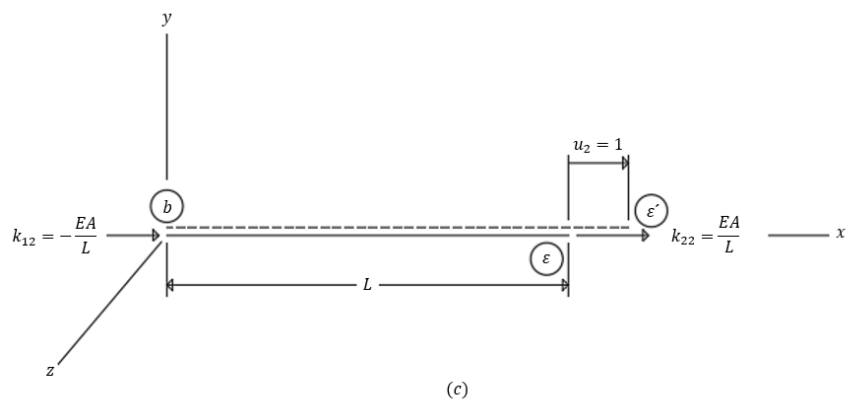
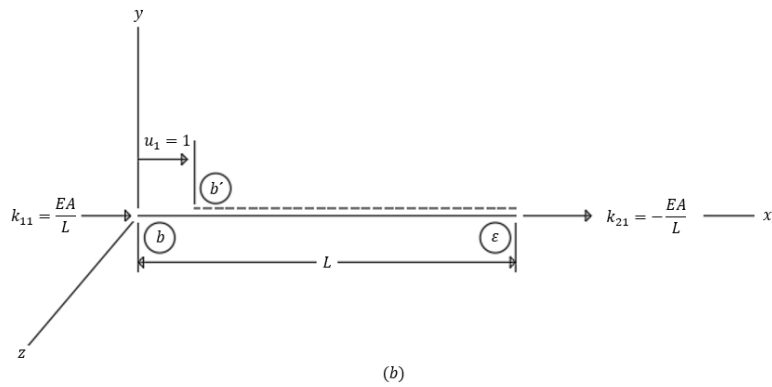


Figura 2.4 b, c Fuerzas y Desplazamientos en coordenadas Locales<sup>4</sup>.

La relación entre las fuerzas locales en los extremos  $q$  y los desplazamientos en los extremos  $u$ , para los elementos de las armaduras espaciales, se escribe como:

$$q = k x u \quad (2)$$

donde  $k$  representa la matriz de rigidez del elemento de  $2 \times 2$  en el sistema de coordenadas local. La forma explícita de  $k$  se puede obtener sometiendo al miembro a los desplazamientos finales unitarios,  $u_1 = 1$  y  $u_2 = 1$ , como se muestra en las Figura 2.4 b y Figura 2.4 c respectivamente, y evaluando las correspondientes fuerzas en los extremos de los miembros dando la siguiente matriz de rigidez local<sup>4</sup>.

$$k = \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad (3)$$



Una vez expresada la matriz de rigidez en coordenadas locales es necesario expresarla en coordenadas globales esto debido a que en una armadura espacial cada elemento tiene su propia dirección de ejes locales y para dar un resultado general estos deben referirse a un solo eje de coordenadas globales, para hacer este cambio se requieren dos matrices de transformación, matriz de transformación de desplazamientos y matriz de transformación de cargas.

### 2.2.3 Matrices de Transformación

Considere un miembro arbitrario  $m$  de una armadura espacial, como se muestra en la Figura 2.5 a, y sean  $X_b, Y_b, Z_b$ , y  $X_e, Y_e, Z_e$  las coordenadas globales de los nudos a los que se unen los extremos b y e, respectivamente.

La longitud y los ángulos directores de los elementos respecto a los ejes globales se pueden obtener mediante las siguientes expresiones y pueden verse representados en la Figura 2.5

$$\lambda_x = \text{Cos } \theta_x = \frac{X_F - X_N}{L} = \frac{X_F - X_N}{\sqrt{(X_F - X_N)^2 + (Y_F - Y_N)^2 + (Z_F - Z_N)^2}} \quad (4a)$$

$$\lambda_y = \text{Cos } \theta_y = \frac{Y_F - Y_N}{L} = \frac{Y_F - Y_N}{\sqrt{(X_F - X_N)^2 + (Y_F - Y_N)^2 + (Z_F - Z_N)^2}} \quad (4b)$$

$$\lambda_z = \text{Cos } \theta_z = \frac{Z_F - Z_N}{L} = \frac{Z_F - Z_N}{\sqrt{(X_F - X_N)^2 + (Y_F - Y_N)^2 + (Z_F - Z_N)^2}} \quad (4c)$$

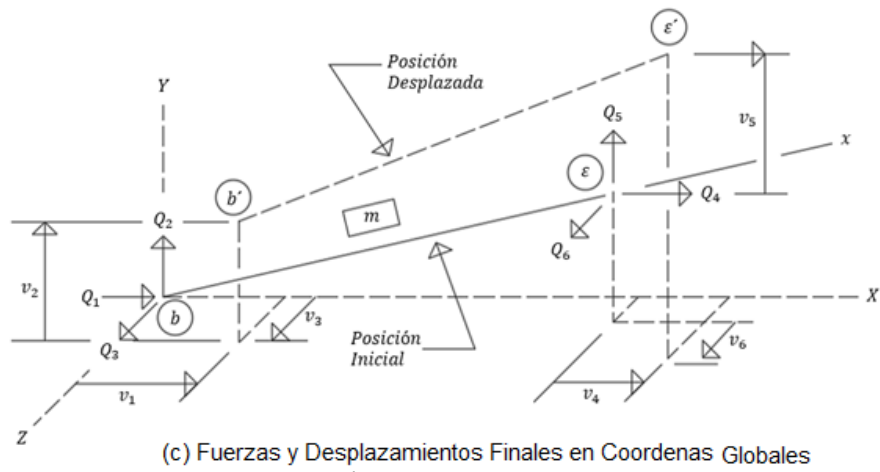
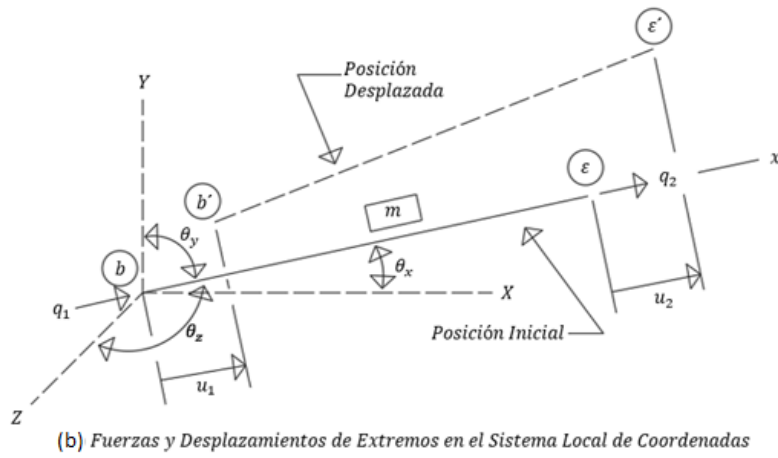
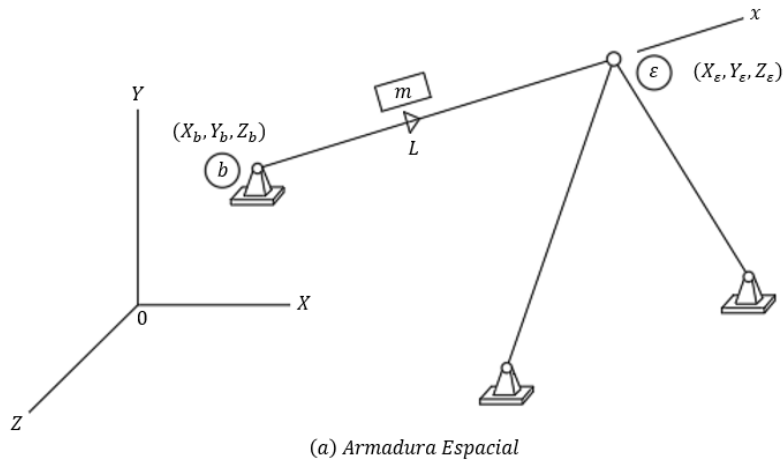


Fig. 2.5 Fuerzas y desplazamientos en elementos<sup>4</sup>.



Consideremos la transformación de las fuerzas y los desplazamientos en los extremos de un sistema de coordenadas global a uno local. Comparando las Figura 2.5 b y Figura 2.5 c, observamos que en el extremo b del elemento, la fuerza local  $q_1$  debe ser igual a la suma algebraica de las componentes de las fuerzas globales  $Q_1$ ,  $Q_2$  y  $Q_3$  en la dirección de el eje x local; es decir:

$$q_1 = Q_1 \cos \theta_x + Q_2 \cos \theta_y + Q_3 \cos \theta_z \quad (5a)$$

$$q_2 = Q_4 \cos \theta_x + Q_5 \cos \theta_y + Q_6 \cos \theta_z \quad (5b)$$

Escritas de forma matricial se tiene

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \cos \theta_x & \cos \theta_y & \cos \theta_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \cos \theta_x & \cos \theta_y & \cos \theta_z \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{bmatrix} \quad (6)$$

Simbólicamente (6) se puede expresar como  $q = T \times F$  y en función de desplazamientos se tiene que  $u = T \times v$ , donde  $T$  representa a la matriz de transformación de fuerzas y considerado que  $\cos \theta_x = \lambda_x$ ,  $\cos \theta_y = \lambda_y$ ,  $\cos \theta_z = \lambda_z$  se tiene:

$$T = \begin{bmatrix} \lambda_x & \lambda_y & \lambda_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & \lambda_z \end{bmatrix} \quad (7)$$

A continuación, examinamos la transformación de las fuerzas en los extremos de los miembros a partir del sistema de coordenadas global. Una comparación de las Figuras 2.5 (b) y (c) indica que en el extremo b del miembro, las fuerzas globales  $Q_1$ ,  $Q_2$  y  $Q_3$  deben ser las componentes de la fuerza local  $q_1$  en las direcciones de los ejes globales  $X$ ,  $Y$  y  $Z$ , respectivamente. Esto es:

$$Q_1 = q_1 \cos \theta_x \quad (8a)$$

$$Q_2 = q_1 \cos \theta_y \quad (8b)$$



$$Q_3 = q_1 \cos \theta_z \quad (8c)$$

Similarmente en el extremo e del elemento, las fuerzas globales  $Q_4$ ,  $Q_5$  y  $Q_6$  pueden expresarse como componentes de la fuerza local  $q_2$  como:

$$Q_4 = q_2 \cos \theta_x \quad (9a)$$

$$Q_5 = q_2 \cos \theta_y \quad (9b)$$

$$Q_6 = q_2 \cos \theta_z \quad (9c)$$

Expresando ecuaciones (8a, 8b, 8c) y (9a, 9b, 9c) de manera matricial se tiene lo siguiente:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{bmatrix} = \begin{bmatrix} \cos \theta_x & 0 \\ \cos \theta_y & 0 \\ \cos \theta_z & 0 \\ 0 & \cos \theta_x \\ 0 & \cos \theta_y \\ 0 & \cos \theta_z \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ \lambda_z & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \\ 0 & \lambda_z \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (10)$$

La matriz izquierda de (10), representa la matriz transpuesta  $T^T$  de la matriz de transformación  $T$  de (7) por lo que simbólicamente se puede escribir como  $Q = T^T x q$  y considerando los desplazamientos se escribe como  $v = T^T x u$ .

