

Capítulo 1

Antecedentes

1.1.- El problema

El problema general en el análisis de placas planas, durante el curso de la carrera de ingeniería civil en nuestra superior casa de estudio, es que sólo se dispone de métodos empíricos aproximados para su análisis. Dichos métodos están desfasados con el avance de la tecnología, por lo que hoy en día disponemos de otras metodologías más precisas, como también las herramientas para llevar a cabo dicho análisis.

La metodología que modela mejor el comportamiento de una placa plana, es el método de los elementos finitos, pero el cálculo manual en sí mediante este método es muy engorroso, ya que conlleva el tratamiento de numerosas matrices de gran tamaño, por lo que es necesaria una herramienta computacional que automatice el proceso.

1.1.1.- Planteamiento del problema

El problema a resolver es la automatización de todo el proceso de cálculo involucrado en el análisis por el método de los elementos finitos, desde la formulación de las matrices de rigidez, hasta la formulación y resolución del sistema de ecuaciones resultante. Para lo cual se desarrolla una herramienta computacional capaz de realizar el análisis de placas planas sometidas a flexión, automatizando el proceso de generación de la malla de elementos, la generación de matrices de rigidez, el proceso de ensamblaje y de resolución del sistema de ecuaciones resultante. De manera más precisa y confiable, sin tener que recurrir a tablas de cálculo, que dan como resultado, valores aproximados y sobre dimensionados debido a la incertidumbre en su elaboración.

1.1.2.- Formulación del problema

Para resolver el problema mencionado anteriormente, se recurre a la teoría clásica de flexión de placas planas y se formula la solución a partir del método de los elementos finitos, obteniendo las matrices de rigidez a partir de las funciones de forma escogidas para el elemento.

El problema a resolver entonces, es determinar el número de nodos y el número de grados de libertad, que tendrá cada nodo del elemento, como también el tipo de función para la interpolación de deformaciones, ya que de esto depende la precisión del método.

1.1.3.- Sistematización del problema

Para la obtención de la matriz de rigidez del elemento de placa plana, se formulan las funciones de forma del elemento, tomando en cuenta que se trata de un elemento de “n” nodos y cada nodo cuenta con tres grados de libertad (una traslación vertical y dos rotaciones paralelas al plano de la placa), dando en total un elemento con $n \times n$ grados de libertad y por tanto su matriz de rigidez será de $n \times n$.

Una vez determinada la matriz de rigidez se procede a su programación en MatLab, de manera que su cálculo sea automático, a partir de las coordenadas del elemento y de las funciones de formas escogidas para la interpolación de esfuerzos y deformaciones.

Una vez formuladas las diferentes matrices de rigidez, se procede a fijar las condiciones de contorno o vínculos de la estructura que se desea analizar.

Posteriormente se presenta los resultados de los cálculos por pantalla junto con las gráficas que muestran la distribución de tensiones y la forma deformada de la placa.

Para la programación del método de los elementos finitos, se procede de la siguiente manera:

El entorno de desarrollo, es la plataforma de lenguaje “m” propia de MatLab, que es un lenguaje similar al lenguaje C++.

El programa se desarrolla mediante el método de la programación orientada a objetos, debido a su gran ventaja sobre la programación estructurada en cuanto a reutilización de código y mejor entendimiento del código fuente para futuras investigaciones y mejoras.

La estructuración de clases de la cuales está compuesto el programa es la siguiente:

- **Clase Elemento:** En esta clase se fijan los atributos y métodos para el manejo y proceso de datos de los diferentes elementos finitos formulados. En esta clase se incluye un método para el cálculo de las matrices de rigidez a partir de sus coordenadas de ubicación del elemento, las cuales son atributos de clase, como también del tipo de material del que está compuesto la placa.
- **Clase Carga:** En esta clase se gestiona los datos y métodos para la evaluación de las diferentes situaciones de carga a los que puede estar sometida una placa plana (losa). En el caso de losas planas se estudia dos tipos de carga: La carga puntual y la carga uniformemente distribuida, por lo que esta clase será una clase madre para las clases de carga antes mencionadas.
- **Clase Vínculo:** En esta clase se gestiona la información necesaria para controlar las restricciones de desplazamiento de los nodos que representan los apoyos de la estructura. Esta clase cuenta con un vector de desplazamientos en cuyas celdas está representado por “1” el grado de libertad desplazable y por “0” el desplazamiento restringido.

1.2.- Objetivos

1.2.1.- Objetivo general

El objetivo general del presente trabajo de investigación, es el desarrollo de un programa interactivo de análisis de placas planas a código abierto, para el apoyo del diseño de losas y el aprendizaje del método de los elementos finitos.

1.2.2.- Objetivos específicos

Los objetivos específicos del presente trabajo son los siguientes:

- Realizar el análisis de placas planas mediante modelación por el método de los elementos finitos.
- Realizar el trazado de los diagramas de fuerzas internas resultantes en la placa (momentos flectores, fuerzas cortantes), para realizar el diseño posteriormente.
- Mostrar de manera gráfica la distribución de tensiones resultantes en la placa, debido al estado de carga.
- Mostrar de manera gráfica la deformación de la placa, debido a las cargas aplicadas.
- Fomentar e incentivar al estudiante, en la investigación más a fondo del método de los elementos finitos y el desarrollo de sus propias herramientas computacionales de análisis.

1.3.- Justificación

La justificación del presente trabajo se da en sus dos formas: justificación académica y justificación técnica.

1.3.1.- Justificación académica

El presente trabajo de investigación, busca mediante la aplicación del lenguaje de programación “m” de MatLab:

- Enriquecer la base conceptual de programación y a la vez enriquecer el criterio de análisis estructural mediante el método de los elementos finitos.
- Incentivar al estudiante a estudiar más a fondo el método de los elementos finitos y sus aplicaciones en los diferentes campos de la ingeniería civil, además del campo del análisis estructural.
- Facilitar al docente la explicación del método de los elementos finitos aplicado al análisis de placas planas, mostrando los cálculos involucrados durante el proceso de solución de los problemas y ejemplos de aplicación.

1.3.2.- Justificación técnica

Para comprender mejor el análisis matricial de estructuras, como su formulación a partir de elementos finitos. Es necesario un programa que automatice el proceso de cálculo y muestre resultados conforme se vaya desarrollando el proceso de cálculo, ya que este procedimiento se vuelve tedioso cuando la estructura es muy compleja o está formada por demasiados elementos.

Por lo tanto en el presente trabajo de investigación, se desarrolla una herramienta computacional que automatice dichos procesos, con el objetivo de mejorar el aprendizaje y la experiencia del usuario en el campo de los elementos finitos, como a la vez de brindar al usuario una herramienta de análisis precisa.

1.4.- Marco conceptual

1.4.1.- Marco espacial

La presente investigación se desarrolla en la ciudad de Tarija y puede ser aplicada a nivel departamental por los docentes de la materia de elementos finitos, como también estudiantes de la carrera de ingeniería civil, como también las demás carreras de ingeniería (por ejemplo ingeniería industrial) donde sea necesario el análisis de placas planas a flexión, dentro de las diferentes instalaciones de la universidad autónoma Juan Misael Saracho.

1.4.2.- Marco temporal

El presente trabajo se desarrolla en el año 2019. El cual contempla el análisis de placas planas por el método matricial de los elementos finitos.

Se programa el método de manera que se automatice el proceso, en el lenguaje de programación “m” propio de MatLab.

1.5.- Alcance de la investigación

En la presente investigación se desarrolla un software de análisis matricial por el método de los elementos finitos aplicado al análisis de placas planas sometidas a flexión. En el cual se muestra de manera secuencial y gráfica el proceso de formulación y cálculo de las matrices de rigidez de los diferentes elementos finitos, como también el proceso de cálculo de los vectores de carga aplicados a los nodos de la estructura discretizada y el proceso de ensamblaje de la matriz de rigidez de la estructura completa, para terminar con la formación de la matriz de rigidez reducida de la cual se encontrará el vector de deformaciones nodales, a partir de los cuales se obtiene la solución al problema, tanto para reacciones como para fuerzas internas. Para finalmente mostrar de manera gráfica la distribución de tensiones y la gráfica de la placa deformada después de ser aplicadas las cargas.

1.5.1.- Hipótesis de trabajo

Las hipótesis del presente trabajo de investigación para simplificar el problema a resolver son:

- El resultado del trabajo propuesto dará como resultado un programa de fácil manejo y comprensión de resultados obtenidos por el programa.
- El usuario podrá ver más de cerca el proceso de formulación de los elementos finitos aplicados en la resolución de placas planas a flexión.
- El usuario entenderá mejor el concepto de función de interpolación o función de forma, al poder visualizar dichas funciones de forma gráfica.

Capítulo 2

Marco Teórico

2.1.- Fundamentos del método de los elementos finitos (MEF)

Debido a la complejidad de las ecuaciones de gobierno de las estructuras continuas y en general, la mayoría de las estructuras se hace imprescindible la aplicación del *método de los elementos finitos o MEF*. Este método determina el comportamiento de una estructura ante las cargas sustituyendo la solución continua, exacta y en la mayoría de los casos imposible del sistema de ecuaciones diferenciales que conforman el problema elástico por una solución discontinua o discreta, por lo tanto aproximada.

Para ello se discretiza la estructura, dividiéndola en elementos no diferenciales o elementos finitos interconectados entre sí, a través de un determinado número de puntos que llamaremos nodos.

Después de estudiar cada elemento por separado se recompone la estructura, restableciendo el equilibrio y la compatibilidad de desplazamientos en los nodos, lo que da lugar a un sistema de ecuaciones algebraicas. La solución de este sistema de ecuaciones permite hallar los desplazamientos de los nodos y a partir de ellos, las restantes incógnitas de la estructura.

Es un método aproximado cuyo grado de aproximación aumenta con el número de elementos en que se divide la estructura, cuando la aplicación del método es correcta.

2.2. Descripción del método de elementos finitos (MEF)

2.2.1.- Fases del método

El método de los elementos finitos consta de las siguientes fases:

1. División en elementos finitos
2. Vector de desplazamientos del elemento
3. Matriz de rigidez del elemento
4. Matriz completa de rigidez de la estructura
5. Respuesta de la estructura

La fase 1 de división en elementos finitos, se realiza utilizando programas de ordenador llamados preprocesadores.

Las fases 2 y 3 de vector de desplazamientos y matriz de rigidez del elemento, constituyen lo que se denomina formulación del elemento, a partir de la cual se llega a la fase 4 de matriz completa de rigidez de la estructura.

Finalmente en la fase 5 de respuesta de la estructura, se determinan los parámetros nodales que permiten calcular la respuesta de los elementos, es decir esfuerzos y deformaciones. En esta fase es preciso utilizar programas de ordenador llamados postprocesadores.

2.2.2.- División en elementos finitos

Esta fase del MEF divide una estructura continua en elementos finitos interconectados entre sí mediante nodos que están situados generalmente en los bordes de los elementos per que pueden estar también es su interior. De esta forma los infinitos grados de libertad de una estructura continua se convierten en un número finito de grados de libertad de la estructura discreta, representado por el número total de parámetros nodales, que comprenden los desplazamientos de los nodos y en muchos casos también sus derivadas.

Es evidente que cuanto mayor sea el número de elementos finitos en que se divide una estructura habrá más similitud entre la estructura discretizada y la estructura continua, en consecuencia, será mayor el grado de aproximación de los resultados que se obtengan. Según el tipo de estructura que se considere los elementos finitos serán unidimensionales, bidimensionales, o tridimensionales.

2.2.2.1.- Elementos unidimensionales

Las estructuras cuyo comportamiento o deformación ante las cargas depende de una sola variable, se discretizan en elementos unidimensionales. En el caso más simple de una viga de sección constante sometida a una carga de tracción P (Imagen 1), utilizaremos un elemento finito de la misma longitud y sección que la viga con nodos 1 y 2 en sus extremos, siendo los parámetros nodales u_1 y u_2 .

Imagen 1: Elemento lineal de dos nodos



2.2.2.2.- Elementos bidimensionales

Cuando el comportamiento de la estructura ante las cargas depende de dos coordenadas, por ejemplo x e y , la estructura se discretiza en elementos bidimensionales. Esta discretización en elementos bidimensionales se aplica a estructuras con estado de esfuerzos plano, como las placas con cargas en su plano, o a estructuras con estado de deformaciones plano como los muros de contención.

Uno de los elementos bidimensionales más utilizados en la discretización de estructuras es el elemento triangular por su fácil adaptación a cualquier superficie plana (Imagen 2), Otros elementos bidimensionales, son el elemento rectangular (Imagen 3), que a su vez pueden subdividirse en elementos triangulares. Los elementos curvados, como el elementos triangular curvado, se caracterizan por su fácil adaptación a las zonas perimetrales de las superficies.

Imagen 2: Elemento bidimensional de tres nodos

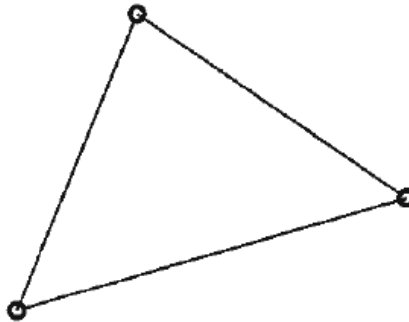
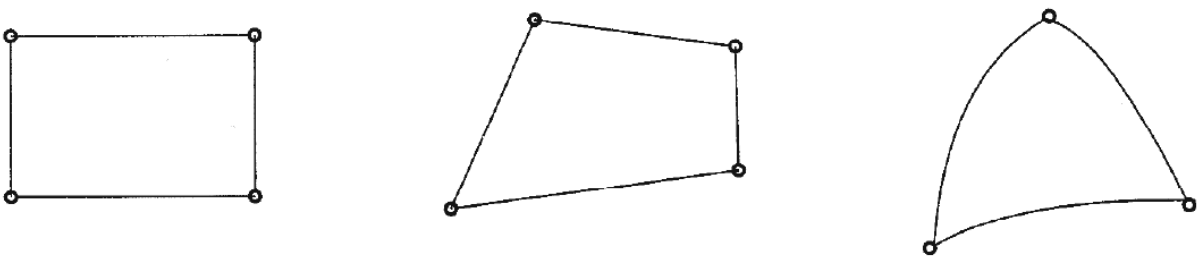


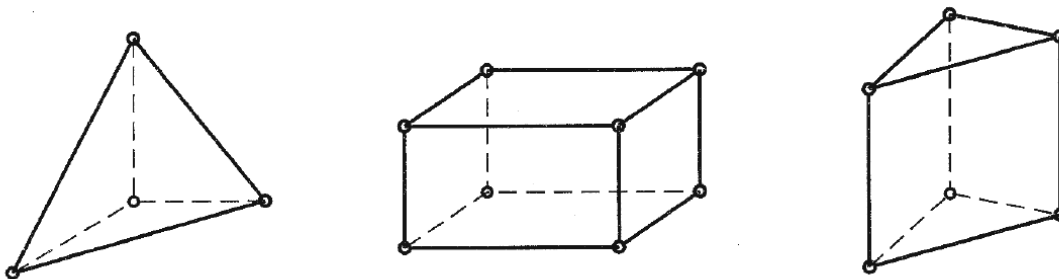
Imagen 3: Elemento bidimensional de cuatro nodos y elemento triangular curvo



2.2.2.3.- Elementos tridimensionales

En muchas ingenierías, se utilizan estructuras espaciales tanto en su forma como en las cargas aplicadas. Estas estructuras exigen una discretización tridimensional, siendo los elementos tridimensionales de uso más frecuente los tetraédricos de cuatro nodos (Imagen 4), estos elementos se consideran equivalentes a los elementos triangulares bidimensionales. Otros elementos tridimensionales muy utilizados son los hexaédricos y los prismáticos (Imagen 4).

Imagen 4: Elementos tridimensionales de 4 nodos, 8 nodos, 6 nodos



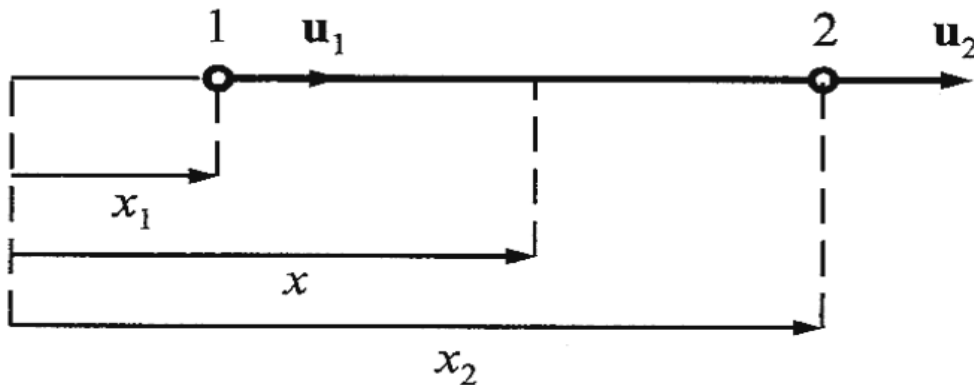
2.2.3.- Vector de desplazamientos del elemento

En esta fase del MEF se determina el vector de desplazamientos del elemento que permite conocer de forma aproximada los desplazamientos en cualquier punto del elemento.

2.2.3.1.- Funciones unidimensionales

Tomamos como ejemplo un elemento finito unidimensional de longitud L , con nodos en sus extremos (Imagen 5), siendo los desplazamientos u_1 y u_2 de sus nodos 1 y 2 los únicos parámetros nodales, los grados de libertad del elemento son 2. El desplazamiento de un punto cualquiera del elemento puede expresarse aproximadamente utilizando la función de desplazamientos:

Imagen 5: Elemento lineal de dos grados de libertad en coordenadas X



$$u(x) = \alpha_1 + \alpha_2 x \quad (2.1)$$

Que es un polinomio con un número de parámetros α_i igual a 2, que es el grado de libertad del elemento. Estos parámetros α_i se obtienen particularizando la expresión anterior en los nodos:

$$u(x_1) = \alpha_1 + \alpha_2 x_1 \quad (2.2)$$

$$u(x_2) = \alpha_1 + \alpha_2 x_2 \quad (2.3)$$

Dando como resultado:

$$\alpha_1 = \frac{x_2 u_1 - x_1 u_2}{L} \quad y \quad \alpha_2 = \frac{u_2 - u_1}{L}$$

Al sustituir estos valores en (2.1), se obtiene:

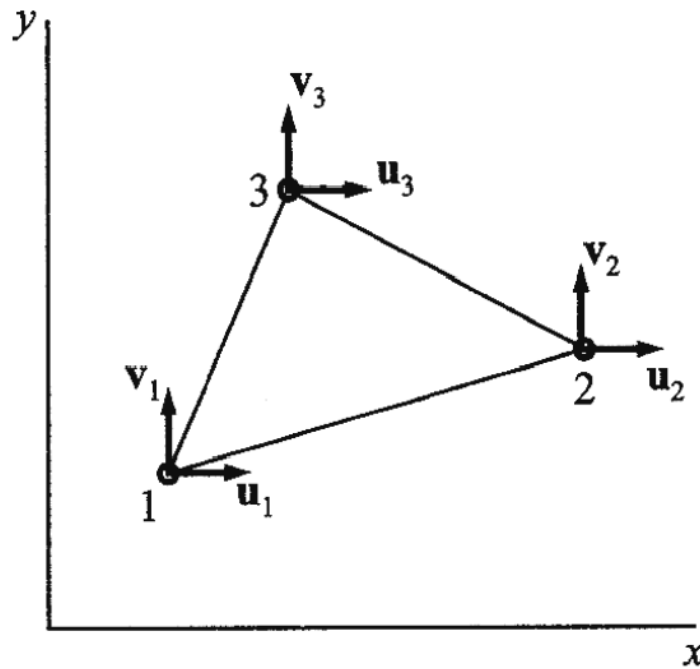
$$u(x) = N_1 u_1 + N_2 u_2 \quad (2.4)$$

Siendo N_1 y N_2 las funciones de interpolación del elemento.

2.2.3.2.- Funciones bidimensionales

Consideremos como elemento bidimensional un elemento triangular con un nodo en cada vértice y desplazamientos $u_1, v_1, u_2, v_2, u_3, v_3$ (Imagen 6).

Imagen 6: Elemento bidimensional con sus grados de libertad



En este caso el desplazamiento de un punto cualquiera del elemento puede ser expresado aproximadamente por las funciones de desplazamientos:

$$u(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \quad (2.5)$$

$$v(x,y) = \alpha_4 + \alpha_5 x + \alpha_6 y \quad (2.6)$$

Que son polinomios que tienen un número de parámetros α_i igual a 6, que es el grado de libertad del elemento. Particularizando las funciones anteriores en los nodos 1, 2 y 3, se determinan los parámetros α_i , que sustituidos en las ecuaciones (2.5) y (2.6), permiten expresar el vector de desplazamientos del elemento bidimensional en la forma:

$$u_e = N a_e \quad (2.7)$$

Siendo el vector de desplazamientos del elemento:

$$a_e = (u_1 \quad v_1 \quad u_2 \quad v_2 \quad u_3 \quad v_3)^T \quad (2.8)$$

Y la matriz de interpolación del elemento:

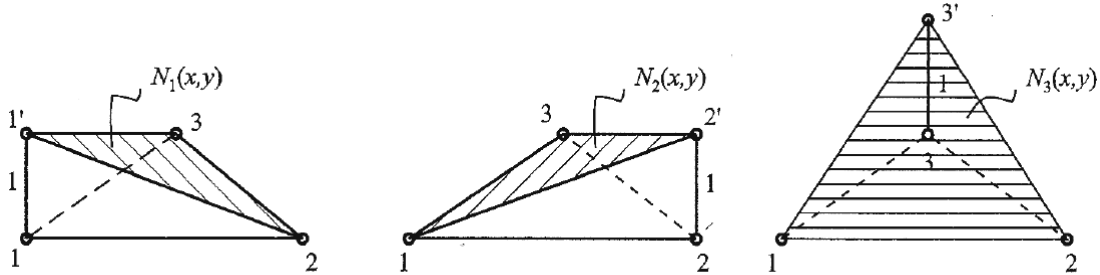
$$N_e = (N_1 \quad N_2 \quad N_3) \quad (2.9)$$

Los desplazamientos u tienen que depender solamente de u_1, u_2, u_3 . De la misma forma, los desplazamientos v dependerán solamente de v_1, v_2, v_3 y, además ambas dependencias tienen que ser iguales. Por esta razón la matriz de forma del elemento N_e ha de ser:

$$N_e = \begin{pmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{pmatrix} \quad (2.10)$$

Siendo las funciones de interpolación N_i polinomios de primer grado en x, y , cuyos coeficientes dependen de las coordenadas nodales. Al igual que en el caso del elemento unidimensional, la función de interpolación N_i determina los desplazamientos de los puntos del elemento cuando se le da un valor unidad al desplazamiento del nodo i , manteniendo nulos los desplazamientos de los otros nodos.

Imagen 7: Funciones de forma del elemento bidimensional



2.2.3.3.- Funciones de forma tridimensionales

En el caso de un elemento tridimensional con n_n nodos y cada nodo con 3 parámetros u_i, v_i, w_i (Imagen 8), el vector de desplazamientos del elemento es:

$$u_e = N_e a \quad (2.11)$$

Donde la matriz de forma del elemento N_e es:

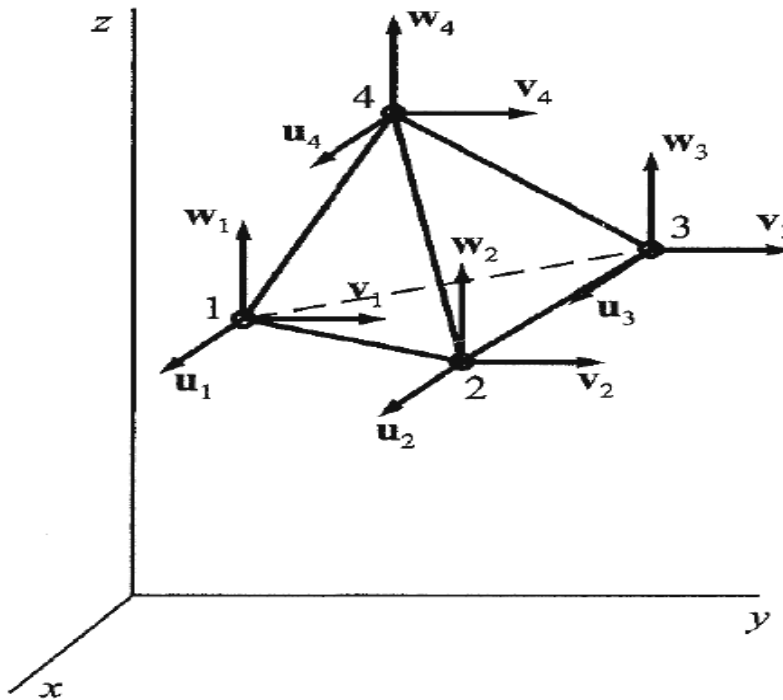
$$N_e = (N_1 \quad N_2 \quad N_3 \quad \dots \quad N_i) \quad (2.12)$$

Donde los elementos N_i son de la forma:

$$N_i = \begin{pmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{pmatrix} \quad (2.13)$$

En este caso, las funciones de interpolación N_i son polinomios de primer grado en x, y, z , cuyos coeficientes dependen de las coordenadas nodales.