:



## 2.2.4 Matriz de Rigidez en Coordenadas Globales

Partimos de (2) basados en la ley de Hooke,

$$q = k x u$$

A partir de la matriz de la transformación de desplazamientos se obtuvo que  $u = T x v$ , sustituyendo en (2) se tiene:

$$q = k(Tv) \quad (11)$$

Multiplicando por  $T^T$  ambos lados de la ecuación se obtiene:

$$T^T q = T^T k T v \quad (12)$$

A partir del análisis de la matriz de transformación de fuerzas se obtuvo que  $Q = T^T q$ , dando lo siguiente:

$$Q = T^T k T v \quad (13)$$

De (13) puede verse que se tienen las fuerzas  $Q$  globales y los desplazamientos  $v$  globales por los que entonces  $T^T k T = K$ , representa la matriz de rigidez elemental en coordenadas globales.

$$K = \begin{bmatrix} \lambda_x & 0 \\ \lambda_y & 0 \\ \lambda_z & 0 \\ 0 & \lambda_x \\ 0 & \lambda_y \\ 0 & \lambda_z \end{bmatrix} \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_x & \lambda_y & \lambda_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & \lambda_z \end{bmatrix}$$





$$K = \frac{AE}{L} \begin{pmatrix} Nx & Ny & Nz & Fx & Fy & Fz \\ \lambda_x^2 & \lambda_x \lambda_y & \lambda_x \lambda_z & -\lambda_x^2 & -\lambda_x \lambda_y & -\lambda_x \lambda_z \\ \lambda_y \lambda_x & \lambda_y^2 & \lambda_y \lambda_z & -\lambda_y \lambda_x & -\lambda_y^2 & -\lambda_y \lambda_z \\ \lambda_z \lambda_x & \lambda_z \lambda_y & \lambda_z^2 & -\lambda_z \lambda_x & -\lambda_z \lambda_y & -\lambda_z^2 \\ -\lambda_x^2 & -\lambda_x \lambda_y & -\lambda_x \lambda_z & \lambda_x^2 & \lambda_x \lambda_y & \lambda_x \lambda_z \\ -\lambda_y \lambda_x & -\lambda_y^2 & -\lambda_y \lambda_z & \lambda_y \lambda_x & \lambda_y^2 & \lambda_y \lambda_z \\ -\lambda_z \lambda_x & -\lambda_z \lambda_y & -\lambda_z^2 & \lambda_z \lambda_x & \lambda_z \lambda_y & \lambda_z^2 \end{pmatrix} \begin{pmatrix} Nx \\ Ny \\ Nz \\ Fx \\ Fy \\ Fz \end{pmatrix} \quad (14)$$

Dejando la ecuación final para el análisis por el método de rigideces como:

$$F = K v \quad (15)$$

Los pasos para resolver una estructura según el Método de Rigidez son:

1. Definir la geometría de la estructura, así como las condiciones de apoyo y las acciones.
2. Identificar el número de desplazamientos incógnita que determinan la deformación de la estructura, a base de considerar las correspondientes condiciones de compatibilidad en los nodos, desplazamientos conocidos  $v_k$  y desplazamientos desconocidos  $v_u$ .

$$\{v\} = \begin{Bmatrix} v_u \\ v_k \end{Bmatrix} \quad (16)$$

3. Resolver los elementos individuales, en función de los movimientos de sus extremos usando (14), a base de satisfacer las condiciones de equilibrio y compatibilidad en los elementos y posteriormente realizar un ensamble de las matrices individuales para crear la matriz general de la estructura.
4. Imponer las necesarias condiciones de equilibrio en los nudos (fuerzas conocidas  $Q_u$ ).



5. Imponer las condiciones de apoyo de la estructura (fuerzas desconocidas  $Q_K$ ).

$$\{Q\} = \begin{Bmatrix} Q_K \\ Q_U \end{Bmatrix} \quad (17)$$

6. Determinar los movimientos incógnita (desplazamientos desconocidos), a base de resolver el sistema de ecuaciones resultante.

$$\begin{Bmatrix} Q_K \\ Q_U \end{Bmatrix} = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{Bmatrix} v_u \\ v_k \end{Bmatrix} \quad (18)$$

Escribiendo estas ecuaciones en forma simultánea tenemos

$$Q_k = k_{11}v_u + k_{12}v_k \quad (19a)$$

$$Q_u = k_{21}v_u + k_{22}v_k \quad (19b)$$

7. Determinar las reacciones, fuerzas y esfuerzos en la estructura.

De (19a) despejamos los desplazamientos desconocidos teniendo

$$v_U = K_{11}^{-1} * [Q_K - K_{12}v_k] \quad (20)$$

Después de obtener todos los desplazamientos nodales se procede a calcular las fuerzas externas en la estructura esto es resolver directamente usando (19b).

Cabe mencionar que en muchos casos, como el que se presenta ahora, los desplazamientos restringidos son cero reduciendo el cálculo de vectores de carga y desplazamiento desconocidos a:

$$v_U = K_{11}^{-1} * Q_K$$

$$Q_U = K_{21} * v_U$$



Para el cálculo de fuerzas locales de los elementos se parte de (11)

$$q = k(Tv)$$

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} * \begin{bmatrix} \lambda_x & \lambda_y & \lambda_z & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_x & \lambda_y & \lambda_z \end{bmatrix} * \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}$$

Sabemos que  $q_1$  y  $q_2$  tienen la misma magnitud así que basta solo calcular cualquiera de las dos, dado que  $q_2$  está en tensión (Figura 2.5b), calcularemos solo  $q_2$  de la siguiente forma:

$$q_2 = \frac{AE}{L} \begin{bmatrix} -\lambda_x & -\lambda_y & -\lambda_z & \lambda_x & \lambda_y & \lambda_z \end{bmatrix} * \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix}$$

Para el cálculo de esfuerzos se utiliza la formula de esfuerzo normal promedio:

$$\sigma = E\delta/L \quad (21)$$

El desarrollo del sistema de ecuaciones es complejo y requiere del uso de una computadora para su solución y es por ello que se opta por desarrollar un software basado en un lenguaje de programación con filosofía libre y gratuita como es Scilab que a continuación se describirá brevemente.

### 2.2.5 SCILAB

Es un programa de código abierto implementado en un sólo ambiente para tener herramientas de programación, cálculo numérico, y gráficos. Se considera que fue creado por el INRIA (Institut Nationale de Recherche en Informatique et en Automatique) y el ENPC (Ecole Nationale des Ponts et Chaussées) de Francia. Es similar a MATLAB y otros programas de cálculo numérico. Puede ser utilizado en una variedad de sistemas operativos.

Desde el punto de vista del usuario, Scilab presenta algunas ventajas tales como:



- Disponibilidad de versiones en la web;
- El programa es legal para ser utilizado, copiado y divulgado;
- Los resultados puede ser utilizados para divulgación sin ninguna restricción;
- Se tiene acceso al código principal<sup>12</sup>.



## Capítulo 3: Metodología

En este capítulo se contemplan dos partes, en la primera se va a abordar el desarrollo del código motor y en la segunda el desarrollo de la interfaz gráfica.

### 3.1 Código Motor

El código motor es una serie de funciones y procedimientos que conforman entre sí un algoritmo con la capacidad de efectuar los cálculos necesarios de acuerdo a la información recibida a través de la interfaz gráfica, es decir que el motor tiene la capacidad de adaptarse de acuerdo a los requerimientos de un usuario final, restringido primordialmente por los formatos propios del cálculo de rigidez y por las capacidades del lenguaje de programación.

El código desarrollado inicia leyendo la estructura de datos que proporciona la interfaz gráfica, el primer paso es la lectura de las coordenadas ingresadas y con esto asigna para cada uno de los nodos su coordenada en  $x$ ,  $y$  y  $z$ .

```
pestananodos=hojacalculo(1)
NUMERONODOval=pestananodos.value(:,1)
NN=max(NUMERONODOval)
for i=1:NN
... NUMERONODO(i,1)=NUMERONODOval(i+1,1)
end
coorXval=pestananodos.value(:,2)
for i=1:NN
... coorX(i,1)=coorXval(i+1,1)
end
coorYval=pestananodos.value(:,3)
for i=1:NN
... coorY(i,1)=coorYval(i+1,1)
end
coorZval=pestananodos.value(:,4)
for i=1:NN
... coorZ(i,1)=coorZval(i+1,1)
end
```

El siguiente paso hace una lectura de los datos de elementos donde asigna para cada elemento un nodo inicial (NI) y nodo final (NF) para establecer el sentido positivo de los ejes locales en cada elemento, posteriormente lee la estructura de datos de los valores asignados para área (A) y módulo de elasticidad (E) y los asigna como propiedades en cada elemento, mostrando una tabla de: *Elemento*, *NI*, *NF*, *A*, *E*.



```
pestaanelementos=hojacalculo(2)
NUMERODEELEMENTOval=pestaanelementos.value(:,1)
NE=max(NUMERODEELEMENTOval)
for i=1:NE
... NUMERODEELEMENTO(i,1)=NUMERODEELEMENTOval(i+1,1)
end
Nival=pestaanelementos.value(:,2)
for i=1:NE
... NI(i,1)=Nival(i+1,1)
end
NFval=pestaanelementos.value(:,3)
for i=1:NE
... NF(i,1)=NFval(i+1,1)
end
Aval=pestaanelementos.value(:,4)
for i=1:NE
... A(i,1)=Aval(i+1,1)
end
Eval=pestaanelementos.value(:,5)
for i=1:NE
... E(i,1)=Eval(i+1,1)
end
```

De acuerdo a la teoría del método expuesta en el capítulo 2 una vez asignados los elementos se debe proceder a crear cada una de las matrices de rigidez elementales, es decir la matriz individual por elemento, para esto de entrada se tiene que cada nodo de la armadura cuenta con 3 grados de libertad y con esto se define que el orden de la matriz global de la estructura será 3 veces el número de nodos, a cada elemento se le calculan los cosenos directores utilizando las ecuaciones (4a), (4b) y (4c), y con los resultados se usa la ecuación (7) para crear la matriz de transformación de desplazamientos, finalmente se calcula la matriz de rigidez elemental en coordenadas globales con la ecuación  $\mathbf{k} = \mathbf{T}^T \mathbf{k} \mathbf{T}$ .

```
L=zeros(NE,1);
LX=zeros(NE,1);
LY=zeros(NE,1);
LZ=zeros(NE,1);
NGL=3*NN;
KACUM=zeros(NGL,NGL);
for i=1:NE
... K=zeros(NGL,NGL);
... L(i,1)=sqrt((coorX(NF(i,1),1)-coorX(NI(i,1),1))^2 + (coorY(NF(i,1),1)-coorY(NI(i,1),1))^2 + (coorZ(NF(i,1),1)-coorZ(NI(i,1),1))^2);
... LX(i,1)=(coorX(NF(i,1),1)-coorX(NI(i,1),1))/L(i,1);
... LY(i,1)=(coorY(NF(i,1),1)-coorY(NI(i,1),1))/L(i,1);
... LZ(i,1)=(coorZ(NF(i,1),1)-coorZ(NI(i,1),1))/L(i,1);
... KKG=A(i,1)*E(i,1)/L(i,1)*[1 -1; -1 1];
... T=[LX(i,1) LY(i,1) LZ(i,1) 0 0 0; 0 0 0 LX(i,1) LY(i,1) LZ(i,1)];
... K=K+T'*KKG*T;
```

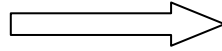
Para cada matriz obtenida se realiza el ensamble es decir que se acomodan los valores obtenidos en una matriz que tiene el orden de la matriz global, finalmente se suman todas las matrices ensambladas y se obtiene la matriz global de la estructura, al ser un software de tipo educativo se pide al programa que imprima cada matriz elemental y la matriz global. Se muestra un ejemplo de ensamblaje un elemento que soporta carga axial en 2D.



$$\lambda_x = \frac{3-0}{3} = 1 \quad \lambda_y = \frac{0-0}{3} = 0$$

$$k_1 = AE \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0.333 & 0 & -0.333 & 0 \\ 0 & 0 & 0 & 0 \\ -0.333 & 0 & 0.333 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

Matriz elemental



$$K = AE \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0.333 & 0 & -0.333 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.333 & 0 & 0.333 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}$$

Matriz ensamblada

Continuando con el flujo de información y el algoritmo propuesto se leen las restricciones establecidas considerando el valor de 1 si está restringido y 0 si está libre y con estos datos se genera una tabla que muestra: Nodo, TX, TY, TZ imprimiendo los grados de libertad restringidos correspondientes a las restricciones marcadas con 1 y los grados de libertad no restringidos correspondientes a los marcados con 0.

```
TXval=pestananodos.value(:,5)
for i=1:NN
    TX(i,1)=TXval(i+1,1)
end
TYval=pestananodos.value(:,6)
for i=1:NN
    TY(i,1)=TYval(i+1,1)
end
TZval=pestananodos.value(:,7)
for i=1:NN
    TZ(i,1)=TZval(i+1,1)
end
mprintf('\nRestricciones de la estructura. (1=restricción, -0=libre).\n\n')
disp('Nodo...TX...TY...TZ')
nodosrestringidos=[NUMERONODO, TX, TY, TZ];
disp(nodosrestringidos)
```

Ahora según la teoría explicada y las ecuaciones 19a y 19b es necesario generar cuatro submatrices a partir de la matriz de rigidez global ( $k_{11}v_u, k_{12}v_k, k_{21}v_u, k_{22}v_k$ ) y estas están en función de los grados de libertad restringidos  $v_k$  y no restringidos  $v_u$ .

$k_{11}v_u$  es una matriz de orden de grados de libertad no restringidos x grados de libertad no restringidos y contienen los valores de rigidez correspondientes a las posiciones de los grados de libertad no restringidos. Se muestra el código utilizado para su planteamiento.



```
K1DU=zeros(NGLNR,NGLNR);
for m=1:NGLNR
    for n=1:NGLNR
        K1DU(m,n)=K(VNGLNR(m,1),VNGLNR(n,1));
    end
end
K1DK=zeros(NGLNR,NGLR);
for m=1:NGLNR
    for n=1:NGLR
        K1DK(m,n)=K(VNGLNR(m,1),VNGLR(n,1));
    end
end
```

$k_{12}v_k$  es una matriz de orden de grados de libertad no restringidos x grados de libertad restringidos y contienen los valores de rigidez correspondientes a las posiciones de los grados de libertad no restringidos x las posiciones de los grados de libertad restringidos. Se muestra el código utilizado para su planteamiento.

```
K1DK=zeros(NGLNR,NGLR);
for m=1:NGLNR
    for n=1:NGLR
        K1DK(m,n)=K(VNGLNR(m,1),VNGLR(n,1));
    end
end
```

$k_{21}v_u$  es una matriz de orden de grados de libertad restringidos x grados de libertad no restringidos y contienen los valores de rigidez correspondiente a las posiciones de los grados de libertad restringidos x las posiciones de los grados de libertad no restringidos. Se muestra el código utilizado para su planteamiento.

```
K2DU=zeros(NGLR,NGLNR);
for m=1:NGLR
    for n=1:NGLNR
        K2DU(m,n)=K(VNGLR(m,1),VNGLNR(n,1));
    end
end
```

$k_{22}v_k$  es una matriz de orden de grados de libertad restringidos x grados de libertad restringidos y contienen los valores de rigidez correspondiente a las posiciones de los grados de libertad restringidos. Se muestra el código utilizado para su planteamiento.

```
K2DK=zeros(NGLR,NGLR);
for m=1:NGLR
    for n=1:NGLR
        K2DK(m,n)=K(VNGLR(m,1),VNGLR(n,1));
    end
end
```





El siguiente paso hace que el flujo de información pase a la siguiente parte del código el cual se encarga de revisar los desplazamientos prescritos, identifica como desplazamientos conocidos ( $v_k$ ) mismos que serán necesario para la resolución del sistema de ecuaciones.

En la siguiente etapa el programa debe leer la estructura de datos generada por la interfaz de cargas en los nodos por cada eje y muestra los datos numéricos y los identifica como fuerzas conocidas  $Q_k$ , con esto se termina el pre proceso necesario para la resolución del sistema de ecuaciones y la obtención de desplazamientos desconocidos  $v_u$ , reacciones es apoyos  $Q_u$ , fuerzas internas  $q$  y esfuerzos  $\sigma$ .

```
FXval=pestananodos.value(:,11)
for i=1:NN
    ... FX(i,1)=FXval(i+1,1)
end
FYval=pestananodos.value(:,12)
for i=1:NN
    ... FY(i,1)=FYval(i+1,1)
end
FZval=pestananodos.value(:,13)
for i=1:NN
    ... FZ(i,1)=FZval(i+1,1)
end
```

Para la obtención de los desplazamientos desconocidos  $v_u$  se realizó la operación matricial contemplada en la ecuación (20), para las fuerzas desconocidas  $Q_u$  se utilizó la operación matricial contemplada en la ecuación (19b). Con los resultados el programa imprime un vector global de desplazamientos y un vector global de fuerzas en la estructura.

El cálculo de las fuerzas internas de los elementos de la estructura se realiza con la operación matricial contemplada en la ecuación (11) para el valor de  $q_2$ , y para el cálculo del esfuerzos se emplea la ley de Hooke contemplada en la ecuación (21). Para finalizar el programa calcula las coordenadas deformadas de los nodos y grafica los elementos en 3D para visualizar en color rojo la estructura deformada por la aplicación de las cargas. Esta visualización de la estructura deformada se



realiza utilizando factores de escalamiento en cada dirección, estos factores son introducidos por el usuario. Un factor de escalamiento igual a 1 indica la visualización real de la estructura deformada mientras que un factor de escalamiento igual a 10 visualiza la deformación aumentando 10 veces el tamaño de las deformaciones nodales.

Nota: El código completo se pondrá como un anexo al final de la tesis.

### **3.2 Interfaz gráfica**

En esta parte se explicará cómo está desarrollada la interfaz gráfica y la manera en que el usuario va a interactuar para dar una estructura de datos que alimente el código motor y realice el análisis de la armadura.

Todo inicia con la etapa donde el usuario va a modelar la estructura a fin de definir las condiciones geométricas de la misma. La interfaz cuenta con 3 pestañas principales NODOS, ELEMENTOS y ANÁLISIS, en la Figura 3.1 se muestra la pestaña de nodos donde el usuario tendrá varios iconos para crear, eliminar o modificar nodos, revisar coordenadas, restricciones, desplazamientos prescritos y fuerzas en los nodos. Así mismo, del lado derecho a las variables se tiene una tabla de datos donde se almacenan los nodos y sus correspondientes coordenadas ingresadas.

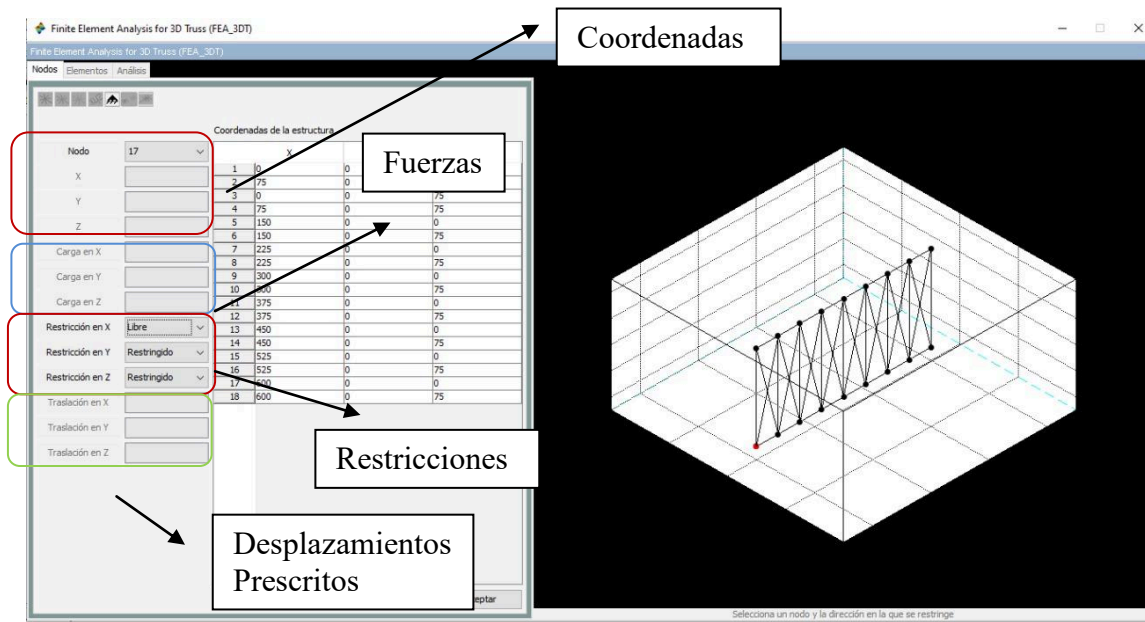


Figura 3.1 Pestaña de Nodos en la interfaz.

Para crear un nodo el usuario al dar clic se activa el panel de  $x$ ,  $y$  y  $z$ , permitiendo ingresar las coordenadas, cada vez que se requiera generar un nodo se da clic en aceptar para que se almacene la información en la tabla de datos, y se haga el dibujo en la gráfica, todos se pintan en color negro. Es importante comentar que si el usuario ingresa un dato en las coordenadas que no sea numérico el programa le indica que solamente se pueden meter valores numéricos.

Si el usuario va a eliminar un nodo al seleccionarlo éste se pinta en color rojo y al darle aceptar éste se elimina, si va a modificar alguna coordenada se pinta de color azul y se activa el panel de coordenadas para que pueda cambiar la coordenada, en ambos casos al ser eliminado o modificado la tabla de información se actualiza y, se redibujan los nodos y muestra los restantes con números consecutivos, es decir si tienes 3 nodos ingresados y se elimina el 2, el nodo 3 se convierte en 2 para el análisis. Mientras el usuario este trabajando con las coordenadas los demás paneles están inactivos.

Para ubicar un nodo en la gráfica y saber su coordenada únicamente debe dar clic en el ícono de coordenadas y seleccionar en el menú el nodo que le interesa, una



vez seleccionado aparecerán las coordenadas ingresadas y en la gráfica el nodo seleccionado cambiará a color rojo para identificarlo.

Ya que se tienen los nodos ingresados nos pasamos a la pestaña de elementos donde el usuario podrá crear, modificar o eliminar elementos, asignar perfiles y materiales.

En la Figura 3.2 se muestra la imagen de la pestaña de nodos, para un nuevo elemento el usuario deberá dar clic sobre crear elemento y se activa el panel de nodo inicial (NI) y nodo final (NF) así como la descripción del elemento, ahora el usuario podrá ir indicando los nodos de inicio y final de cada elemento además de indicar alguna referencia en la descripción, para poder generar los elementos se da clic en aceptar para que la información se almacene en la tabla ubicada en la parte inferior de la pestaña y se realice la gráfica 3D poniendo cada elemento en color negro. Mientras el usuario interactúa con los elementos los demás paneles quedan inactivos. Si el usuario va a eliminar un elemento al seleccionarlo este se pone de color rojo o si va a modificar algún elemento al seleccionarlo este se pone de color azul, una vez eliminado o modificado la tabla de información se actualiza y se redibujan los elementos restantes.

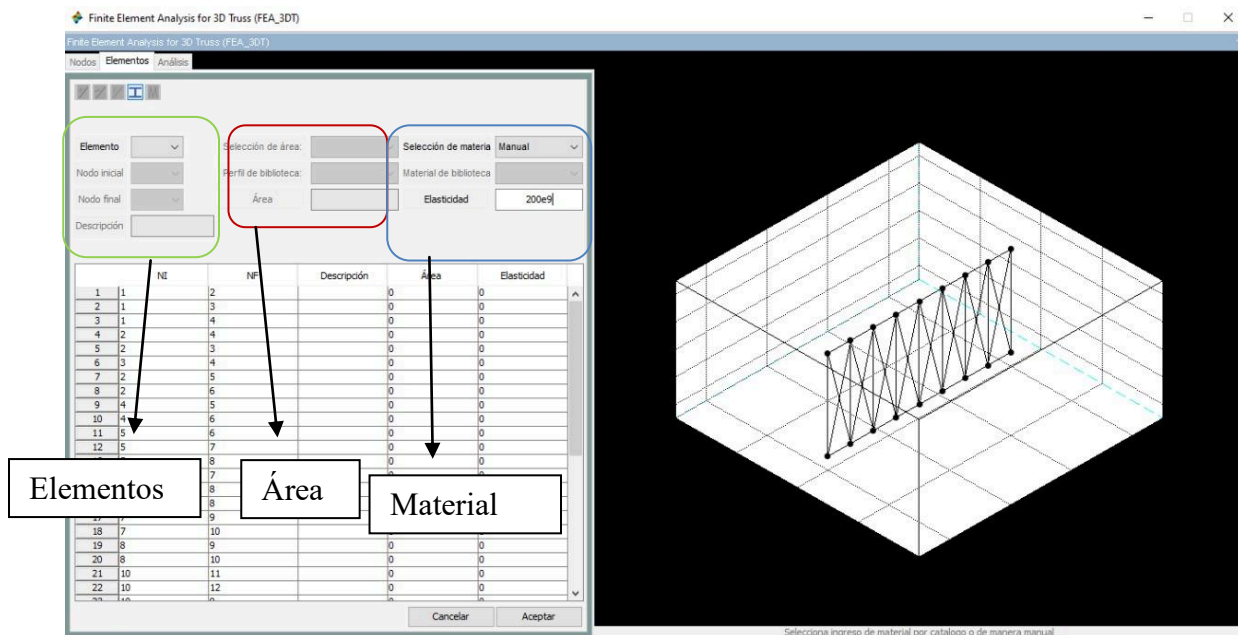


Figura 3.2 Pestaña de elementos.