Imagen 8: Elemento tridimensional con sus grados de libertad



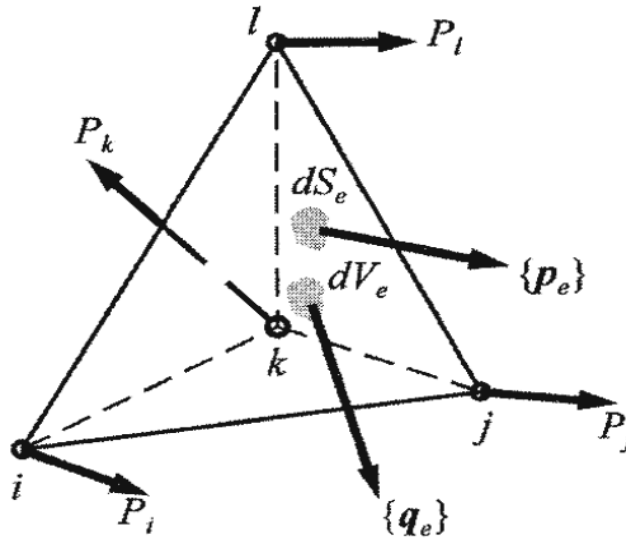
2.2.3.- Matriz de rigidez del elemento

Esta fase del método de los elementos finitos determina la matriz de rigidez del elemento que permite calcular los parámetros nodales de un elemento, en función de las fuerzas nodales que actúan sobre él. A continuación se formula la matriz de rigidez del elemento utilizando, primero el teorema de los trabajos virtuales.

2.2.3.1.- Teorema de los trabajos virtuales

Consideremos que una estructura en equilibrio se ha discretizado en elementos finitos. Cada elemento estará en equilibrio sometida a un sistema de fuerzas externas F_e que comprenden las fuerzas aplicadas en los nodos, las fuerzas q_e distribuidas en su volumen V_c y las fuerzas p_e distribuidas en su superficie S_e (Imagen 9).

Imagen 9: Elemento finito sometido a cargas nodales de volumen, superficie y cargas distribuidas



Estas fuerzas originan en cada punto del elemento un estado de esfuerzos definido por el vector de esfuerzos σ .

Supongamos que en ese elemento en equilibrio se originan unos desplazamientos nodales virtuales δa , que satisfacen las condiciones de contorno. Estos desplazamientos nodales dan lugar a unos desplazamientos virtuales δu en los puntos del elemento que originan cada punto un estado de deformaciones virtuales definido por el vector de deformaciones $\delta \epsilon$.

Según el teorema de los trabajos virtuales, ha de ser nula la suma del trabajo W_{ext} que realizan las fuerzas externas F_e durante los desplazamientos virtuales δa y δu del trabajo W_{int} que realizan las fuerzas internas, debidas a los esfuerzos σ , durante las deformaciones virtuales $\delta \epsilon$.

El trabajo que realizan las fuerzas externas F_e durante los desplazamientos virtuales δa y δu es:

$$W_{ext} = \delta a^T P_e + \int_V \delta u^T q_e dV + \int_S \delta u^T p dS \quad (2.14)$$

Según la ecuación (2.7), la relación entre los desplazamientos de los puntos del elemento u y los parámetros nodales a, es $u = N a$. Aplicando a esta relación la igualdad matricial $(AB)^T = B^T A^T$, resulta:

$$u^T = a^T N^T \quad (2.15)$$

Igualmente con el desplazamiento virtual:

$$\delta u^T = \delta a^T N^T \quad (2.16)$$

Sustituyendo esta relación en la ecuación (2.14), se obtiene:

$$W_{ext} = \delta a^T P_e + \int_V \delta a^T N^T q_e dV + \int_S \delta a^T N^T p_e dS \quad (2.17)$$

Es decir:

$$W_{ext} = \delta a^T \left(P_e + \int_V N^T q_e dV + \int_S N^T p_e dS \right) \quad (2.18)$$

Asimismo, el trabajo W_{int} que realizan las fuerzas internas, debidas a los esfuerzos σ , durante las deformaciones virtuales $\delta \varepsilon$ es:

$$W_{int} = \int_V \delta \varepsilon^T \sigma dV \quad (2.19)$$

Teniendo en cuenta que:

$$\varepsilon = B \delta a \quad (2.20)$$

Dónde:

$$B = \partial N \quad (2.21)$$

Teniendo en cuenta ahora la relación entre los esfuerzos y las deformaciones, se obtiene:

$$\sigma = DB \delta a \quad (2.22)$$

Aplicando de nuevo la relación $(AB)^T = B^T A^T$ a la ecuación (2.20), resulta:

$$\varepsilon^T = \delta a^T B^T \quad (2.23)$$

Sustituyendo en la ecuación (2.22) y (2.23) en el trabajo virtual interno se obtiene:

$$W_{int} = \int_V \delta \varepsilon^T \sigma dV = \int_V \delta a^T B^T DB \delta a dV = \delta a^T \left(\int_V B^T DB dV \right) \delta a \quad (2.24)$$

Igualando ahora el trabajo externo W_{ext} con el trabajo interno W_{int} se obtiene:

$$\left(\int_V B^T D B dV \right) \delta a = P_e + \int_V N^T q_e dV + \int_S N^T p_e dS \quad (2.25)$$

Que representa la ecuación matricial de equilibrio del elemento y equivale a:

$$F_e = K \delta a \quad (2.26)$$

Donde K es la matriz de rigidez del elemento definido por:

$$K = \int_V B^T D B dV \quad (2.27)$$

El vector F_e es el vector de fuerzas nodales del elemento, igual a:

$$F_e = F_p + F_q + F_p \quad (2.28)$$

Siendo:

$$F_p = P_e \quad (2.29)$$

El vector de fuerzas aplicadas directamente a los nodos del elemento,

$$F_q = \int_A N^T q_e dA \quad (2.30)$$

El vector de fuerzas nodales equivalente a las fuerzas distribuidas en el volumen del elemento y:

$$F_p = \int_S N^T p_e dS \quad (2.31)$$

El vector de fuerzas nodales equivalente a las fuerzas distribuidas en la superficie del elemento.

Todas las integrales que figuran en la matriz de rigidez del elemento y en los vectores de fuerzas, salvo en estructuras muy sencillas, se resuelven utilizando métodos de integración numérica, de los cuales el más generalizado es el de la cuadratura de Gauss.

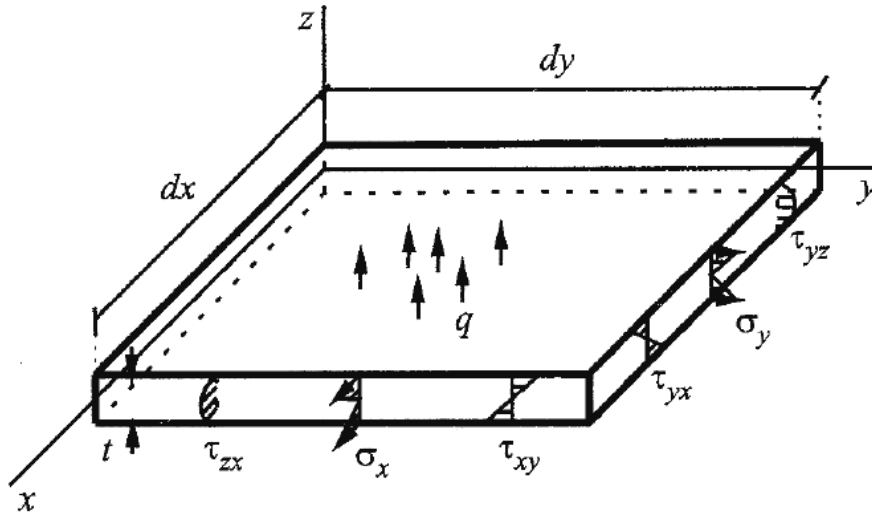
2.3.- Placas delgadas

Una estructura se considera una placa delgada cuando una de sus dimensiones (espesor) es mucho menor que otras dos. Esta placa trabajará a flexión cuando está sometida a un estado de cargas formado por fuerzas aplicadas normalmente a su superficie.

2.3.1.- Ecuación diferencial de gobierno

Consideremos una placa delgada sometida a flexión debida a una carga transversal. Aislado un elemento de placa rectangular de dimensiones $dx dy$ (Imagen 10), se elige un sistema de ejes en que los ejes (x,y) se hallan situados en la superficie media de la placa. De forma análoga a la flexión de vigas, se admite que los esfuerzos normales σ_x, σ_y varían linealmente con z mientras los esfuerzos normales σ_z se consideran despreciables. Así mismo se admite que los esfuerzos cortantes τ_{xy} varían linealmente con z , mientras los esfuerzos cortantes τ_{yz}, τ_{zx} varían de forma cuadrática.

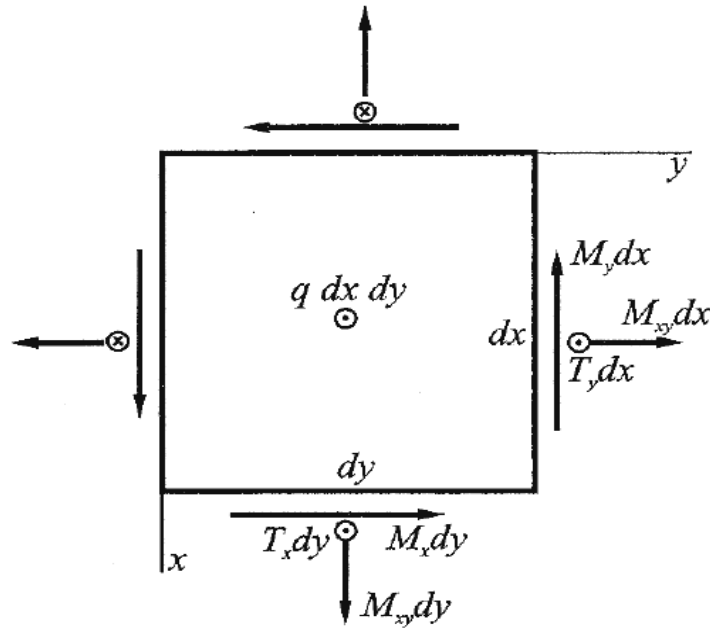
Imagen 10: Elemento de placa plana sometido a una carga distribuida



Las sollicitaciones por unidad de longitud que actúan sobre el elemento diferencial de placa de espesor t (Imagen 11), son los momentos flectores M_x, M_y , los momentos torsores M_{xy} y las fuerzas cortantes T_x, T_y , representadas todas ellas con los sentidos considerados positivos. Al establecer las relaciones existentes entre las sollicitaciones y los esfuerzos se obtiene:

$$\begin{aligned}
 M_x &= \int_{-t/2}^{t/2} \sigma_x z \, dz \\
 M_y &= \int_{-t/2}^{t/2} \sigma_y z \, dz \\
 M_{xy} &= \int_{-t/2}^{t/2} \tau_{xy} z \, dz \quad (2.32) \\
 T_x &= \int_{-t/2}^{t/2} \tau_{zx} \, dz \\
 T_y &= \int_{-t/2}^{t/2} \tau_{yz} \, dz
 \end{aligned}$$

Imagen 11: Elemento diferencial de placa



Siendo $q(x,y)$ la carga transversal de la placa. Del equilibrio de este elemento diferencial de placa, despreciando los infinitésimos de segundo orden, se obtiene:

$$\begin{aligned} \sum Z = 0, & \quad \frac{\partial T_x}{\partial x} + \frac{\partial T_y}{\partial y} + q_{(x,y)} = 0 \\ \sum M_x = 0, & \quad \frac{\partial M_y}{\partial y} + \frac{\partial M_{xy}}{\partial x} - T_y = 0 \\ \sum M_y = 0, & \quad \frac{\partial M_x}{\partial x} + \frac{\partial M_{xy}}{\partial y} - T_x = 0 \end{aligned} \quad (2.33)$$

Sustituyendo en la primera de estas ecuaciones, la derivada de la segunda respecto a “y”, y la derivada de la tercera respecto a “x”, se obtiene:

$$\frac{\partial^2 M_x}{\partial x^2} + 2 \frac{\partial^2 M_{xy}}{\partial x \partial y} + \frac{\partial^2 M_y}{\partial y^2} = -q_{(x,y)} \quad (2.34)$$

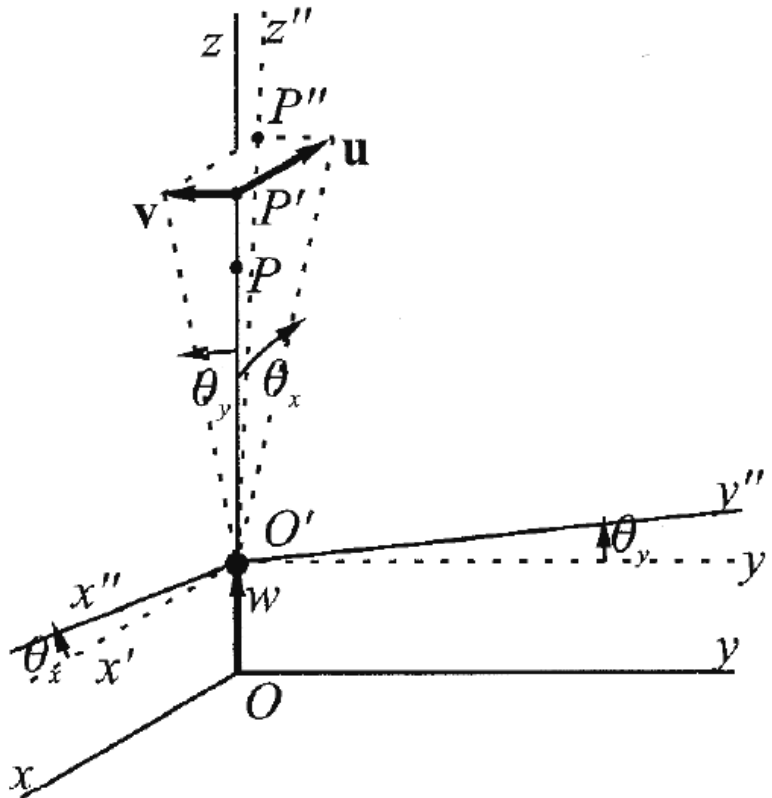
La teoría de Kirchoff establece las siguientes hipótesis referentes a la deformación de placas delgadas:

- Los desplazamientos transversales w de los puntos situados en la superficie media son pequeños en relación con el espesor t de la placa, siendo despreciables sus desplazamientos longitudinales u y v . En consecuencia, al ser la superficie media $u=v=0$, son nulos los esfuerzos σ_x , σ_y en esa superficie que es la superficie neutra en la flexión de la placa.