Una vez creados los elementos se procede a la asignación de propiedades de la sección transversal para asignar área (A) y módulo de elasticidad (E), para esto se da clic en el ícono de perfil del elemento y viendo la Figura 3.2 se activará el panel de selección de área para elegir el elemento a asignar, el usuario tiene dos opciones, elegir la sección del perfil de biblioteca o asignar el área manualmente, es importante indicar que las áreas de los perfiles de biblioteca están en  $\text{mm}^2$  al dar clic en aceptar para cada elemento se almacena la información en la tabla, si se decide ingresar el área de manera manual es responsabilidad del usuario ingresar correctamente las unidades. Es indispensable que previo a indicar el área se haya seleccionado del menú el elemento al cual se le asignará esta área de lo contrario no se guardará en la tabla de información al dar clic en aceptar.

Para ingresar el módulo de elasticidad se da clic en el icono de material en la Figura 3.2, el cual activa el panel de selección de material para elegir el elemento a asignar, el usuario tiene dos opciones, elegir material de biblioteca o asignar el módulo de elasticidad manualmente, las unidades de la biblioteca son MPa, si el usuario decide ingresarla manualmente es su responsabilidad utilizar la unidades correctas, al dar clic en aceptar para cada elemento se almacena la información en la tabla de datos ubicada en la parte inferior de la pestaña.

Si un elemento no tiene material o perfil asignado al ejecutar el análisis se manda un mensaje diciendo el elemento no tiene área asignada o módulo de elasticidad.

Con los pasos descritos anteriormente tenemos ya concluida la parte geométrica de la armadura, ahora se procede a asignar las restricciones en los nodos, desplazamientos prescritos y fuerzas en los nodos.

Las restricciones se consideran las reacciones en los apoyos, por lo que para asignarlos se da clic en el icono de restricciones y retomando la Figura 3.1 se activará únicamente el panel de restricciones  $x$ , restricciones  $y$  y restricciones  $z$ , con esto el usuario deberá seleccionar el número de nodo y vamos a indicar para cada eje si está restringido o libre, si el nodo queda restringido en un determinado



eje el sistema no permite agregar una carga en dicho sentido, únicamente en sentidos donde quedaron libres las restricciones, para un nodo que va a quedar totalmente libre no es necesario asignar restricciones.

Si en el problema a analizar se presenta un desplazamiento prescrito es decir algún asentamiento en el sistema de apoyo o bien que el usuario quiera deformar la estructura en cierta dirección para ver su comportamiento, se procede a ingresar la información dando clic en el icono de desplazamientos prescritos y se activará únicamente el panel de traslaciones en  $x$ , traslaciones en  $y$  y traslaciones en  $z$  Figura 3.1, así podrá elegir el nodo a asignar e ingresar los valores de desplazamientos en  $x$ ,  $y$  o  $z$ . Si en alguno de los ejes no se va a manejar desplazamiento la interfaz asigna por default el valor de 0, si el nodo quedo libre no se permite ingresar datos de traslaciones, al dar clic en aceptar se guardan los datos en el sistema.

Para la asignación de cargas, se da clic en el ícono cargas nodo, y viendo la Figura 3.1 se activará únicamente el panel de cargas en  $x$ , cargas en  $y$  y cargas en  $z$ , así se podrá elegir el nodo a asignar e ingresar los valores de cargas en  $x$ ,  $y$  o  $z$ , éstas deben ingresarse en Newtons (N) para que concuerde con la biblioteca en sistema internacional, cualquier otro sistema de unidades puede utilizarse de forma manual a responsabilidad del usuario, al dar clic en aceptar estas quedan guardadas en el sistema.

Para ejecutar el análisis se va a la pestaña de análisis, al dar clic en analizar se enlazan la estructura de datos con el código motor para realizar el análisis y dar los resultados.

Una vez que finaliza el análisis sin inconvenientes se manda un mensaje en pantalla diciendo que el análisis se ha desarrollado correctamente. Cuando el análisis se desarrolla este se imprime en la consola de Scilab.



El post proceso da la opción de imprimir el reporte en un archivo .dat ( para block de notas) o en un archivo .tex (para latex). Se puede guardar el archivo en un .csv y se pueden leer archivos de Excel (como el código motor) y esto ayuda a simular estructuras grandes de manera más rápida.

Este software estará disponible en los correos electrónicos siguientes:  
mmadrid@uach.mx, mrincon@uach.mx, ruben.castaneda@cimav.edu.mx.



## Capítulo 4. Validación del código motor

La validación de todo software requiere una comparativa con respecto a otros programas de análisis estructural para evaluar la eficiencia y exactitud del desarrollo, en este caso se presentaran a continuación una serie de pruebas basadas en ejercicios que pueden dar como resultado una comparativa entre lo hecho y desarrollado en esta tesis versus los software comerciales.

Caso de Validación Número uno.

Este caso fue seleccionado debido a que en la teoría se considera la estructura más sencilla de una armadura espacial, en el cual se realizará el análisis y se presentaran resultados y su comparativa.

Para la armadura espacial mostrada se requiere determinar los desplazamientos en nodos, reacciones en apoyos, fuerzas y esfuerzos en elementos, véase la Figura 5.1, considerando que sus elementos son de sección 2L 2" x 2" x 1/4" con  $A = 1219.35 \text{ mm}^2$  y un módulo de elasticidad  $E = 200 \text{ GPa}$ .

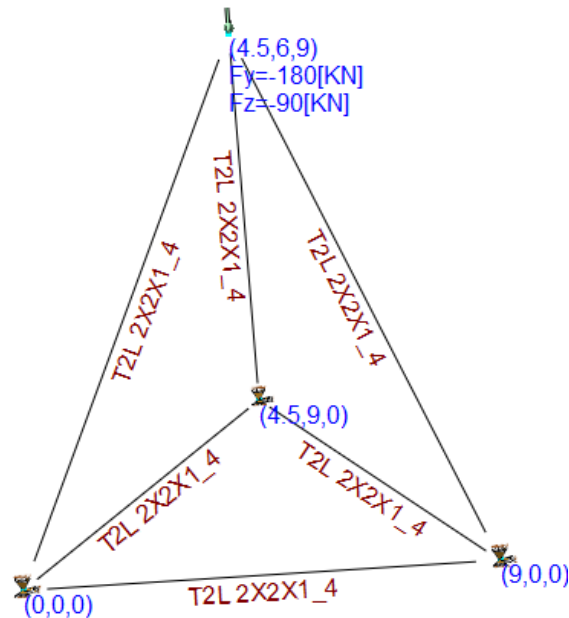


Figura 5.1 Caso de validación número uno<sup>13</sup>.





La Figura 5.2 a, b y c se muestra la estructura modelada en el software desarrollado y los dos software comparativos de análisis estructural.

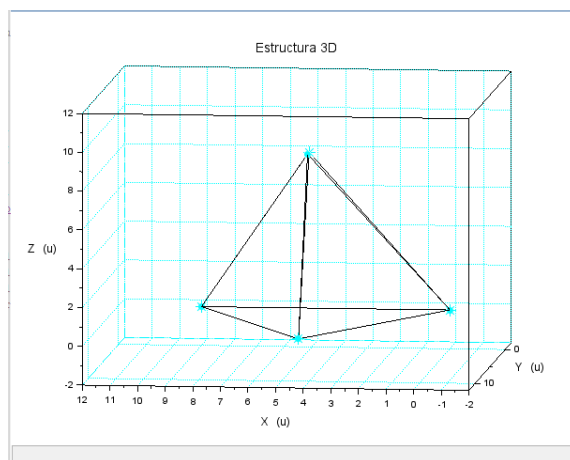


Figura 5.2 a. Modelo en Software desarrollado.

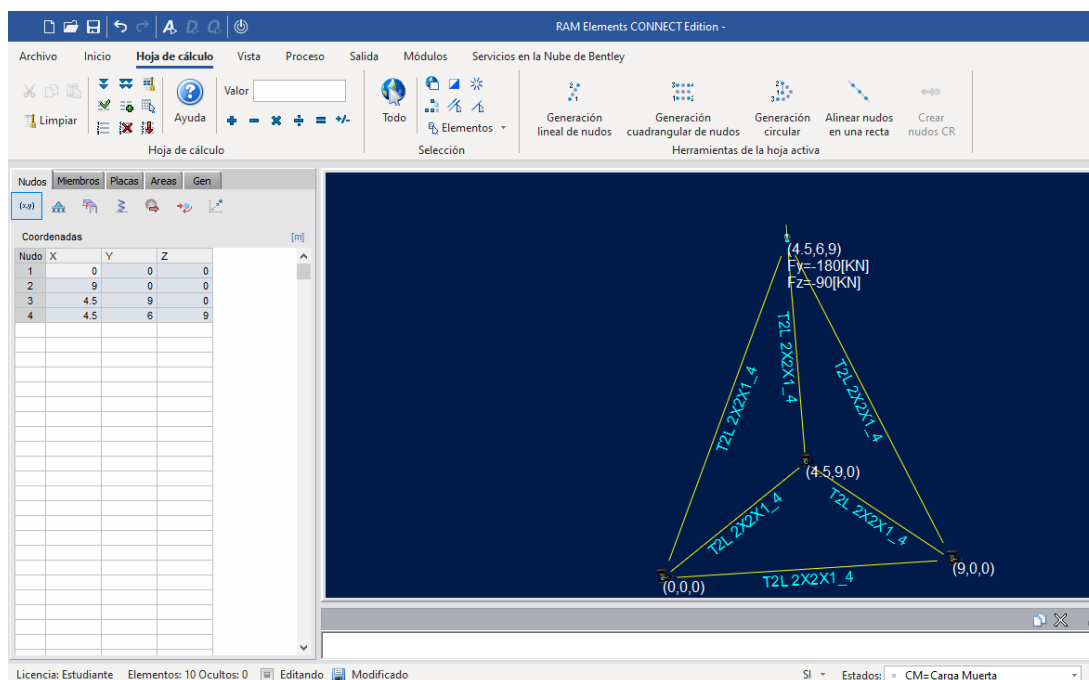


Figura 5.2 b. Modelo en Ram Elements.

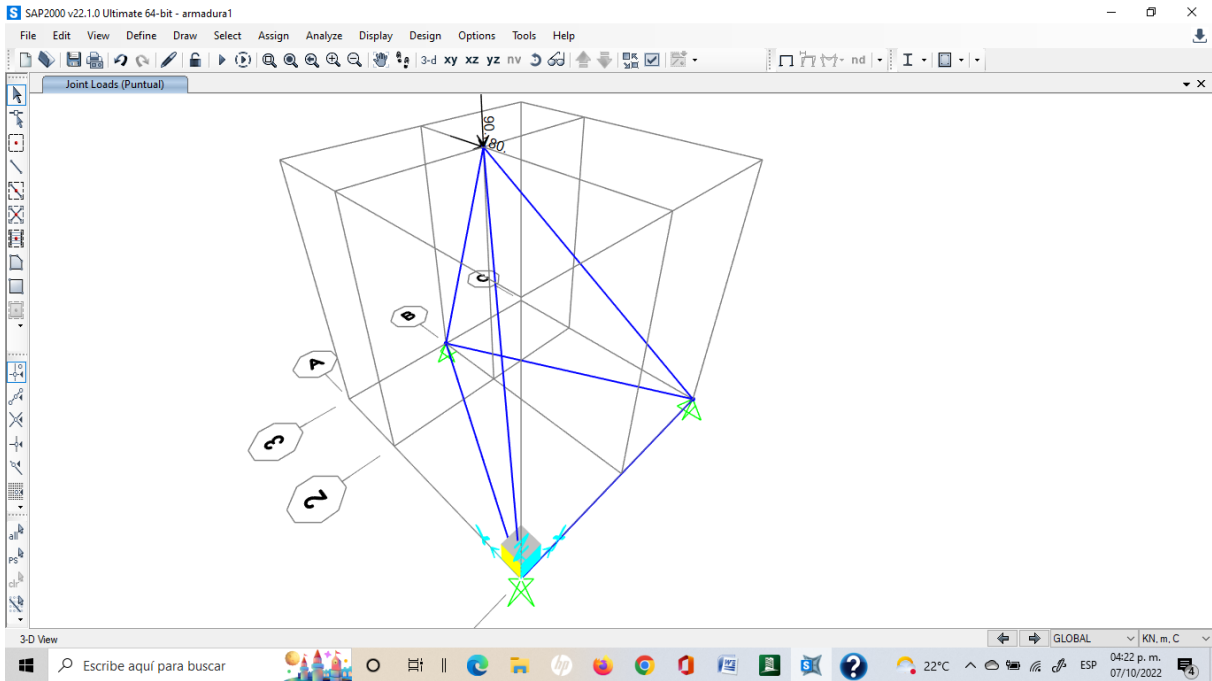


Figura 5.2 c Modelado en SAP 2000.