- Los puntos situados en una normal a la superficie media tienen el mismo desplazamiento transversal w y además permanecen situados en una normal a la superficie media deformada. La primera condición exige que $\varepsilon_z = \partial w / \partial z = 0$, y la última que no existan las distorsiones γ_{yz}, γ_{zx} , condición que equivale a la hipótesis de Bernoulli de que las secciones transversales permanezcan planas durante la flexión de vigas.

Sea O un punto situado en la superficie media de la placa y P el punto de la normal OP a esa superficie, tal que $OP = Z$ (Imagen 12). De acuerdo con la teoría de Kirchoff, la recta OP normal a la superficie media se transforma en la recta $O'P'$ normal a la superficie media deformada mediante un proceso de deformación que consta de las siguientes fases: una traslación sobre si misma de valor w que lleva OP a $O'P'$ y un giro θ alrededor de un eje que pasa por O' , de componentes θ_x, θ_y , que son los giros de z alrededor de los ejes y' y x' respectivamente. El giro θ_x se considera positivo cuando lleva el eje x' sobre el eje z y análogamente, el giro θ_y se considera positivo cuando lleva el eje x' sobre el eje y' sobre el eje z . Debido a estos giros, el punto P' experimenta los desplazamientos u y v definidos por:

Imagen 12: Rotación y traslación de un punto del interior de una placa plana sometida a flexión



$$u = -O'P'\theta_x \quad v = -O'P'\theta_y$$

Y por tanto:

$$\begin{aligned} u &= -z\theta_x = -z \frac{\partial w}{\partial x} \\ v &= -z\theta_y = -z \frac{\partial w}{\partial y} \end{aligned} \quad (2.35)$$

Teniendo en cuenta que las ecuaciones de compatibilidad de deformaciones, se determinan las deformaciones:

$$\begin{aligned} \varepsilon_x &= \frac{\partial u}{\partial x} = -z \frac{\partial^2 w}{\partial x^2} \\ \varepsilon_y &= \frac{\partial v}{\partial y} = -z \frac{\partial^2 w}{\partial y^2} \\ \gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = -2z \frac{\partial^2 w}{\partial x \partial y} \end{aligned} \quad (2.36)$$

Que definen el vector de deformaciones:

$$\varepsilon = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{pmatrix} -z \frac{\partial^2 w}{\partial x^2} \\ -z \frac{\partial^2 w}{\partial y^2} \\ -2z \frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} \quad (2.37)$$

Al ser $\sigma_z = 0$, para determinar el correspondiente vector de esfuerzos σ , utilizaremos la matriz constitutiva D de un estado de esfuerzos plano:

$$D = \frac{E}{1 - \mu^2} \begin{pmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1 - \mu}{2} \end{pmatrix} \quad (2.38)$$

Teniendo en cuenta que $\sigma = D\varepsilon$, al multiplicar las matrices (2.38) y (2.37), se obtiene el vector de esfuerzos σ :

$$\sigma = \begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = -\frac{Ez}{1-\mu^2} \begin{pmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{pmatrix} \begin{pmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} \quad (2.39)$$

Sustituyendo los valores de los esfuerzos σ_x , σ_y , τ_{xy} en las ecuaciones (2.32) de los momentos flectores M_x , M_y y momentos torsores M_{xy} e integrando, resulta:

$$\begin{pmatrix} M_x \\ M_y \\ M_{xy} \end{pmatrix} = -\frac{Et^3}{12(1-\mu^2)} \begin{pmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{pmatrix} \begin{pmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2\frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} \quad (2.40)$$

Sustituyendo la ecuación (2.40) en las ecuaciones de (2.32), se hallan las fuerzas cortantes en función del desplazamiento w :

$$\begin{aligned} T_x &= -D \left(\frac{\partial^3 w}{\partial x^3} + \mu \frac{\partial^3 w}{\partial x \partial y^2} \right) \\ T_y &= -D \left(\frac{\partial^3 w}{\partial y^3} + \mu \frac{\partial^3 w}{\partial y \partial x^2} \right) \end{aligned} \quad (2.41)$$

Finalmente, sustituyendo la ecuación (2.40) en la ecuación (2.34) se determina la ecuación diferencial de gobierno de la placa:

$$\frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \frac{q_{(x,y)}}{D} \quad (2.42)$$

Que es una ecuación diferencial de cuarto orden, que junto con las condiciones de contorno permite calcular el desplazamiento w en función de la carga transversal $q_{(x,y)}$ y de la rigidez a flexión de la placa D .

2.3.2.- Matriz de rigidez del elemento

Para satisfacer la continuidad del desplazamiento transversal w de un elemento de placa a flexión se eligen como parámetros nodales el desplazamiento w y los giros θ_x y θ_y . Es decir, en general habrá tres parámetros nodales y por ello, para un elemento de n nodos el número de variables será $3n$ que es el grado de libertad del elemento. Por tanto, la función de desplazamientos del elemento:

$$w_{(x,y)} = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \dots \quad (2.43)$$

Habr  de tener $3n$ t rminos cuyos coeficientes α_i se determinan imponiendo las condiciones de que w , $\theta_x = \partial w / \partial x$ y $\theta_y = \partial w / \partial y$ tengan unos valores determinados en los n nodos del elemento. Utilizando las funciones de forma $N_{i(x,y)}$, la funci n de desplazamientos del elemento es:

$$w_{(x,y)} = \sum_{i=1}^n (N_{wi} w_i + N_{\theta_x} \theta_x + N_{\theta_y} \theta_y) \quad (2.44)$$

Es decir:

$$w_{(x,y)} = Na \quad (2.45)$$

$$N = (N_1 \quad N_2 \quad \cdots \quad N_i \quad \cdots \quad N_{ne}) \quad (2.46)$$

$$a = (a_1 \quad a_2 \quad \cdots \quad a_i \quad \cdots \quad a_{ne})^T \quad (2.47)$$

$$a_i = (w_i \quad \theta_{xi} \quad \theta_{yi})^T \quad (2.48)$$

La matriz de rigidez de un elemento finito cualquiera es:

$$K = \int_V B^T D B dV \quad (2.49)$$

Siendo D la matriz constitutiva y B la matriz de deformaci n del elemento.

En un elemento finito de placa sometido a flexi n con nodos en sus extremos y desplazamientos nodales w_i , θ_{xi} , θ_{yi} , el vector de deformaciones es:

$$\varepsilon = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{pmatrix} -z \frac{\partial^2 w}{\partial x^2} \\ -z \frac{\partial^2 w}{\partial y^2} \\ -2z \frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} = \begin{pmatrix} -z \frac{\partial^2 N}{\partial x^2} \\ -z \frac{\partial^2 N}{\partial y^2} \\ -2z \frac{\partial^2 N}{\partial x \partial y} \end{pmatrix} a \quad (2.50)$$

De esto se deduce que la matriz de deformaci n del elemento es:

$$B = \begin{pmatrix} -z \frac{\partial^2 N}{\partial x^2} \\ \frac{\partial^2 N}{\partial y^2} \\ -2z \frac{\partial^2 N}{\partial x \partial y} \end{pmatrix} \quad (2.51)$$

Al sustituir la matriz B y $dV = dz dA$ en la ecuación (2.49):

$$K = \int_V B^T D B z^2 dV = \int_A B^T D B dA \int_{-t/2}^{t/2} z^2 dz \quad (2.52)$$

Se obtiene:

$$K = \int_A B^T D_p B dA \quad (2.53)$$

Dónde:

$$D_p = \frac{t^3}{12} D \quad (2.54)$$

2.3.3.- Respuesta de la placa

Conocidos los parámetros nodales en toda la placa, se conocerán los de cada elemento y ellos nos permitirán determinar los vectores de momentos M , de deformaciones ε y de esfuerzos σ en cada punto, es decir, la respuesta del elemento a las cargas aplicadas.

La expresión del vector de momentos es:

$$\begin{pmatrix} M_x \\ M_y \\ M_{xy} \end{pmatrix} = -\frac{Et^3}{12(1-\mu^2)} \begin{pmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \frac{1-\mu}{2} \end{pmatrix} \begin{pmatrix} \frac{\partial^2 w}{\partial x^2} \\ \frac{\partial^2 w}{\partial y^2} \\ 2 \frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} = D_p B a \quad (2.55)$$

El vector de deformación en cualquier punto de un elemento es:

$$\varepsilon = z B_p a \quad (2.56)$$

El vector de esfuerzos en el elemento es:

$$\sigma = z D B_p a \quad (2.57)$$

Esta ecuación se puede expresar en forma matricial como sigue:

$$w_{(x,y)} = (1 \quad x \quad y \quad x^2 \quad xy \quad y^2 \quad x^3 \quad x^2y \quad xy^2 \quad y^3 \quad x^3y \quad xy^3) \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \\ \alpha_9 \\ \alpha_{10} \\ \alpha_{11} \\ \alpha_{12} \end{pmatrix} \quad (2.59)$$

Es decir: $w_{(x,y)} = P\alpha \quad (2.60)$

Los doce coeficientes α_i , se hallan particularizando en la ecuación (2.59) los desplazamientos w y los giros $\partial w/\partial x = \theta_x$, $\partial w/\partial y = \theta_y$, en los cuatro nodos, obteniéndose:

$$a = C\alpha \quad (2.61)$$

Donde a es el vector de parámetros nodales:

$$a = (w_1 \quad \theta_{x1} \quad \theta_{y1} \quad w_2 \quad \theta_{x2} \quad \theta_{y2} \quad w_3 \quad \theta_{x3} \quad \theta_{y3} \quad w_4 \quad \theta_{x4} \quad \theta_{y4})^T \quad (2.62)$$

Y C la matriz:

$$C = \begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 & x_1^3 & x_1^2y_1 & x_1y_1^2 & y_1^3 & x_1^3y_1 & x_1y_1^3 \\ 0 & 1 & 0 & 2x_1 & y_1 & 0 & 3x_1^2 & 2x_1y_1 & y_1^2 & 0 & 3x_1^2y_1 & y_1^3 \\ 0 & 0 & 1 & 0 & x_1 & 2y_1 & 0 & x_1^2 & 2x_1y_1 & 3y_1^2 & x_1^3 & 3x_1y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_4 & y_4 & x_4^2 & x_4y_4 & y_4^2 & x_4^3 & x_4^2y_4 & x_4y_4^2 & y_4^3 & x_4^3y_4 & x_4y_4^3 \\ 0 & 1 & 0 & 2x_4 & y_4 & 0 & 3x_4^2 & 2x_4y_4 & y_4^2 & 0 & 3x_4^2y_4 & y_4^3 \\ 0 & 0 & 1 & 0 & x_4 & 2y_4 & 0 & x_4^2 & 2x_4y_4 & 3y_4^2 & x_4^3 & 3x_4y_4^2 \end{pmatrix} \quad (2.63)$$

De la ecuación (2.61), se deduce que:

$$\alpha = C^{-1}a \quad (2.63)$$

Que sustituida en la función de desplazamientos, ecuación (2.60), resulta:

$$w_{(x,y)} = PC^{-1}a \quad (2.63)$$

O más brevemente: $w_{(x,y)} = Na$ (2.64)

Dónde la matriz de funciones de forma está dada por:

$$N = PC^{-1} \quad (2.65)$$

2.3.4.2.- Matriz de rigidez

Sustituyendo la ecuación (2.58) de la función de desplazamientos en el vector de deformaciones (2.56), resulta:

$$\varepsilon = \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} = \begin{pmatrix} -z \frac{\partial^2 w}{\partial x^2} \\ -z \frac{\partial^2 w}{\partial y^2} \\ -2z \frac{\partial^2 w}{\partial x \partial y} \end{pmatrix} = -z \begin{pmatrix} 2\alpha_4 + 6\alpha_7x + 2\alpha_8y + 6\alpha_{11}xy \\ 2\alpha_6 + 2\alpha_9x + 6\alpha_{10}y + 6\alpha_{12}xy \\ 2\alpha_5 + 4\alpha_8x + 4\alpha_9y + 6\alpha_{11}x^2 + 6\alpha_{12}y^2 \end{pmatrix} \quad (2.66)$$