## Resumen Comparativo

Resultados de desplazamientos en nodos, se muestran en la Tabla 5.1

Tabla 5.1 Desplazamientos en nodos caso de validación número uno.

<b>Desplazamiento</b>	<b>Código Motor [mm]</b>	<b>Ram Elements [mm]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [mm]</b>	<b>Diferencia [%]</b>
D1	0	0	0	0	0 %
D2	-13.733	-13.737	0	-13.737	.03%
D3	0.609	0.6090	0	.6090	0 %



Resultados de reacciones en apoyos se muestran en la Tabla 5.2

Tabla 5.2 Reacciones en apoyos, caso de validación número uno

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
Q4	52.5	52.5	0	52.5	0
Q5	70	70	0	70	0
Q6	105	105	0	105	0
Q7	-52.5	-52.5	0	-52.5	0
Q8	70	70	0	70	0
Q9	105	105	0	105	0
Q10	0	0	0	0	0
Q11	40	40	0	40	0
Q12	-120	-120	0	-120	0

Resultados de fuerzas internas en elementos se muestran en la Tabla 5.3

Tabla 5.3 Fuerzas internas en elementos, caso de validación número uno.

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements, [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
q1	0	0	0	0	0
q2	0	0	0	0	0
q3	0	0	0	0	0
q4	-136.68	-136.68	0	-136.68	0
q5	-136.68	-136.68	0	-136.68	0
q6	126.49	126.49	0	126.49	0



Resultados de fuerzas internas en elementos se muestran en la Tabla 5.4

Tabla 5.4 Esfuerzos en elementos, validación número uno.

<b>Esfuerzos</b>	<b>Código Motor [kPa]</b>	<b>Ram Elements [kPa]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kPa]</b>	<b>Diferencia [%]</b>
$\sigma_1$	0	0	0	0	0
$\sigma_2$	0	0	0	0	0
$\sigma_3$	0	0	0	0	0
$\sigma_4$	-112092	-112091.80	.002	-112091.77	.002
$\sigma_5$	-112092	-112091.80	.002	-112091.77	.002
$\sigma_6$	103737	103736.30	.007	103736.3	.007

Realizando una comparativa puede verse que no hay diferencia en los resultados obtenidos, esto se debe a que nuestro desarrollo está basado en el mismo método que los demás programas.



## Caso de validación número dos

Una estructura común en el área de armaduras tridimensionales son las torres eléctricas, en este ejemplo se muestra una versión de torre eléctrica simplificada.

Para la armadura espacial mostrada se busca determinar los desplazamientos en nodos, reacciones en apoyos, fuerzas y esfuerzos en elementos, ver la Figura 5.3, considerando que sus elementos son de sección redonda de 3/4" con  $A = 284.52 \text{ mm}^2$  y considerando un módulo de elasticidad  $E = 200 \text{ GPa}$ . Carga en Nodo 9:  $F_x = 90 \text{ kN}$ ,  $F_z = -135 \text{ kN}$

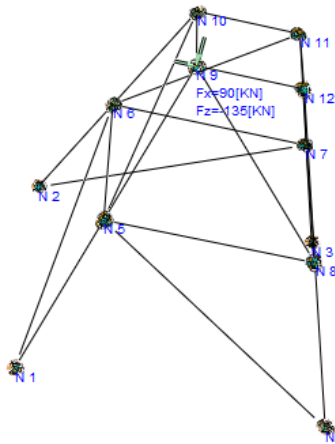


Figura 5.3 Caso de validación dos<sup>12</sup>.

Tabla de coordenadas y nodos.

Nudo	X [m]	Y [m]	Z [m]	Nudo	X [m]	Y [m]	Z [m]
1	0	0	0	7	15	15	9
2	0	18	0	8	15	3	9
3	18	18	0	9	6	6	18
4	18	0	0	10	6	12	18
5	3	3	9	11	12	12	18
6	3	15	9	12	12	6	18

La Figura 5.4 a, b y c muestra la estructura modelada en el software desarrollado y los dos software comparativos de análisis estructural.

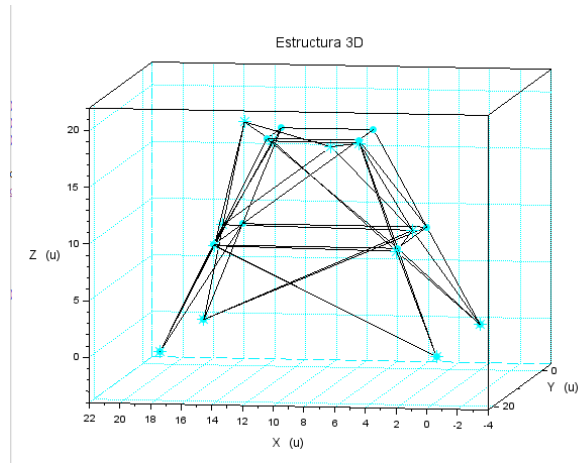


Figura 5.4 a. Modelo en Software desarrollado.

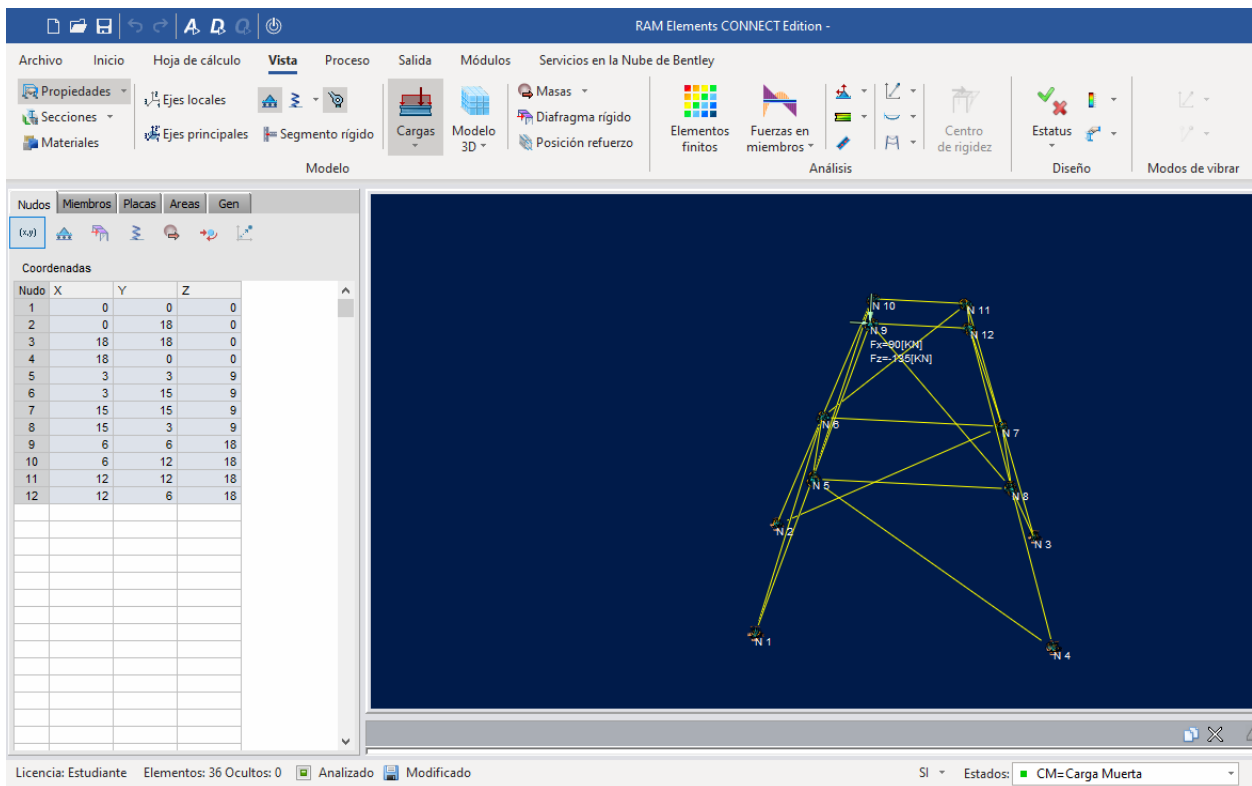


Figura 5.4 b. Modelado en Ram Elements.

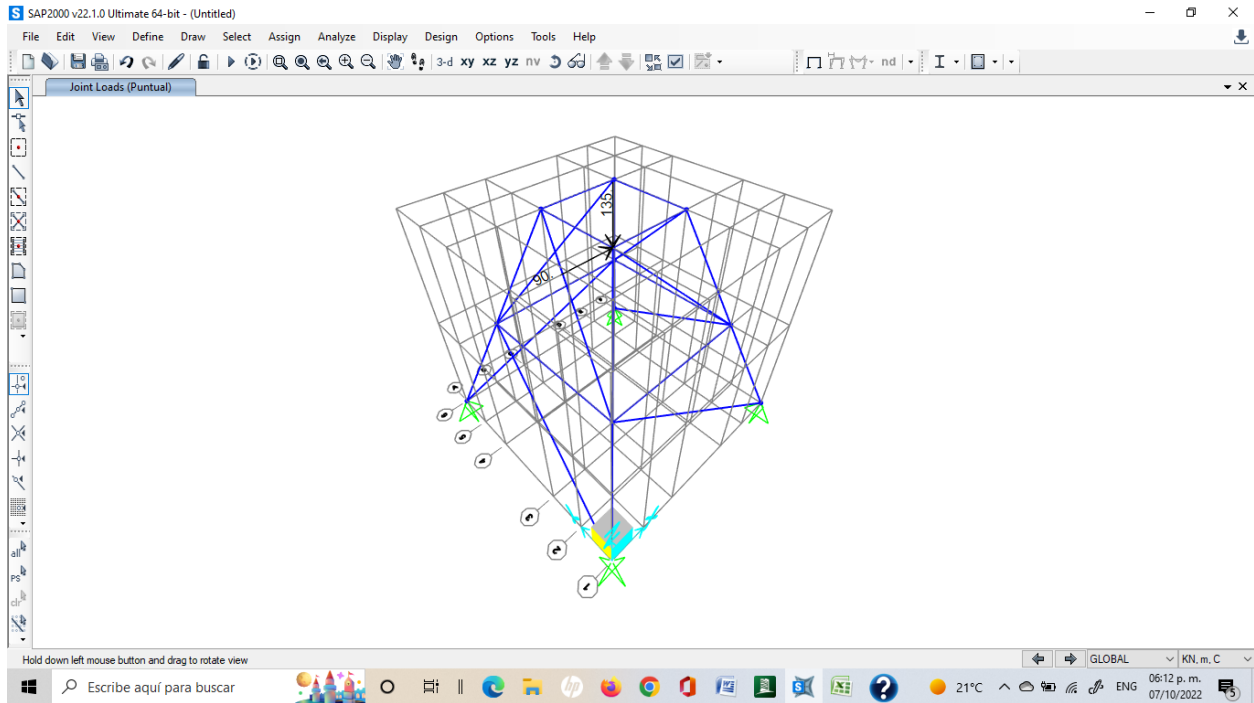


Figura 5.4 c. Modelado en SAP 2000.

## Comparativo de Resultados

Resultados de desplazamientos en nodos, se muestran en la Tabla 5.5

Tabla 5.5 Desplazamientos en nodos, caso de validación número dos.

Desplazamiento	Código Motor [mm]	Ram Elements [mm]	Diferencia [%]	SAP 2000 [mm]	Diferencia [%]
D1	24.0752	24.0594	0.066	23.913	0.68
D2	16.3772	16.3657	0.07	16.266	0.68
D3	-6.2709	-6.2669	0.064	-6.229	0.67
D4	0	0	0	0	0
D5	11.6321	11.6239	0.071	11.553	0.68
D6	-5.7403	-5.7362	0.071	-5.701	0.69
D7	0	0	0	0	0
D8	-10.8192	-10.8122	0.065	-10.746	0.68
D9	-3.6064	-3.6041	0.064	-3.582	0.68
D10	38.3100	38.2849	0.066	38.051	0.68

D11	-10.8192	-10.8122	0.065	-10.746	0.68
D12	-5.2621	-5.2587	0.065	-5.227	0.67
D13	72.2250	72.1776	0.066	71.738	0.68
D14	43.2466	43.2173	0.068	42.954	0.68
D15	-38.4901	-38.4647	0.066	-38.231	0.68
D16	-7.2132	-7.2081	0.071	-7.164	0.69
D17	38.5016	38.4755	0.068	38.241	0.68
D18	-1.5922	-1.5911	0.069	-1.581	0.71
D19	-7.2132	-7.2081	0.071	-7.164	0.69
D20	-21.6385	-21.6244	0.065	-21.493	0.67
D21	-9.6172	-9.6109	0.066	-9.552	0.68
D22	72.2250	72.1776	0.066	71.738	0.68
D23	-21.6385	-21.6244	0.065	-21.493	0.68
D24	9.6493	9.6429	0.066	9.584	0.68

Resultados de reacciones en apoyos se muestran en la Tabla 5.6.

Tabla 5.6 Reacciones en apoyos, Caso de validación número dos.

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
Q25	-15	-15	0	-15	0
Q26	-30	-30	0	-30	0
Q27	-45	-45	0	-45	0
Q28	15	15	0	15	0
Q29	-15	-15	0	-15	0
Q30	45	45	0	45	0
Q31	0	0	0	0	0
Q32	0	0	0	0	0
Q33	0	0	0	0	0
Q34	-90	-90	0	-90	0



Q35	45	45	0	45	0
Q36	135	135	0	135	0

Resultados de fuerzas internas en elementos se muestran en la Tabla 5.7

Tabla 5.7 Fuerzas internas en elementos, caso de validación número dos

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
q1	37.312	37.3120	0	37.312	0
q2	22.1853	22.1853	0	22.185	0
q3	-22.5	-22.5	0	-22.5	0
q4	-49.7494	-49.749	0	-49.749	0
q5	0	0	0	0	0
q6	0	0	0	0	0
q7	0	0	0	0	0
q8	0	0	0	0	0
q9	0	0	0	0	0
q10	-111.936	-111.936	0	-111.936	0
q11	-66.5559	-66.5559	0	-66.556	0
q12	67.5	67.5	0	67.5	0
q13	-37.312	-37.312	0	-37.312	0
q14	49.0376	49.0376	0	49.038	0
q15	-45	-45	0	-45	0
q16	-37.312	-37.312	0	-37.312	0
q17	0	0	0	0	0
q18	0	0	0	0	0
q19	0	0	0	0	0
q20	0	0	0	0	0
q21	0	0	0	0	0
q22	0	0	0	0	0

q23	-147.113	-147.113	0	-147.113	0
q24	0	0	0	0	0

Resultados de fuerzas internas en elementos se muestran en la Tabla 5.8.

Tabla 5.8 Esfuerzos en elementos, caso de validación dos.

<b>Esfuerzos</b>	<b>Código Motor [kPa]</b>	<b>Ram Elements [kPa]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kPa]</b>	<b>Diferencia [%]</b>
$\sigma_1$	131149	131022.3	0.097	130224.34	0.71
$\sigma_2$	77980	77904.35	0.097	77429.88	0.71
$\sigma_3$	-79086.1	-79009.43	0.097	-78528.23	0.71
$\sigma_4$	-174866	-174696.4	0.097	-173632.44	0.71
$\sigma_5$	0	0	0	0	0
$\sigma_6$	0	0	0	0	0
$\sigma_7$	0	0	0	0	0
$\sigma_8$	0	0	0	0	0
$\sigma_9$	0	0	0	0	0
$\sigma_{10}$	-393427	-393067	0.092	-390673.19	0.70
$\sigma_{11}$	-233927	-233713.1	0.092	-232289.57	0.70
$\sigma_{12}$	237245	237028.3	0.091	235584.62	0.70
$\sigma_{13}$	-131142	-131022.3	0.091	-130224.33	0.70
$\sigma_{14}$	172355	172197.1	0.092	171148.3	0.71
$\sigma_{15}$	-158164	-158018.9	0.092	-157056.44	0.71
$\sigma_{16}$	-131142	-131022.3	0.091	-130224.28	0.70
$\sigma_{17}$	0	0	0	0	0
$\sigma_{18}$	0	0	0	0	0
$\sigma_{19}$	0	0	0	0	0
$\sigma_{20}$	0	0	0	0	0
$\sigma_{21}$	0	0	0	0	0
$\sigma_{22}$	0	0	0	0	0

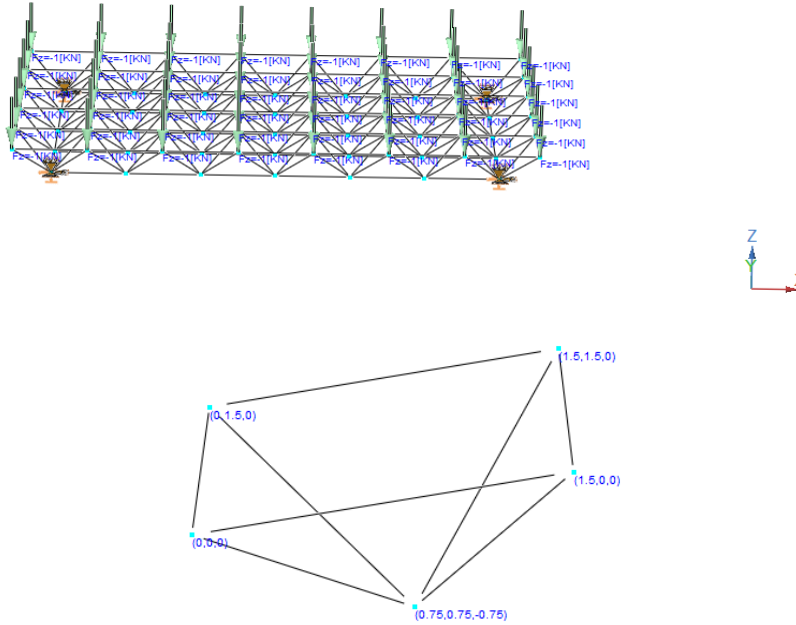
$\sigma_{23}$	-517064	-516591.2	0.092	-513445.07	0.70
$\sigma_{24}$	0	0	0	0	0

Realizando una comparativa de los resultados obtenidos es posible ver que en los 3 software se tienen prácticamente los mismos resultados, solo se ve que entre el Sap y nuestro software hay diferencia mayor en la parte de esfuerzos y eso puede ser debido a los decimales que se consideran en el área del perfil considerado.

#### Caso de validación número tres

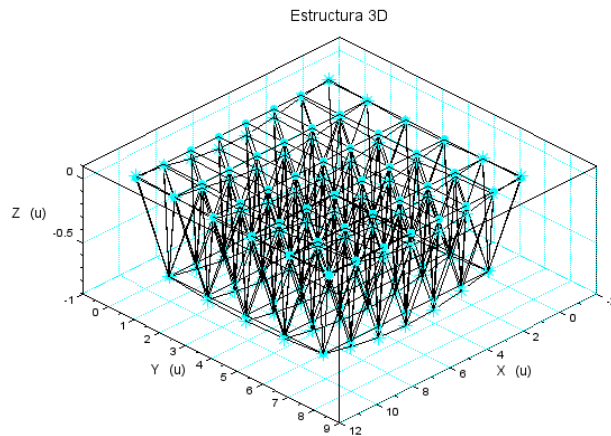
En este ejercicio se muestra una estructura muy común para techos, una estructura de tridilosa.

Para la tridilosa mostrada se busca determinar los desplazamientos en nodos, reacciones en apoyos, fuerzas y esfuerzos en elementos, ver Figura 5.5, considerando que sus elementos son de sección redonda de 3/4" con  $A = 284.52 \text{ mm}^2$  y un módulo de elasticidad  $E = 200 \text{ GPa}$ . La base para modelarlo es un tetraedro con medidas de  $1.5 \times 1.5 \text{ m}$  con  $.75 \text{ m}$  de peralte, replicándose esta sección 7 veces en dirección x y 5 veces en dirección y. Carga en todos los nodos superiores de  $-1 \text{ kN}$  en el eje z.



**Figura 5.5 Caso de validación tres**

En la Figura 5.6 a, b y c se muestra el modelado en el software desarrollado y los dos programas de comparativa.



**Figura 5.6 a. Modelo en Software desarrollado.**

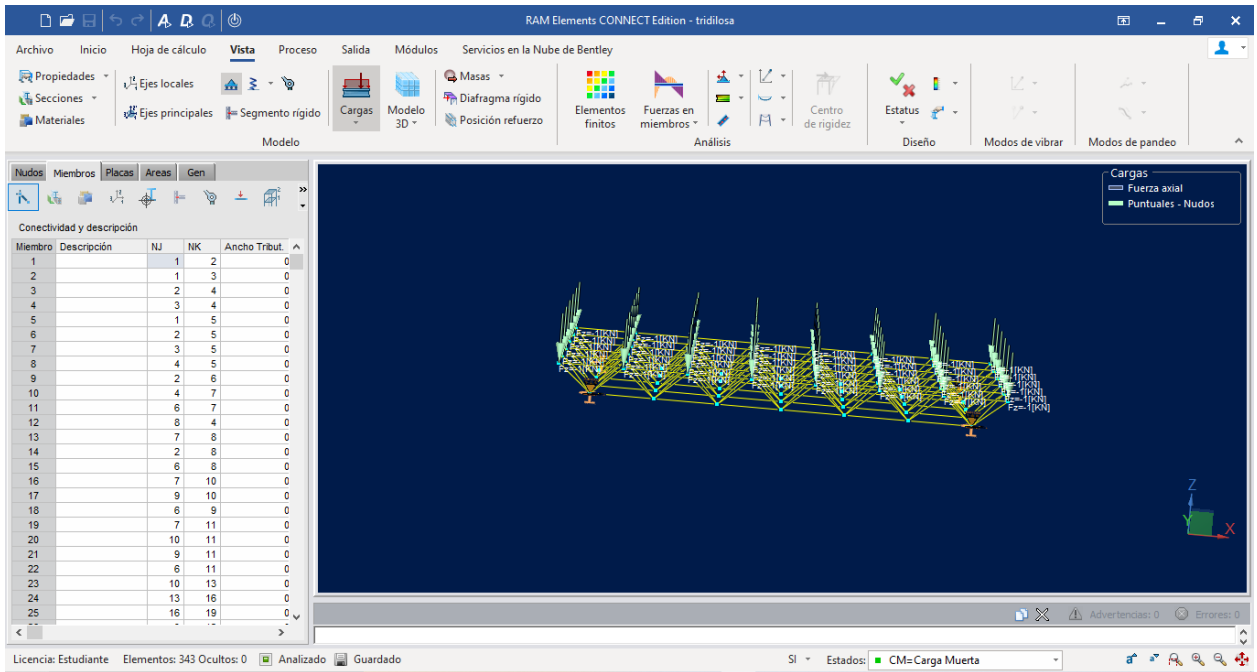


Figura 5.6 b. Modelado en Ram Elements.

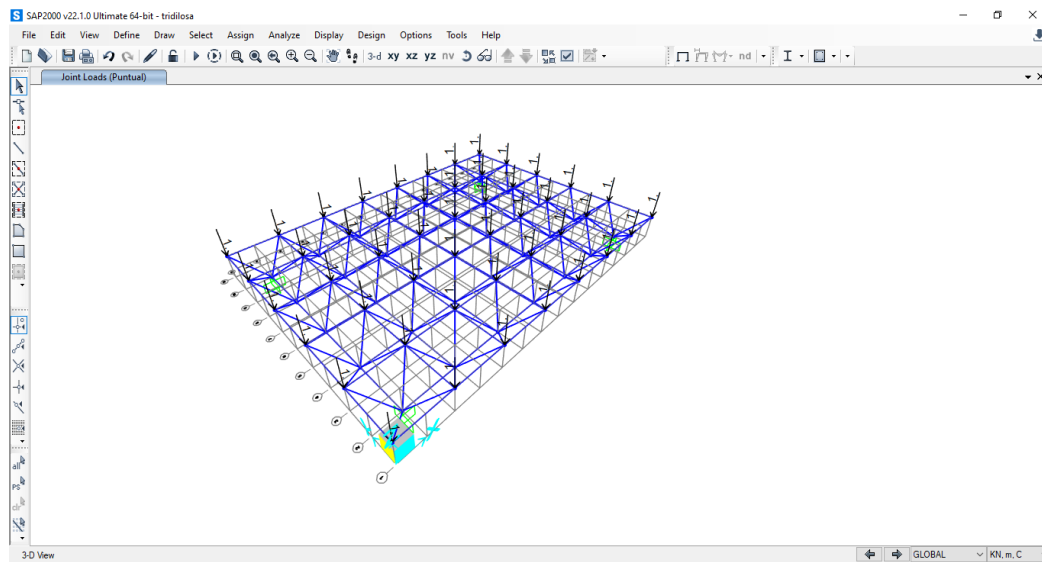


Figura 5.6 c. Modelado en SAP 2000.



## Comparativa de Resultados

Resultados de reacciones en apoyos, se muestran en la Tabla 5.9.