Es decir: $\varepsilon = -zQ\alpha$ (2.67)

Siendo:

$$Q = \begin{pmatrix} 0 & 0 & 0 & 2 & 0 & 0 & 6x & 2y & 0 & 0 & 6xy & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 2x & 6y & 0 & 6xy \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4x & 4y & 0 & 6x^2 & 6y^2 \end{pmatrix} \quad (2.68)$$

Teniendo en cuenta la ecuación (2.63) se tiene:

$$\varepsilon = -zQC^{-1}a \quad (2.69)$$

Y como:

$$\varepsilon = B_e a$$

Resulta:

$$B_e = -zQC^{-1} \quad (2.70)$$

Sustituyendo esta expresión en la ecuación (2.53), se obtiene la matriz de rigidez del elemento:

$$K = (C^{-1})^T \left(\int_{-b}^b \int_{-a}^a Q^T D_p Q \, dx \, dy \right) C^{-1} \quad (2.71)$$

2.3.4.3.- Vector de fuerzas nodales equivalente

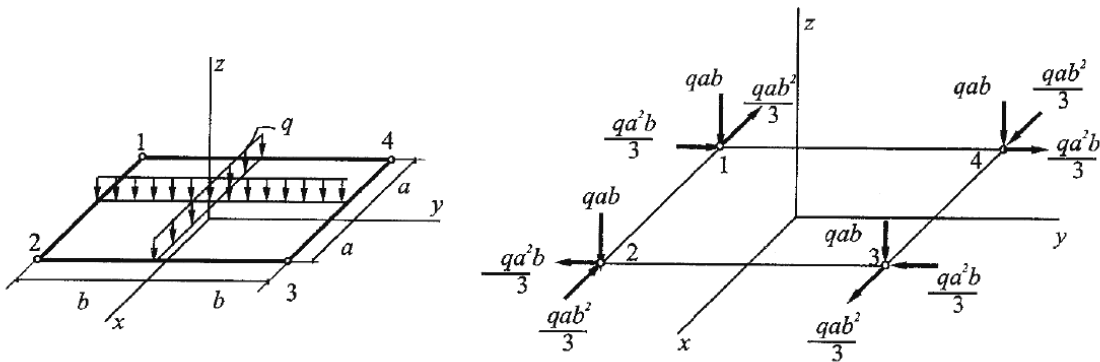
El vector de fuerzas nodales del elemento equivalente a la carga uniformemente distribuida q , de acuerdo con la ecuación (2.30) es:

$$F_{eq} = \int_A N^T q dA \quad (2.72)$$

Sustituyendo la ecuación (2.65) en (2.72) resulta el valor del vector de carga equivalente:

$$F_{eq} = (C^{-1})^T \int_{-b}^b \int_{-a}^a P^T q dx dy \quad (2.73)$$

Imagen 15: carga uniformemente distribuida y carga nodal equivalente



2.3.4.4.- Análisis de la conformidad del elemento

En los bordes $x = a$ o $y = b$ del elemento rectangular considerado (Imagen 16), la función de desplazamientos se convierte en una función cúbica en y o una función cúbica en x . En ambos casos, el desplazamiento en los bordes está definido por cuatro coeficientes que se determinan mediante cuatro parámetros de los dos nodos que delimitan el borde correspondiente. En efecto, haciendo $x = a$ se obtiene la función de desplazamientos del borde 2-3 que es la función cúbica:

$$w_{2-3} = w_{(a,y)} = \alpha'_1 + \alpha'_2 y + \alpha'_3 y^2 + \alpha'_4 y^3$$

Cuyos cuatro coeficientes se determinan mediante los cuatro parámetros nodales w_2 , $(\partial w / \partial y)_2$, w_3 , $(\partial w / \partial y)_3$.

Por consiguiente, al quedar totalmente definidos los desplazamientos w en los bordes en función de los parámetros nodales, el elemento satisfará la condición de continuidad inter elemental en los desplazamientos w .

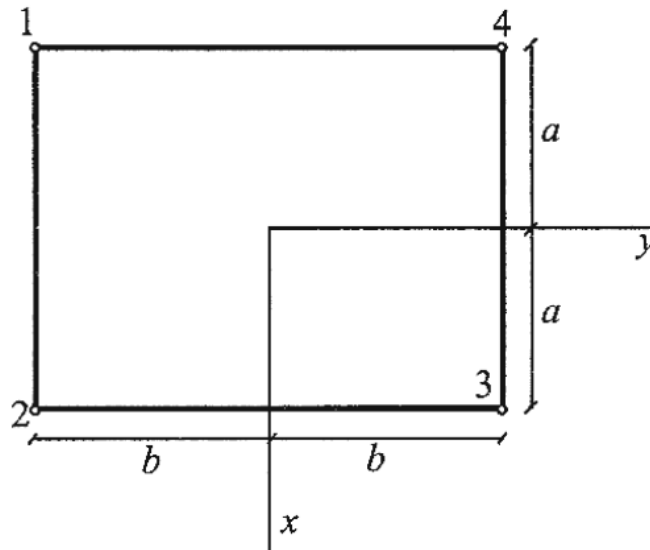
Por el contrario, no existe continuidad en los bordes en la derivada $\partial w/\partial n$, del desplazamiento $w_{(x,y)}$ respecto a la normal al borde. En efecto, $(\partial w/\partial x)$ a lo largo del borde 2-3, es decir, para $x = a$ es la función cúbica:

$$\left(\frac{\partial w}{\partial x}\right)_{2-3} = \alpha''_1 + \alpha''_2 y + \alpha''_3 y^2 + \alpha''_4 y^3$$

Cuyos cuatro coeficientes no pueden ser determinados mediante los dos parámetros nodales $(\partial w/\partial x)_2$ y $(\partial w/\partial x)_3$. En consecuencia, al haber un coeficiente indeterminado no está asegurada la continuidad inter elemental de las segundas derivadas en el borde 2-3 y lo mismo sucederá en otros bordes.

En resumen, habiendo elegido como parámetros nodales el desplazamiento w y sus primeras derivadas $\partial w/\partial x$, $\partial w/\partial y$ no se ha podido garantizar la continuidad en los bordes de la derivada normal $\partial w/\partial n$ ni la de la derivada cruzada $\partial^2 w/\partial x \partial y$. Debido a ello el elemento no es conforme pero produce resultados convergentes cuando se disminuye el tamaño de las mallas y por ello su uso es satisfactorio.

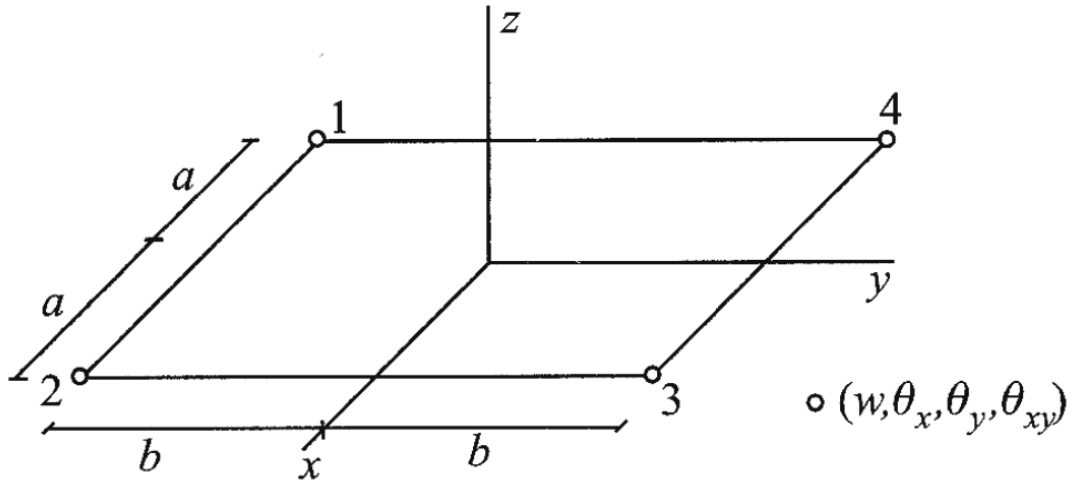
Imagen 16: Elemento finito rectangular (análisis de conformidad)



2.3.5.- Elemento de placa rectangular de 16 grados de libertad

Para satisfacer la continuidad en los bordes de la derivada normal $\partial w/\partial n$ y de la segunda derivada $\partial^2 w/\partial x \partial y$, que no tiene el elemento rectangular de 12 g.d.l., se incorpora la derivada $\partial^2 w/\partial x \partial y$ como parámetro nodal. En este caso el elemento de placa rectangular tiene en cada nodo 4 grados de libertad w , $\partial w/\partial x = \theta_x$, $\partial w/\partial y = \theta_y$, $\partial^2 w/\partial x \partial y = \theta_{xy}$ (Imagen 16) y, por tanto, un total de 16 grados de libertad, lo que exige un polinomio de desplazamientos de 16 términos.

Imagen 17: Elemento finito de placa de 16 grados de libertad



Eligiendo como función de desplazamientos un polinomio completo de tercer grado y como términos adicionales los términos de cuarto orden x^3y , x^2y^2 , xy^3 , los términos de quinto orden x^3y^2 , x^2y^3 y el término de sexto orden x^3y^3 , se obtiene:

$$w_{(x,y)} = \alpha_1 + \alpha_2x + \alpha_3y + \alpha_4x^2 + \alpha_5xy + \alpha_6y^2 + \alpha_7x^3 + \alpha_8x^2y + \alpha_9xy^2 + \alpha_{10}x^3 + \alpha_{11}x^3y + \alpha_{12}x^2y^2 + \alpha_{13}xy^3 + \alpha_{14}x^3y^2 + \alpha_{15}x^2y^3 + \alpha_{16}x^3y^3 \quad (2.74)$$

Los 16 coeficientes α_i de la función de desplazamientos w se obtienen particularizando en la ecuación (2.74) los parámetros nodales w , $\partial w/\partial x$, $\partial w/\partial y$, $\partial^2 w/\partial x \partial y$. Si se utilizan funciones de forma, la función de desplazamientos del elemento rectangular considerado es $w_{(x,y)} = Na$.

En los bordes $x = \pm a$ o $y = \pm b$ del elemento rectangular de 16 g.d.l., la función de desplazamientos (7.70) se convierte en una función cúbica en y o una función cúbica en x . En efecto, haciendo $x = +a$ se obtiene la función de desplazamientos del borde 2 - 3 que es la función cúbica en y :

$$w_{2-3} = w_{(a,y)} = \alpha'_1 + \alpha'_2y + \alpha'_3y^2 + \alpha'_4y^3$$

Cuyos cuatro coeficientes se determinan mediante los cuatro parámetros nodales w_2 , $(\partial w/\partial y)_2$, w_3 , $(\partial w/\partial y)_3$. Por consiguiente, al quedar totalmente definidos los desplazamientos w en los bordes en función de los parámetros nodales, el elemento satisfará la condición de continuidad inter elemental en los desplazamientos w . También existe continuidad en los bordes de la derivada $\partial w/\partial n$ del desplazamiento w respecto a la normal al borde. En efecto, $\partial w/\partial x$ a lo largo del borde 2 - 3, es decir, para $x = +a$ es una función cúbica en y :

$$\left(\frac{\partial w}{\partial x}\right)_{2-3} = \alpha''_1 + \alpha''_2 y + \alpha''_3 y^2 + \alpha''_4 y^3$$

Cuyos cuatro coeficientes quedan determinados mediante los cuatro parámetros nodales $(\partial w/\partial x)_2$, $(\partial^2 w/\partial x \partial y)_2$, $(\partial w/\partial x)_3$ y $(\partial^2 w/\partial x \partial y)_3$. Por esta razón, está asegurada la continuidad de las derivadas $\partial w/\partial n$ en los bordes. Al derivar respecto a y la pendiente $\partial w/\partial x$ a lo largo del borde 2 - 3, se obtiene:

$$\left(\frac{\partial^2 w}{\partial x \partial y}\right)_{2-3} = \alpha''_2 + 2\alpha''_3 y + 3\alpha''_4 y^2$$

Cuyos tres coeficientes han sido ya determinados mediante los parámetros nodales. En consecuencia, está también asegurada la continuidad de las segundas derivadas en el borde 2 - 3 y lo mismo sucederá en los restantes bordes.