Tabla 5.9 Reacciones en apoyos, caso de validación tres.

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
Q238	11.906	11.906	0	12.336	3.49
Q239	8	8	0	8	0
Q240	12	12	0	12	0
Q241	-11.906	-11.906	0	-12.336	3.49
Q242	8	8	0	8	0
Q243	12	12	0	12	0
Q244	11.906	11.906	0	12.336	3.49
Q245	-8	-8	0	-8	0
Q246	12	12	0	12	0
Q247	-11.906	-11.906	0	-12.336	3.49
Q248	-8	-8	0	8	0
Q249	12	12	0	12	0

Debido a que con muchos valores de desplazamientos para la comparativa se utilizaran únicamente los 3 valores máximos en positivos y negativos obtenidos en el análisis.

Resultados de desplazamientos en nodos se muestran en la Tabla 5.10.

Tabla 5.10 Desplazamientos en nodos, caso de validación tres.

<b>Desplazamiento</b>	<b>Código Motor [mm]</b>	<b>Ram Elements [mm]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [mm]</b>	<b>Diferencia [%]</b>
D1	-3.8495	-3.8504	0.02	-3.165	21.63
D2	-3.7453	-3.7462	0.024	-3.155	18.71
D3	-3.4946	-3.4954	0.023	-3.07	13.83
D4	.3951	.39515	0.013	.429	7.90
D5	.4185	.4186	0.024	.455	8.02
D6	.4341	.4343	0.046	.461	5.84



Resultados de fuerzas internas en elementos se muestran en la Tabla 5.11.

Tabla 5.11 Fuerzas internas en elementos, caso de validación tres.

<b>Fuerzas</b>	<b>Código Motor [kN]</b>	<b>Ram Elements [kN]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kN]</b>	<b>Diferencia [%]</b>
q1	-9.582	-9.582	0	-9.20	4.15
q2	-8	-8	0	-8.69	7.94
q3	-7	-7	0	-7.05	0.71
q4	4.577	4.577	0	4.07	12.46
q5	4.611	4.611	0	4.28	7.73
q6	5.21	5.21	0	5.353	2.67

Resultados de esfuerzos en elementos se muestran en la Tabla 5.12

Tabla 5.12 Fuerzas esfuerzos en elementos, caso de validación tres.

<b>Esfuerzos</b>	<b>Código Motor [kPa]</b>	<b>Ram Elements [kPa]</b>	<b>Diferencia [%]</b>	<b>SAP 2000 [kPa]</b>	<b>Diferencia [%]</b>
$\sigma_1$	-33647.6	-33647.26	0.001	-32269.29	4.27 %
$\sigma_2$	-28092.5	-28092.24	0.001	-28248.19	0.55 %
$\sigma_3$	-24580.9	-24580.71	0.001	-24726.46	0.59 %
$\sigma_4$	16071.8	16071.62	0.001	15781.1	1.84 %
$\sigma_5$	16193	16192.88	0.001	16000.13	1.21 %
$\sigma_6$	18299.9	18299.79	0.001	18781.33	2.56 %

En la comparativa de resultados se puede ver que tanto en reacciones como fuerzas la diferencia es mínima, la diferencia porcentual para el caso de deformaciones aparentemente es mucho mayor entre los resultados obtenidos en el SAP y el software desarrollado, sin embargo al estar hablando en el orden de los milímetros no representa gran cambio dicha diferencia. Para los esfuerzos la diferencia es menor a 5 % por lo que no representa gran influencia, este aumento

de diferencia puede deberse a los decimales contemplados en el área del perfil para nuestro software.





## Capítulo 5. Conclusiones y trabajos futuros

### Conclusiones.

Se ha desarrollado un software de análisis de armaduras espaciales, donde se pueden incluir armaduras en 2D, que tiene un enfoque educativo debido a que al realizar el análisis va mostrando las diferentes etapas del proceso que ayudan al usuario a visualizar las diferentes matrices y resultados parciales con la finalidad de entender conceptos vistos a lo largo de la teoría.

Se concluye que Scilab es una herramienta adecuada para el desarrollo de una interfaz gráfica, que permite cumplir con el objetivo del análisis correcto de una armadura espacial, además de que es un software de uso libre y con la capacidad para resolver problemas de índole educativa.

Los resultados obtenidos con el software desarrollado presentan una diferencia máxima de 5%, lo cual se considera aceptable comparado con los software Ram Elements y SAP 2000 en versión estudiante, que son software normalmente utilizados en el ramo de la ingeniería y que tienen un renombre a nivel internacional.

El proceso de solución de una armadura espacial por el método de rigideces está contenido en el código motor desarrollado que sirvió de base para el desarrollo de la interfaz gráfica.

El software desarrollado aun no muestra la aceptación bajo criterios de diseño de los elementos, sin embargo, si es posible tomar los valores de esfuerzos en los elementos calculados y compararlos con los esfuerzos permisibles del material utilizado.

Dos tipos de programación han sido empleadas para el desarrollo de esta tesis, el código motor que resume el proceso de solución del sistema de ecuaciones que representa la estructura fue necesaria la programación línea por línea un conjunto



determinado de líneas representa una operación matemática, mientras que en la interfaz gráfica se aplicó la programación por evento donde cada evento representa un cambio o paso en el modelado de la estructura.

La tesis expuesta representa el desarrollo de un software gratuito de análisis de armaduras espaciales que va a contribuir a la mejora en la enseñanza de las clases de Estática, Resistencia de Materiales, Análisis Estructural, Diseño Estructural y Acero. Existen muchos software y seguirán apareciendo versiones sin embargo, esta versión desarrollada seguirá siendo útil porque está basada en un método que no sufrirá cambios con el paso del tiempo.

### **Trabajos futuros.**

Una vez concluido el desarrollo del código motor y su interfaz gráfica en primera versión se abre un abanico de trabajos futuros relacionados con el cálculo estructural.

Existen varios códigos para el diseño de un elemento sometido a carga axial, en el caso de las armaduras generalmente se usa el acero; uno de los códigos reconocidos a nivel mundial para este tipo de estructuras y material es el AISC (American Institute of Steel Construction), este procedimiento puede ser implementado en la etapa de post proceso.

Para favorecer la optimización en el proceso de elección de perfiles estructurales se puede manejar la relación  $\frac{\text{solicitud}}{\text{resistencia}} \leq 1$ , siendo el valor más cercano a uno el perfil óptimo para el diseño.

Es muy común que en las estructuras de obra civil además de tener elementos que trabajan a carga axial se tengan elementos que trabajan a flexión, cortante y torsión (vigas). Para enriquecer el uso de nuestro software se requiere que se implemente a través de una nueva tesis el desarrollo para el análisis de vigas pues debe establecerse un nuevo código motor, ya que estos elementos presentan



diferentes tipos de carga tanto en nodos como en elementos y sus grados de libertad son diferentes logrando así ampliar la interfaz realizada.

Se puede utilizar esta metodología de elementos finitos para aplicarlo en lugar de mecánica de sólidos a las otras disciplinas del método como son: transferencia de calor, fluidos acústicos, flujo potencial, flujos generales, electrostática o magnetostática y desarrollar su interfaz correspondiente.

Esta interfaz puede ser traducida al inglés para tener un mayor alcance.



## Referencias

[1] Navarro Ugena, A., Pérez Castellanos, J. L., Open Course Ware, Universidad Carlos III de Madrid, España, recuperado de: [http://ocw.uc3m.es/mecanica-de-medios-continuos-y-teoria-de-estructuras/ingenieria-estructural/material-de-clase-1/apuntes/Capitulo\\_1\\_I\\_.Introduccion\\_a\\_las\\_estructuras.pdf](http://ocw.uc3m.es/mecanica-de-medios-continuos-y-teoria-de-estructuras/ingenieria-estructural/material-de-clase-1/apuntes/Capitulo_1_I_.Introduccion_a_las_estructuras.pdf).

[2] Aslam, Kassimali. 2015. Análisis Estructural. 5a Edición. Cengage Learning.

[3] Jaramillo S., H. E., Un Análisis de la Resistencia de Materiales a partir de los Postulados de Consideraciones y Demostraciones Matemáticas sobre dos Nuevas Ciencias de Galileo Galilei. Lámpsakos, ISSN: 2145-4086, No. 5, pp. 53-59. Ene-Jun. 2011.

[4] Aslam, Kassimali. 2010. Matrix Analysis of Structures. 2a Edición. Cengage Learning.

[5] Mio, Eduardo, Universidad Nacional Pedro Ruiz Gallo, 2019 recuperado de: <https://www.studocu.com/pe/document/universidad-nacional-pedro-ruiz-gallo/fisica-2/290068274-armaduras-espaciales/8900566>.

[6] Cervera R. Miguel, Blanco D., Elena. Mecánica de Estructuras. 2014, CIMNE, Barcelona, España.

[7] Paico Gasco, Segundo A. El “método matricial” en la enseñanza del análisis estructural: ¡una realidad! UCV-HACER. Revista de Investigación y Cultura, vol. 2, núm. 1, enero-junio, 2013, pp. 151-160 Universidad César Vallejo Chiclayo, Perú.

[8] Hibbeler, R. C., Análisis Estructural, 2012, 8va. Edición, Pearson Prentice Hall.

[9] Madrid Pérez, M. A. Apuntes de Análisis Estructural II. UACH<sup>3</sup>.



[10] Aranibar Zárate, J. A, Flores Mejía, R. F. Apoyo didáctico para la enseñanza y aprendizaje de la asignatura de estructuras hiperestáticas CIV 205, Octubre 2005, Bolivia.

[11] Tutorial Ram Elements V8i, Ejemplo 1, Estructura de Acero.

[12] Introducción a SCILAB, Programa de Cálculo Numérico, Universidad de Asunción, Facultad de Ingeniería, Febrero-Marzo 2005.