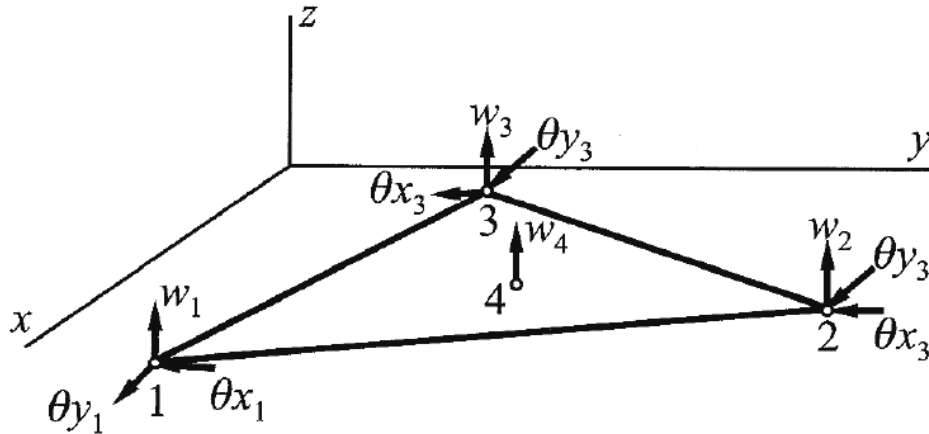
En resumen, habiendo elegido como parámetros nodales el desplazamiento w , sus primeras derivadas $\partial w/\partial x$, $\partial w/\partial y$, y la segunda derivada $\partial^2 w/\partial x \partial y$ se ha podido garantizar la continuidad en los bordes del desplazamiento w , de la derivada $\partial w/\partial n$ y de la segunda derivada $\partial^2 w/\partial x \partial y$, por lo que el elemento rectangular de 16 g.d.l. es conforme. Sin embargo, el añadir como parámetro nodal la segunda derivada $\partial^2 w/\partial x \partial y$ no da resultado cuando los ángulos del elemento no son rectos y, por ello, los elementos cuadriláteros irregulares no son conformes.

2.3.6.- Elemento triangular de 10 grados de libertad

Un elemento de placa triangular con 3 grados de libertad w , $\partial w/\partial x = \theta_x$, $\partial w/\partial y = \theta_y$, en cada nodo y 1 grado de libertad w en un nodo central tendrá un total de 10 grados de libertad (Imagen 18). Ello exige un polinomio de desplazamientos de diez términos, por lo que elegiremos el polinomio completo de tercer grado:

$$w_{(x,y)} = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 x^2 y + \alpha_9 xy^2 + \alpha_{10} y^3 \quad (2.75)$$

Imagen 18: Elemento triangular de 10 grados de libertad



Donde los coeficientes α_i se hallan particularizando la ecuación (2.75), al igual que en los elementos previamente estudiados, tanto la matriz de rigidez, como también la respuesta de la placa, se hallan de la misma manera que en un elemento rectangular, por lo que no se desarrolla las ecuaciones nuevamente.

2.3.6.1.- Análisis de la conformidad del elemento

Al sustituir en la función de desplazamientos (2.75) las ecuaciones que transforman las coordenadas x, y en las coordenadas naturales ξ, η se obtiene:

$$w_{(\xi, \eta)} = \beta_1 + \beta_2 \xi + \beta_3 \eta + \beta_4 \xi^2 + \beta_5 \xi \eta + \beta_6 \eta^2 + \beta_7 \eta^3 + \beta_8 \xi^2 \eta + \beta_9 \xi \eta^2 + \beta_{10} \eta^3 \quad (2.76)$$

En los bordes $\xi = 0, \eta = 0, 1 - \xi - \eta = 0$, la función de desplazamientos (2.76) se convierte en una función cúbica en ξ o η . En ambos casos, el desplazamiento en los bordes está definido por cuatro coeficientes que se determinan mediante cuatro parámetros de los dos nodos que delimitan el borde correspondiente, quedando dos parámetros nodales sin utilizar. Por consiguiente, al quedar definidos los desplazamientos w en los bordes en función de los parámetros nodales, el elemento satisfará la condición de continuidad inter elemental en los desplazamientos w .

Por el contrario, no existe continuidad en los bordes en la derivada $\partial w / \partial n$ del desplazamiento w respecto a la normal al borde. En efecto, la derivada $\partial w / \partial n$ a lo largo de un borde cualquiera es una función cuadrática en ξ o η cuyos tres coeficientes no pueden ser determinados con los dos parámetros nodales que no se han utilizado al establecer la continuidad inter elemental en los desplazamientos w . Por esta razón, no está asegurada la continuidad de las derivadas $\partial w / \partial n$ en los bordes. Debido a ello el elemento no es conforme pero produce resultados convergentes cuando se disminuye el tamaño de las mallas.

2.4.- Programación orientada a objetos (P.O.O.)

La programación orientada a objetos (POO), consiste en encapsular datos y métodos en objetos, de forma que cada objeto sea semiautónomo, encerrando métodos y datos privados (es decir internos) y salvándolos del desorden general que les rodea. Así el objeto puede interactuar con el resto del programa por medio de una interfaz bien definida por su método públicos (es decir, que se les puede invocar desde afuera).

La programación orientada a objetos fue creada para gestionar programas más grandes y descomponerlos en unidades funcionales. Esto nos lleva al siguiente paso, que consiste en dividir un programa en subrutinas, ya que los objetos pueden contener múltiples subrutinas y datos. El resultado de encapsular partes de un programa en un objeto es que es concebido como un elemento sencillo y no hay que tratar todo lo que el objeto hace internamente.

La programación orientada a objetos gira sobre algunos conceptos clave: clases, objetos, miembros de datos, métodos y herencia. Conceptos que se detallan a continuación. (Holzner)

2.5.- Clases

En la programación orientada a objetos, las clases proporcionan una especie de plantilla para los objetos. Es decir, si se piensa en una clase como un molde de galletas, los objetos que se crean a partir de ella, son las galletas. Se puede considerar que una clase es un tipo de objeto; se usa una clase para crear un objeto y luego se puede llamar a los métodos del objeto desde este código.

Para crear un objeto, se invoca al constructor de una clase, que es un método que se llama igual que la clase. Este constructor crea un nuevo objeto de la clase.

2.5.1.- Clases en MatLab

A partir de la versión 5, MatLab permite la creación de nuevas clases por parte del usuario y la posibilidad de definir nuevas operaciones para los tipos de datos básicos.

Los tipos de datos fundamentales (Clases) en MatLab son los siguientes:

- **double.**- Número de coma flotante de doble precisión.
- **single.**- Número de precisión simple en coma flotante.
- **char.**- Variable de tipo carácter.
- **logical.**- Variable de valor verdadero o falso.
- **int8.**- Variable entero con signo de ocho bits de memoria.
- **cell.**- Variable vector de tipo celda.
- **struct.**- Variable de tipo estructura de datos.

2.5.2.- Operaciones

Para cada una de las clases, MatLab define unas operaciones concretas. Por ejemplo, se pueden sumar variables de tipo *double*, pero no variables de tipo *cell*, se pueden concatenar variables de tipo carácter pero no variables de tipo *struct*.

2.5.3.- Clases creadas por el usuario

Para crear una nueva clase hay que crear el directorio de clase. Este directorio debe contener como mínimo dos ficheros “M” del tipo de la nueva clase creada: *Clase.m* y *display.m*. El primero de ellos es el constructor de la clase mientras que el segundo se usa para visualizar la nueva variable en la ventana de comandos. Aparte de estos dos ficheros habrá que definir ficheros “M” de métodos que permitan operar con la nueva clase creada.

2.6.- Objetos

Un objeto es una instancia de una clase, al igual que una variable es una instancia de un tipo de dato. Se puede pensar en una clase como el tipo de un objeto y se puede pensar en el objeto como una instancia de una clase. Los objetos encapsulan métodos y variables de instancia. Para crear un objeto, se llama al constructor de una clase, que tiene el mismo nombre que ella.

2.7.- Miembros de datos

Los miembros de datos de un objeto se llaman miembros de datos de instancia o variables de instancia. Los elementos de datos compartidos por todos los objetos de una clase se llaman miembros de datos de clase o variables de clase. Los miembros de datos pueden hacerse accesibles desde fuera de un objeto, o se puede hacer que sean internos al objeto para usar, de forma privada, los métodos del interior del objeto.

2.8.- Métodos

Los métodos son funciones de una clase. Generalmente los métodos se dividen en aquellos que se usan internamente en la clase, llamados métodos privados (*private*), los que se usan fuera de la clase, llamados métodos públicos (*public*) y los que son usados por la clase y sus derivadas, llamados métodos protegidos (*protected*).

Los métodos privados son, generalmente, llamados en el interior del objeto por otras partes del mismo. Una vez que se tiene un objeto que soporta métodos, se pueden usar los métodos de esos objetos.

La colección de reglas o ficheros “M” que redefinen operadores y funciones, reciben el nombre de métodos. Las operaciones sobre objetos se especifican por medio de métodos que encapsulan los datos y redefinen operadores y funciones.

El encapsulamiento de los objetos impide que ciertas propiedades sean visibles desde la ventana de comandos con lo que para acceder a ellas hay que usar los métodos definidos para la clase.

2.8.1.- Sobrecarga de métodos

La sobrecarga de métodos es una técnica de la programación orientada a objetos que permite definir diferentes versiones de un método, todos con el mismo nombre pero con diferentes listas de parámetros. Cuando se usa un método sobrecargado, el intérprete de MatLab sabrá cuál es el que se quiere utilizar por el número y el tipo de parámetros que se le pasen, y buscará la versión del método con la lista de parámetros correcta.

2.9.- Composición

La composición es el agrupamiento de uno o varios objetos y valores, como atributos que conforman el valor de los distintos objetos de una clase. Normalmente los atributos contenidos se declaran con acceso privado (*private*) y se inicializan en el constructor de la nueva clase.

2.10.- Herencia

La herencia es uno de los aspectos de la programación orientada a objetos que se ha definido formalmente. Utilizando la herencia, se puede derivar una nueva clase a partir de una antigua, y la nueva heredará todos los métodos y miembros de datos de la antigua. La clase nueva se llama clase derivada y la clase original, clase base. La idea es añadir lo que se quiera a la nueva clase para darle más funcionalidad que a la clase base.

2.11.- Directorio de clase

Las reglas definidas para interpretar operadores y funciones son ficheros “M” de tipo función que se guardan en los directorios de clase (*class directories*) de MatLab.

Estos directorios no pueden estar directamente en el *search path* de MatLab, pero son subdirectorios de directorios que sí están en el *search path* de MatLab. Es posible que haya múltiples directorios para el mismo tipo de datos (en ese caso, cuando MatLab busque funciones en estos directorios seguirá el orden dado en el *search path*).

2.12.- Fichero constructor de clase

Debe llamarse como la clase, es decir *nombreClase.m*, debe manejar tres tipos de entradas:

- 1) Si no se le pasan argumentos de entrada, debe generar una variable vacía.
- 2) Si el argumento de entrada es de su misma clase debe pasarlo directamente a la salida.
- 3) Si el argumento de entrada son datos para crear la nueva clase, debe crear una variable de dicha clase.

(Jesús María Zamarreño Cosme)

(11/12b)

3.1.- Procedimiento de cálculo de placas según el método de los elementos finitos

A continuación se expone, de forma ordenada, el procedimiento que se lleva a cabo para hallar la solución de un problema de placa cargada. Se considera que se tiene una placa en el plano XY cargada con cargas puntuales y distribuidas, así como con momentos. Las características del material se suponen conocidas (módulo de elasticidad, espesor y módulo de Poisson). La placa está, asimismo, sujeta a una serie de condiciones de contorno (apoyos) para evitar la singularidad del sistema de ecuaciones.

Usando una discretización del espacio continuo, es decir, creando una malla sobre la placa, se quiere calcular los desplazamientos y tensiones en los nodos y aproximar la solución en cualquier otro punto de la placa.

3.1.1.- Creación de la malla

Inicialmente se crea una malla de elementos sobre la superficie de la placa, tanto el tipo de elemento como la forma y densidad de la malla, deben ser escogidos por el usuario del programa. Una mayor densidad en la malla suele proporcionar resultados más satisfactorios pero, evidentemente, requiere un mayor esfuerzo computacional.

Asimismo los elementos rectangulares creados deben ser rectángulos y tener los lados paralelos a los ejes x e y .

Los elementos triangulares pueden tener cualquier forma, pero se debe asignar los nodos del elemento en el sentido contrario a las agujas del reloj.

En el presente proyecto se ha desarrollado tres algoritmos sencillos para generar mallas dependiendo del tipo de elemento escogido para modelar y resolver el problema, los cuales se explican más adelante.

3.1.2.- Propiedades geométricas de los elementos finitos adoptados para el programa

Cuando se ha definido el tipo de elemento y la forma de la malla, se les asigna a los elementos que conforman la malla, las propiedades materiales (módulo de elasticidad, espesor y módulo de Poisson) y se procede a calcular las propiedades geométricas de cada uno. Esto es, las diferentes variables y matrices auxiliares (como son las coordenadas, las diferentes submatrices y las matrices de funciones de forma) y la matriz de rigidez del elemento.

3.1.3.- Creación de los vectores de carga

Cada uno de los nodos del elemento tiene asignado un vector de cargas resultantes. El vector de cargas resultantes es la suma de todas las cargas aplicadas sobre el nodo y tiene la forma:

$$f_i = (F_z \quad M_x \quad M_y)^T$$

El vector de carga debe ser obtenido según el tipo de elemento finito elegido, a partir de sus correspondientes funciones de forma.

Para las cargas puntuales, el vector se suma directamente al vector resultante del nodo donde están aplicadas.

3.1.4.- Ensamblaje del vector de cargas y la matriz de rigidez total de la estructura

Para obtener la solución del sistema es necesario calcular la matriz de rigidez global y el vector de cargas nodales global. Se requiere, por lo tanto, ensamblar las matrices de rigidez de todos los elementos en una única matriz y ensamblar también los vectores de cargas nodales en un único vector. Para ello se considera una numeración global y unívoca de todos los nodos del sistema.

Sea un sistema con n nodos y m grados de libertad para cada nodo. La matriz de rigidez global es una matriz simétrica $mn \times mn$ y el vector de cargas nodales global tiene dimensiones $mn \times 1$, y se puede dividir en $n \times n$ sub matrices $m \times m$, como se ve en la ecuación (3.1). $[K_{ij}]_{global}$ es la sub matriz de nodos ij según la numeración global de los mismos.

$$K_{global}^{mn \times mn} = \begin{pmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \ddots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{pmatrix} \quad (3.1)$$

Sea la ecuación (3.1) la matriz de rigidez de un elemento, se divide la matriz en mn sub matrices K_{ij} , de dimensión $m \times m$, numeradas según la numeración local de los nodos del elemento. Así $[K_{ij}]_{elemento}$ representa los nodos locales del elemento.

3.1.5.- Asignación de las condiciones de contorno

Para que el sistema tenga solución, es necesario añadir las condiciones de contorno pertinentes a los nodos. Las condiciones de contorno corresponden a los valores de los movimientos nodales prescritos. Las condiciones más habituales son:

- 1) Borde empotrado: $w = \theta_x = \theta_y = 0$
- 2) Borde simplemente apoyado en X: $w = \theta_y = 0$
- 3) Borde simplemente apoyado en Y: $w = \theta_x = 0$
- 4) Borde simplemente apoyado: $w = 0$

En el caso del elemento rectangular de 16 grados de libertad, se asigna una condición adicional en el giro rotacional, siendo este el cuarto grado de libertad asignado al nodo, dependiendo si el giro está restringido o no.

Se asignan, entonces, las condiciones de contorno correspondientes a los nodos, anulando las flechas y giros según sea conveniente.

3.1.6.- Solución general por el método de los desplazamientos en elementos finitos

En el caso general de cualquier estructura, el planteamiento matricial de la misma por el método de los elementos finitos es:

$$F = K\delta \quad (3.2)$$

Después de una reordenación adecuada, se obtiene la siguiente expresión:

$$\begin{pmatrix} F_n \\ \dots \\ R_a \end{pmatrix} = \begin{pmatrix} K_{nn} & \vdots & K_{na} \\ \dots & \dots & \dots \\ K_{an} & \vdots & K_{aa} \end{pmatrix} \begin{pmatrix} \delta_n \\ \dots \\ \delta_a \end{pmatrix} \quad (3.3)$$

Dónde:

- F_n : es el vector de cargas aplicadas (conocidas)
- R_a : es el vector de las reacciones de los apoyos (desconocidas)
- δ_n : es el vector de desplazamientos de los nudos libres (desconocidos)
- δ_a : es el vector de los desplazamientos de los apoyos (conocidos y generalmente iguales a cero)

Expandiendo la ecuación (3.3), se obtiene:

$$\begin{aligned} F_n &= K_{nn}\delta_n + K_{na}\delta_a & (a) \\ R_a &= K_{an}\delta_n + K_{aa}\delta_a & (b) \end{aligned} \quad (3.4)$$

Y despejando el vector de δ_n :

$$\delta_n = K_{nn}^{-1}(F_n - K_{na}\delta_a) \quad (3.5)$$

Reemplazando este valor en la ecuación (3.4b)

$$R_a = K_{an}K_{nn}^{-1}F_n + (K_{aa} - K_{an}K_{nn}^{-1}K_{na})\delta_a \quad (3.6)$$

Las ecuaciones (3.5) y (3.6) constituyen la base de la solución matricial de una estructura por el método de los elementos finitos.

3.1.7.- Aproximación de los desplazamientos y tensiones en los puntos interiores de la estructura

Una vez obtenidos los desplazamientos nodales, se pueden aproximar los desplazamientos y tensiones en cualquier punto de la placa. Los desplazamientos de un punto se obtienen interpolando los desplazamientos nodales del elemento donde se encuentra el punto, según las funciones de forma del elemento.

La forma general para los diferentes elementos finitos empleados en este proyecto de investigación, se muestra a continuación:

3.1.7.1.- Elemento rectangular de 12 grados de libertad

Este elemento es el más común y debe ser siempre un rectángulo de ángulo recto, cuenta con cuatro nodos y cada nodo con tres grados de libertad, los cuales son: w_i , θ_{ix} , θ_{iy} , los desplazamientos internos se muestran a continuación:

$$\begin{pmatrix} W(x,y) \\ \theta_x(x,y) \\ \theta_y(x,y) \end{pmatrix} = \sum_{i=1}^4 \left(N_{i(x,y)} \quad \frac{\partial N_i}{\partial x} \quad \frac{\partial N_i}{\partial y} \right) \begin{pmatrix} w_i \\ \theta_{ix} \\ \theta_{iy} \end{pmatrix} \quad (3.7)$$

3.1.7.2.- Elemento rectangular de 16 grados de libertad

En este elemento además de los grados de libertad mencionados: w_i , θ_{ix} , θ_{iy} , se añade un grado de libertad adicional que es el giro rotacional θ_{ixy} . Los desplazamientos internos se determinan por:

$$\begin{pmatrix} W(x,y) \\ \theta_x(x,y) \\ \theta_y(x,y) \\ \theta_{xy}(x,y) \end{pmatrix} = \sum_{i=1}^4 \left(N_{i(x,y)} \quad \frac{\partial N_i}{\partial x} \quad \frac{\partial N_i}{\partial y} \quad \frac{\partial^2 N_i}{\partial x \partial y} \right) \begin{pmatrix} w_i \\ \theta_{ix} \\ \theta_{iy} \\ \theta_{ixy} \end{pmatrix} \quad (3.8)$$

3.1.7.3.- Elemento triangular de 10 grados de libertad

El elemento triangular de 10 grados de libertad cuenta con cuatro nodos, tres ubicados en cada uno de sus vértices y el cuarto nodo ubicado en su centroide, los nodos ubicados en sus vértices poseen tres grados de libertad mientras que el nodo en su centroide solo posee un grado de libertad que resulta ser su desplazamiento vertical.

3.2.- Entorno de desarrollo utilizado: MatLab

Para crear el programa de cálculo de placas por MEF se ha optado por MatLab de MathWorks. Se trata de un *software* de uso muy extendido y que goza de gran popularidad. Es una potente herramienta de desarrollo de *software* técnico, que aun así conserva una gran simplicidad de uso y la programación es cómoda. Entre sus principales características están la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Una de las características más importantes para este proyecto, es que MatLab, permite tanto la programación orientada a objetos como la programación procedimental.

La programación orientada a objetos, permite la reutilización de código, la herencia, el encapsulamiento y el comportamiento de referencia sin necesidad de prestar atención a las tareas habituales de bajo nivel requeridas en otros lenguajes.

Este nuevo paradigma de programación supone un gran paso adelante y una evolución respecto a lenguajes de programación procedimental “clásicos” como FORTRAN, que se han usado extensivamente para calcular problemas de elementos finitos. Permite una

creación más cómoda, entendible, potente, adaptable y reutilizable de código. Dada la filosofía abierta y colaborativa de este proyecto, estas características son imprescindibles para cumplir los objetivos establecidos.

3.3.- Estructura del programa

El programa fue desarrollado con la siguiente estructura jerárquica de las clases que lo componen, la cual se muestra de manera simplificada:

3.3.1.- Clase Nodo: Esta clase no cuenta con clases derivadas, aquí se generan los nodos de la malla de la estructura y se crean los vínculos.

3.3.2.- Clase Elemento: sus clases derivadas son:

1. Clase ElemRec12GDL
2. Clase ElemRec16GDL
3. Clase ElemTriang10GDL

Cada una de estas clases se encarga de definir las propiedades del tipo de elemento finito que definen y luego envían sus resultados a la clase estructura, que es donde se resuelve el problema final.

3.3.3.- Clase Estructura: Esta clase no cuenta con clases derivadas. Aquí se calculan los desplazamientos nodales, las reacciones en los vínculos de la estructura y se determinan la distribución de tensiones y los momentos flectores internos.

3.4.- Descripción de las clases desarrolladas en el desarrollo del programa

3.4.1.- Clase Nodo

Esta es la clase que gestiona los datos de los diferentes nodos que aparecen en la estructura discretizada, tanto como sus coordenadas (x,y) como también el número de grados de libertad o las restricciones con las que cuenta el objeto Nodo en cuestión. A continuación se describen sus propiedades y sus métodos más relevantes.

Tabla 1: Propiedades de la clase Nodo

Propiedad	Descripción
x,y	Coordenadas x,y del nodo.
numGDL	Número de grados de libertad del nodo.
vecCarga	Vector de carga aplicado al nodo que está en función del número de grados de libertad.
vecDesp	Vector de desplazamientos del nodo que está en función del número de grados de libertad.
numNodo	Número índice del nodo de su ubicación en la estructura discretizada.

Tabla 2: Métodos de la clase Nodo

Método	Descripción
Nodo(numNodo,x,y,numGDL)	Este es el método constructor de la clase Nodo, que recibe como parámetros: el número de nodo, las coordenadas x,y y el número de grados de libertad del nodo.
distanciaEntreNodos(n1,n2)	Este método calcula la distancia euclidiana entre los nodos dados por parámetro.
formanAngRect(n1,n2,n3)	Este método determina si los tres nodos dados por parámetro, forman un ángulo recto tomando como base el nodo $n1$.
Vinculo(nombreVinculo)	Este método asigna el tipo de vínculo que corresponde al nodo y modifica su vector de desplazamientos y su vector de carga, según sea el caso del tipo de vínculo dado por parámetro.
mallasInternaTriangular(ni,nj,nk, numDiv)	Este es uno de los métodos más importantes, ya que genera una malla de nodos equi espaciados a partir de los nodos: ni, nj, nk , y el número de divisiones dados por parámetro.
mallasRectangular(longX,longY, divVer, divHor,numGDL)	Este método genera una malla rectangular de nodos igualmente espaciados, a partir de las dimensiones del rectángulo que se desea discretizar y del número de grados de libertad que tendrá cada nodo.

3.4.2.- Clase Elemento

La clase Elemento es la clase madre de cual heredan sus atributos los diferentes tipos de elementos finitos formulados en la presente investigación, para resolver una placa sometida a flexión, ya sean triangulares o rectangulares y con diferentes grados de libertad.

A continuación se muestran sus propiedades y sus métodos más relevantes.

Tabla 3: Propiedades de la clase Elemento

Propiedad	Descripción
numNodos	Número de nodos que componen el elemento.
numElemento	Número de elemento.
numGDL	Número total de grados de libertad del elemento.
modElast	Módulo de elasticidad del material del elemento.
mu	Módulo de Poisson del material.
espesor	Espesor de la placa.
matN	Matriz de coeficientes de las funciones de forma del elemento.
matC	Matriz de coeficientes con los valores de la función de forma particularizada a las coordenadas de los nodos del elemento.
matH	Matriz H resultante del triple producto matricial $QT D Q$ y de la doble integración de sus elementos.

matConst	Matriz de constitución del elemento. (Matriz D).
matRig	Matriz de rigidez del elemento. (Matriz K).
vecDesplazamiento	Vector de desplazamientos nodales del elemento.
vecFuerza	Vector de fuerzas nodales equivalentes, resultado de las diferentes cargas aplicadas a la estructura.

Tabla 4: Métodos de la clase Elemento

Método	Descripción
Elemento(numElemento,numNodo, numGDL,nodElast,mu,espesor)	Método constructor de la clase Elemento que recibe como parámetros: el número de elemento, número de nodos, número de grados de libertad, módulo de elasticidad del material, módulo de Poisson, espesor de la placa.
matrizD(E,mu,t)	Método que determina la matriz de constitución del material a partir de: módulo de elasticidad, módulo de Poisson y del espesor de la placa.
setMatrizRigidez(e,matC,matH)	Método que determina la matriz de rigidez del elemento a partir de la matriz C y de la matriz H.
setEcuacionesDeformacion	Este método determina las ecuaciones de deformación del elemento, después de que son hallados sus desplazamientos nodales.
setEcuacionesMomento	Método que determina las ecuaciones de los momentos flectores del elemento después de ser hallados sus desplazamientos nodales.

3.4.3.- Clase ElemRec12GDL

Esta clase es una clase derivada de la clase Elemento y está particularizada para elementos de 12 grados de libertad. A continuación se muestran sus propiedades y sus métodos más relevantes:

Tabla 5: Propiedades de la clase ElemRec12GDL

Propiedad	Descripción
Propiedades heredadas de la clase Elemento.	Las propiedades de material, número de grados de libertad y número de nodos, se crean directamente en el constructor de la clase Elemento.
nodo1, nodo2, nodo3, nodo4	Nodos del elemento rectangular de 12 grados de libertad.