## Apéndice

### Código Motor

```
clc
clear
//Este es un programa que resuelve una armadura por el método de la rigidez en 3D
disp("Este es un programa que resuelve una armadura por el método de la rigidez en 3D.")
//Lectura e impresión de las coordenadas desde archivo.xls
hojacalculo=readxls('C:\Users\mjrc3586\Desktop\interfaz MRB3D septiembre 2022\MRB3D.xls')
pestananodos=hojacalculo(1)
NUMERONODOval=pestananodos.value(:,1)
NN=max(NUMERONODOval)
for i=1:NN
    NUMERONODO(i,1)=NUMERONODOval(i+1,1)
end
coorXval=pestananodos.value(:,2)
for i=1:NN
    coorX(i,1)=coorXval(i+1,1)
end
coorYval=pestananodos.value(:,3)
for i=1:NN
    coorY(i,1)=coorYval(i+1,1)
end
coorZval=pestananodos.value(:,4)
for i=1:NN
    coorZ(i,1)=coorZval(i+1,1)
end
disp('Coordenadas de la estructura')
coordenadas=[NUMERONODO coorX coorY coorZ]
disp('Nodo X Y Z')
disp(coordenadas)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Gráfica de la estructura
//Nodos
scatter3d(coorX,coorY,coorZ,36,"fill",".");
minX=min(coorX);
maxX=max(coorX);
Largoex=maxX-minX;
Xmin=minX-Largoex/8;
Xmax=maxX+Largoex/8;
minY=min(coorY);
maxY=max(coorY);
Largoey=maxY-minY;
Ymin=minY-Largoey/8;
Ymax=maxY+Largoey/8;
minZ=min(coorZ);
maxZ=max(coorZ);
Largoenz=maxZ-minZ;
Zmin=minZ-Largoenz/8;
Zmax=maxZ+Largoenz/8;
set(gca(),"data_bounds",matrix([Xmin,Xmax,Ymin,Ymax,Zmin,Zmax],2,-1));
title('Estructura 3D')
xlabel('X (u)')
ylabel('Y (u)')
zlabel('Z (u)')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Lectura de elementos
pestanaelementos=hojacalculo(2)
NUMERODEELEMENTOval=pestanaelementos.value(:,1)
NE=max(NUMERODEELEMENTOval)
for i=1:NE
    NUMERODEELEMENTO(i,1)=NUMERODEELEMENTOval(i+1,1)
end
NIval=pestanaelementos.value(:,2)
for i=1:NE
    NI(i,1)=NIval(i+1,1)
end
```



```
end
NFval=pestaanelementos.value(:,3)
for i=1:NE
    NF(i,1)=NFval(i+1,1)
end
Aval=pestaanelementos.value(:,4)
for i=1:NE
    A(i,1)=Aval(i+1,1)
end
Eval=pestaanelementos.value(:,5)
for i=1:NE
    E(i,1)=Eval(i+1,1)
end
disp('Elementos de la estructura')
ELEMENTOS=[NUMERODEELEMENTO NI NF A E]
disp('Elemento NI NF A E')
disp(ELEMENTOS)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Gráfica de los elementos
for i=1:NE
    XE=[coorX(NI(i,1),1) coorX(NF(i,1),1)];
    YE=[coorY(NI(i,1),1) coorY(NF(i,1),1)];
    ZE=[coorZ(NI(i,1),1) coorZ(NF(i,1),1)];
    param3d1(XE,YE,ZE)
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Cálculo de matrices de rigidez
L=zeros(NE,1);
LX=zeros(NE,1);
LY=zeros(NE,1);
LZ=zeros(NE,1);
NGL=3*NN;
KACUM=zeros(NGL,NGL);
for i=1:NE
    K=zeros(NGL,NGL);
    L(i,1)=sqrt((coorX(NF(i,1),1)-coorX(NI(i,1),1))^2 + (coorY(NF(i,1),1)-coorY(NI(i,1),1))^2 + (coorZ(NF(i,1),1)-coorZ(NI(i,1),1))^2);
    LX(i,1)=(coorX(NF(i,1),1)-coorX(NI(i,1),1))/L(i,1);
    LY(i,1)=(coorY(NF(i,1),1)-coorY(NI(i,1),1))/L(i,1);
    LZ(i,1)=(coorZ(NF(i,1),1)-coorZ(NI(i,1),1))/L(i,1);
    KKK=A(i,1)*E(i,1)/L(i,1)*[1 -1; -1 1];
    //eval(['KKK' num2str(i) ' = KKK;'])
    T=[LX(i,1) LY(i,1) LZ(i,1) 0 0 0; 0 0 0 LX(i,1) LY(i,1) LZ(i,1)];
    //eval(['T' num2str(i) ' = T;'])
    KK=T*KKK*T;
    //eval(['KK' num2str(i) ' = KK'])
    //Ensamblaje de la matriz de rigidez elemental en coordenadas globales
    //Fila 1 matriz de rigidez elemental
    K(3*NI(i,1)-2,3*NI(i,1)-2)=KK(1,1);
    K(3*NI(i,1)-2,3*NI(i,1)-1)=KK(1,2);
    K(3*NI(i,1)-2,3*NI(i,1))=KK(1,3);
    K(3*NI(i,1)-2,3*NF(i,1)-2)=KK(1,4);
    K(3*NI(i,1)-2,3*NF(i,1)-1)=KK(1,5);
    K(3*NI(i,1)-2,3*NF(i,1))=KK(1,6);
    //Fila 2 matriz de rigidez elemental
    K(3*NI(i,1)-1,3*NI(i,1)-2)=KK(2,1);
    K(3*NI(i,1)-1,3*NI(i,1)-1)=KK(2,2);
    K(3*NI(i,1)-1,3*NI(i,1))=KK(2,3);
    K(3*NI(i,1)-1,3*NF(i,1)-2)=KK(2,4);
    K(3*NI(i,1)-1,3*NF(i,1)-1)=KK(2,5);
    K(3*NI(i,1)-1,3*NF(i,1))=KK(2,6);
    //Fila 3 matriz de rigidez elemental
    K(3*NI(i,1),3*NI(i,1)-2)=KK(3,1);
    K(3*NI(i,1),3*NI(i,1)-1)=KK(3,2);
    K(3*NI(i,1),3*NI(i,1))=KK(3,3);
    K(3*NI(i,1),3*NF(i,1)-2)=KK(3,4);
    K(3*NI(i,1),3*NF(i,1)-1)=KK(3,5);
    K(3*NI(i,1),3*NF(i,1))=KK(3,6);
```



```

//Fila 4 matriz de rigidez elemental
K(3*Nf(i,1)-2,3*Ni(i,1)-2)=KK(4,1);
K(3*Nf(i,1)-2,3*Ni(i,1)-1)=KK(4,2);
K(3*Nf(i,1)-2,3*Ni(i,1))=KK(4,3);
K(3*Nf(i,1)-2,3*Nf(i,1)-2)=KK(4,4);
K(3*Nf(i,1)-2,3*Nf(i,1)-1)=KK(4,5);
K(3*Nf(i,1)-2,3*Nf(i,1))=KK(4,6);
//Fila 5 matriz de rigidez elemental
K(3*Nf(i,1)-1,3*Ni(i,1)-2)=KK(5,1);
K(3*Nf(i,1)-1,3*Ni(i,1)-1)=KK(5,2);
K(3*Nf(i,1)-1,3*Ni(i,1))=KK(5,3);
K(3*Nf(i,1)-1,3*Nf(i,1)-2)=KK(5,4);
K(3*Nf(i,1)-1,3*Nf(i,1)-1)=KK(5,5);
K(3*Nf(i,1)-1,3*Nf(i,1))=KK(5,6);
//Fila 6 matriz de rigidez elemental
K(3*Nf(i,1),3*Ni(i,1)-2)=KK(6,1);
K(3*Nf(i,1),3*Ni(i,1)-1)=KK(6,2);
K(3*Nf(i,1),3*Ni(i,1))=KK(6,3);
K(3*Nf(i,1),3*Nf(i,1)-2)=KK(6,4);
K(3*Nf(i,1),3*Nf(i,1)-1)=KK(6,5);
K(3*Nf(i,1),3*Nf(i,1))=KK(6,6);
//eval(['K' num2str(i) ' = K;'])
KACUM=K+KACUM;
//Escritura en pantalla de cálculos
mprintf('Elemento %g\n',i)
mprintf(' L = %g\n',L(i,1))
mprintf(' LX = %g\n',LX(i,1))
mprintf(' LY = %g\n',LY(i,1))
mprintf(' LZ = %g\n',LZ(i,1))
mprintf(' Matriz de rigidez de elemento %g\n',i)
disp(KK)
//mprintf(' Matriz de rigidez ensamblada de elemento %g\n',i)
//disp(K)
end
mprintf('\nMatriz de Rigidez global de la estructura.\n\nK=\n\n')
// Matriz global de rigidez
K=KACUM;
disp(K)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Lectura de grados de libertad restringidos
TXval=pestananodos.value(:,5)
for i=1:NN
    TX(i,1)=TXval(i+1,1)
end
TYval=pestananodos.value(:,6)
for i=1:NN
    TY(i,1)=TYval(i+1,1)
end
TZval=pestananodos.value(:,7)
for i=1:NN
    TZ(i,1)=TZval(i+1,1)
end
mprintf('\nRestricciones de la estructura (1=restricción, 0=libre).\n\n')
disp('Nodo TX TY TZ')
nodosrestringidos=[NUMERONODO, TX, TY, TZ];
disp(nodosrestringidos)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Contador de la dimensión de NGLR
NGLR=0;
for i=1:NN
    if TX(i,1) == 1
        NGLR=NGLR+1;
    end
    if TY(i,1) == 1
        NGLR=NGLR+1;
    end
    if TZ(i,1) == 1
        NGLR=NGLR+1;
    end
end

```







```
//Lectura de grados de libertad restringidos
DIXval=pestananodos.value(:,8)
for i=1:NN
    DIX(i,1)=DIXval(i+1,1)
end
DIYval=pestananodos.value(:,9)
for i=1:NN
    DIY(i,1)=DIYval(i+1,1)
end
DIZval=pestananodos.value(:,10)
for i=1:NN
    DIZ(i,1)=DIZval(i+1,1)
end
//Cálculo de los Desplazamientos Konocidos DK
DK=zeros(NGLR,1);
n=1;
for i=1:NN
    if TX(i,1) == 1
        DK(n,1)=DIX(i,1);
        n=n+1;
    end
    if TY(i,1) == 1
        DK(n,1)=DIY(i,1);
        n=n+1;
    end
    if TZ(i,1) == 1
        DK(n,1)=DIZ(i,1);
        n=n+1;
    end
end
mprintf("\nLos Desplazamientos Konocidos en la estructura DK son\n')
disp(DK)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Lectura de cargas prescritas en los nodos
FXval=pestananodos.value(:,11)
for i=1:NN
    FX(i,1)=FXval(i+1,1)
end
FYval=pestananodos.value(:,12)
for i=1:NN
    FY(i,1)=FYval(i+1,1)
end
FZval=pestananodos.value(:,13)
for i=1:NN
    FZ(i,1)=FZval(i+1,1)
end
FK=zeros(NGLNR,1);
n=1;
for i=1:NN
    if TX(i,1) == 0
        FK(n,1)=FX(i,1);
        n=n+1;
    end
    if TY(i,1) == 0
        FK(n,1)=FY(i,1);
        n=n+1;
    end
    if TZ(i,1) == 0
        FK(n,1)=FZ(i,1);
        n=n+1;
    end
end
mprintf("\nLas fuerzas Konocidas en la estructura FK son\n')
disp(FK)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//PROCESO
//Vector de desplazamientos desconocidos (Ahora conocidos)
DU=K1DU^-1*(FK-K1DK*DK);
```



```
mprintf("\nLos desplazamientos desconocidos de la estructura son\n')
disp(DU)
//Se obtiene el vector de fuerzas desconocidas de la estructura
FU=K2DU*DU+K2DK*DK;
mprintf("\nLas fuerzas desconocidas de la estructura (reacciones de apoyos) son\n')
disp(FU)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D=zeros(NGL,1);
//Vector desplazamientos globales de la estructura
for i=1:NGL
    for n=1:NGLR
        if i==VNGLR(n,1)
            D(i,1)=DK(n,1);
        end
    end
    for n=1:NGLNR
        if i==VNGLNR(n,1)
            D(i,1)=DU(n,1);
        end
    end
end
mprintf("\nEl vector de desplazamientos globales de la estructura es\n')
for i=1:NGL
    mprintf('D%g = %g\n',i,D(i,1))
end
F=zeros(NGL,1);
//Vector fuerzas globales de la estructura
for i=1:NGL
    for n=1:NGLR
        if i==VNGLR(n,1)
            F(i,1)=FU(n,1);
        end
    end
    for n=1:NGLNR
        if i==VNGLNR(n,1)
            F(i,1)=FK(n,1);
        end
    end
end
mprintf("\nEl vector de fuerzas globales de la estructura es\n')
for i=1:NGL
    mprintf('F%g = %g\n',i,F(i,1))
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Cálculo de fuerzas internas
mprintf("\nLas fuerzas internas en los elementos son:\n')
q=zeros(NE,1);
for i=1:NE
    q(i,1)=A(i,1)*E(i,1)/L(i,1)*[-LX(i,1) -LY(i,1) -LZ(i,1) LX(i,1) LY(i,1) LZ(i,1)]*[D(3*NI(i,1)-2,1) D(3*NI(i,1)-1,1)
D(3*NI(i,1),1) D(3*NF(i,1)-2,1) D(3*NF(i,1)-1,1) D(3*NF(i,1),1)];
    mprintf('q%g = %g\n',i,q(i,1))
end
mprintf("\nLos esfuerzos en los elementos son:\n')
S=zeros(NE,1);
for i=1:NE
    S(i,1)=E(i,1)/L(i,1)*[-LX(i,1) -LY(i,1) -LZ(i,1) LX(i,1) LY(i,1) LZ(i,1)]*[D(3*NI(i,1)-2,1) D(3*NI(i,1)-1,1) D(3*NI(i,1),1)
D(3*NF(i,1)-2,1) D(3*NF(i,1)-1,1) D(3*NF(i,1),1)];
    mprintf('S%g = %g\n',i,S(i,1))
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
//Coordenadas deformadas
//Factor de escalamiento
FEXval=pestananodos.value(:,14)
FEX=FEXval(2,1)
FEYval=pestananodos.value(:,15)
FEY=FEYval(2,1)
FEZval=pestananodos.value(:,16)
FEZ=FEZval(2,1)
```



```
coorXn=zeros(NN,1);
coorYn=zeros(NN,1);
coorZn=zeros(NN,1);
for i=1:NN
    coorXn(i,1)=coorX(i,1)+FEX*D(3*i-2,1);
    coorYn(i,1)=coorY(i,1)+FEY*D(3*i-1,1);
    coorZn(i,1)=coorZ(i,1)+FEZ*D(3*i,1);
end
scatter3d(coorXn,coorYn,coorZn,100,"fill","*");
gcf().color_map = coolcolormap(64);
//Elementos deformados
for i=1:NE
    XE=[coorXn(NI(i,1),1) coorXn(NF(i,1),1)];
    YE=[coorYn(NI(i,1),1) coorYn(NF(i,1),1)];
    ZE=[coorZn(NI(i,1),1) coorZn(NF(i,1),1)];
    param3d1(XE,YE,ZE,'r')
end
```