Tabla 6: Métodos de la clase ElemRec12GDL

Método	Descripción
ElemRec12GDL(numEl,modElast, mu,esp,n1,n2,n3,n4)	Método constructor de la clase ElemRec12GDL, que recibe como parámetros: número de elemento, módulo de elasticidad, módulo de Poisson, espesor de

	placa, los nodos n1, n2, n3, n4.
setFuncionesDeForma(e)	Método que calcula los coeficientes de las funciones de forma del elemento rectangular de 12 GDL (matriz N), a partir de un objeto elemento. En este método también se calcula la matriz C.
setMath(e)	Método que calcula la matriz resultante del producto QT D Q y de integrar sus elementos.
fijarNodos(e,n1,n2,n3,n4)	Método que verifica si los cuatro nodos dados por parámetro definen un rectángulo en el plano.

Las clases ElemRec16GDL y ElemTriang10GDL, presentan las mismas cualidades que la clase elemento rectangular de 12 grados de libertad, cada una orientada específicamente a su tipo de elemento.

3.4.4.- Clase Estructura

Esta es la clase principal del programa, ya que en esta clase es donde se definen las cargas, los nodos vinculados con sus respectivos tipos de vínculo y se determinan a partir de las matrices de rigidez de los diferentes elementos que conforman la estructura, la matriz de rigidez ensamblada de la estructura como también su vector de cargas ensamblado. A continuación se describe sus atributos y métodos más relevantes.

Tabla 7: Propiedades de la clase Estructura

Propiedad	Descripción
listElemento	Lista de elementos discretizados que conforman la estructura.
listNodo	Lista de nodos que definen la estructura discretizada.
matRigidez	Matriz de rigidez ensamblada de la estructura.
vecFuerza	Vector de fuerzas nodales equivalentes, resultado de las diferentes cargas a las que se somete la estructura.
vecDesplazamiento	Vector de desplazamientos nodales de la estructura.

Tabla 8: Métodos de la clase Estructura

Método	Descripción
Estructura(listElemento)	Método constructor de la clase Estructura, que recibe como parámetro la lista de elementos discretizados que conforman la estructura.
setListNodo(listElemento)	Método que crea la lista de nodos de la estructura.
ensamblarMatriz(elemento)	Método que ensambla la matriz de rigidez del elemento dado por parámetro y lo ensambla a la matriz de rigidez de la estructura.
ensamblarCarga(elemento)	Método que ensambla el vector de carga del elemento dado por parámetro y lo ensambla al vector de carga de

	la estructura.
ensamblar(estructura)	Método que recorre la lista de elementos que conforman la estructura y ensambla sus matrices de rigidez y sus vectores de carga
vinculo(numNodo,tipoVinculo)	Método que fija el tipo de vínculo dado por parámetro al nodo cuyo número índice está dado por parámetro.
vincularNodos(listNodo,tipoVinculo)	Método que recibe una lista con los números de nodo a ser vinculados con el tipo de vínculo dado por parámetro.
cargar(elemento,carga)	Método que asigna la carga uniforme superficial dada por parámetro al elemento dado por parámetro.
cargarEstructura(carga)	Método que carga todos los elementos de la estructura con la carga superficial dada por parámetro.
resolver(estructura)	Método que resuelve la estructura ya cargada y restringida con sus condiciones de frontera. Determina el vector de desplazamientos nodales de la estructura y el vector de reacciones en los nodos restringidos por su tipo de vínculo.
setDatosEnNodos(estructura)	Método que fija los valores de desplazamientos nodales y vectores de reacción a los nodos de la estructura.
graficarFlecha(estructura)	Método que dibuja la superficie deformada de la estructura y la muestra por pantalla.
graficarGiroX(estructura)	Método que dibuja la superficie de la función de rotaciones del ángulo θ_x alrededor del eje Y de la estructura.
graficarGiroY(estructura)	Método que dibuja la superficie de la función de rotaciones del ángulo θ_y alrededor del eje X de la estructura.
graficarSigmaX(estructura,fibra)	Método que dibuja la distribución de tensiones en la dirección del eje X de la estructura en la fibra dada por parámetro.
graficarSigmaY(estructura,fibra)	Método que dibuja la distribución de tensiones en la dirección del eje Y de la estructura en la fibra dada por parámetro.
graficarTxy(estructura,fibra)	Método que dibuja la distribución de tensiones de cortante en la fibra dada por parámetro.
graficarMomentoX(estructura)	Método que dibuja la distribución de momentos flectores a lo largo del eje X, paralelos al eje Y.
graficarMomentoY(estructura)	Método que dibuja la distribución de momentos flectores a lo largo del eje Y, paralelos al eje X.
graficarMomentoXY(estructura)	Método que dibuja la distribución de momentos torsores en el elemento.

Para una información más detallada, el lector puede consultar el CD adjunto en la cual se encuentran todos los códigos utilizados en el desarrollo de la presente investigación junto con la explicación en cada uno de sus comentarios.

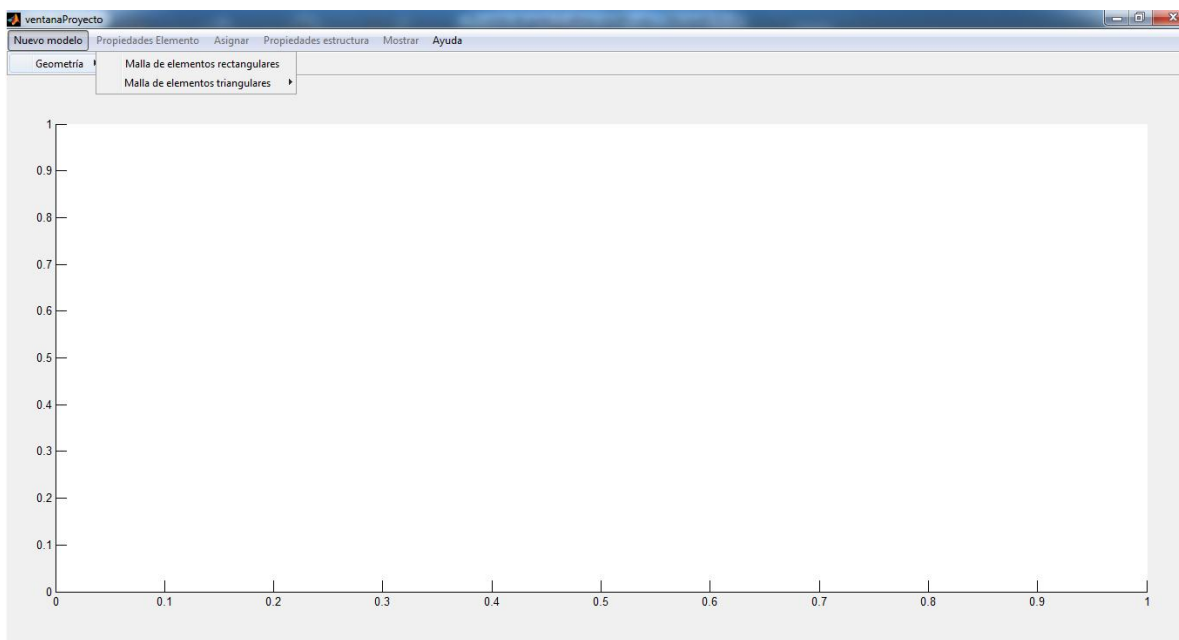
3.5.- Interfaz gráfica de usuario

El programa desarrollado cuenta con una interfaz gráfica principal y con 28 interfaces gráficas auxiliares, las cuales son llamadas con cada acción que se ejecute en el programa, a continuación vamos a ver las más importantes.

3.5.1.- Ventana principal

Es la ventana que aparece cuando se ejecuta el programa, en esta ventana se puede ver el sistema de ejes coordenados y la cinta de opciones con las acciones posibles a realizar por el programa. (Figura 19).

Imagen 19: Ventana principal del programa



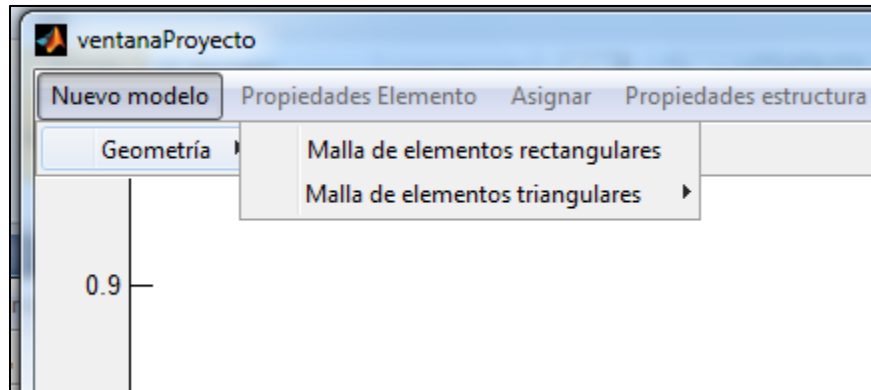
Se puede apreciar que cuenta con 6 elementos en la cinta de opciones, los cuales a su vez tienen cada uno, un submenú con diferentes funcionalidades.

En la gráfica se puede ver que las opciones: Propiedades elemento, Asignar, Propiedades estructura y Mostrar, se encuentran inactivas ya que por defecto el programa no tiene ningún modelo a ser analizado. Estas opciones se activan a medida que se van introduciendo datos al programa para su respectivo análisis de otra manera al no haber datos a ser analizados, se produciría errores de funcionamiento.

3.5.2.- Opción “Nuevo modelo”

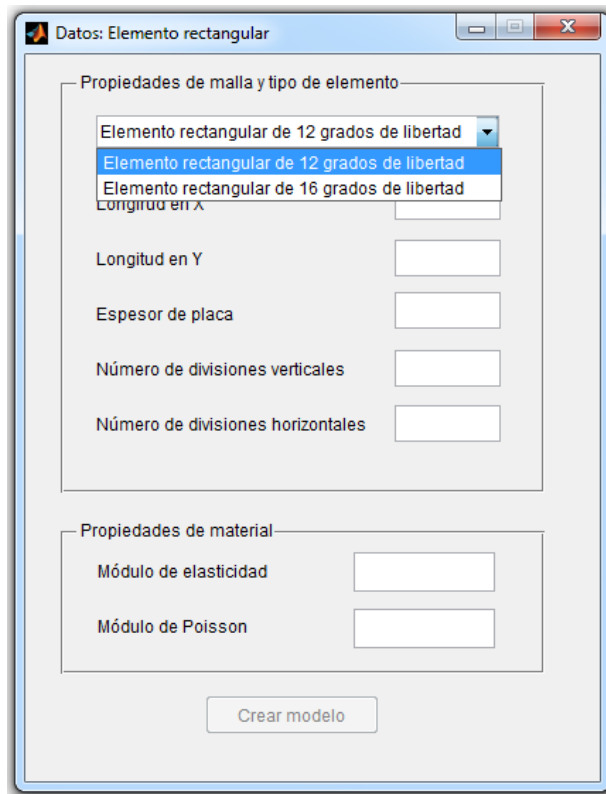
Al seleccionar esta opción aparece una pestaña que dice “Geometría” (Figura 20) en la cual hay dos opciones para generar un nuevo modelo los cuales veremos a continuación.

Imagen 20: Menú "Nuevo modelo"



Cuando escogemos la opción “Malla de elementos rectangulares”, nos aparece la siguiente ventana. (Figura 21)

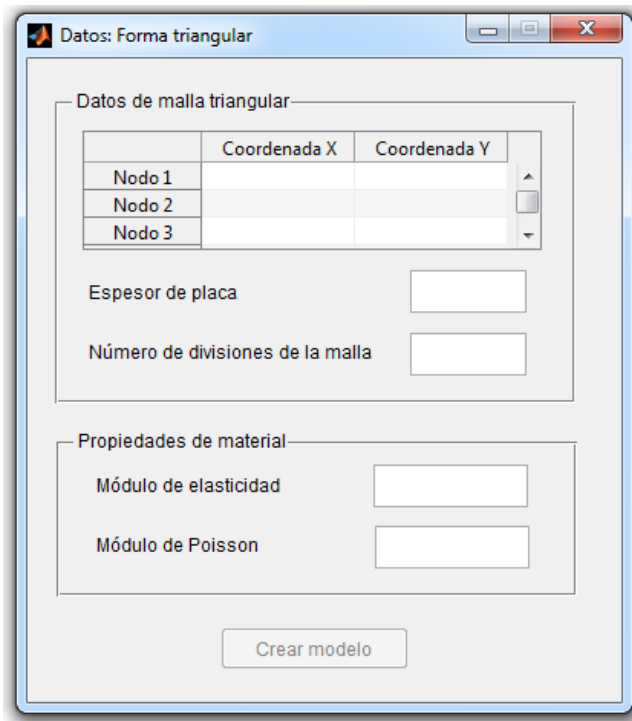
Imagen 21: ventana de creación de elementos rectangulares



Que nos muestra las opciones para crear una malla de elementos finitos rectangulares, ya sean de 12 o 16 grados de libertad, según sea el caso que deseemos estudiar.

Al elegir la opción “Malla de elementos triangulares” nos aparecerá lo siguiente:

Imagen 22: Ventana de creación de una malla de elementos finitos triangulares.

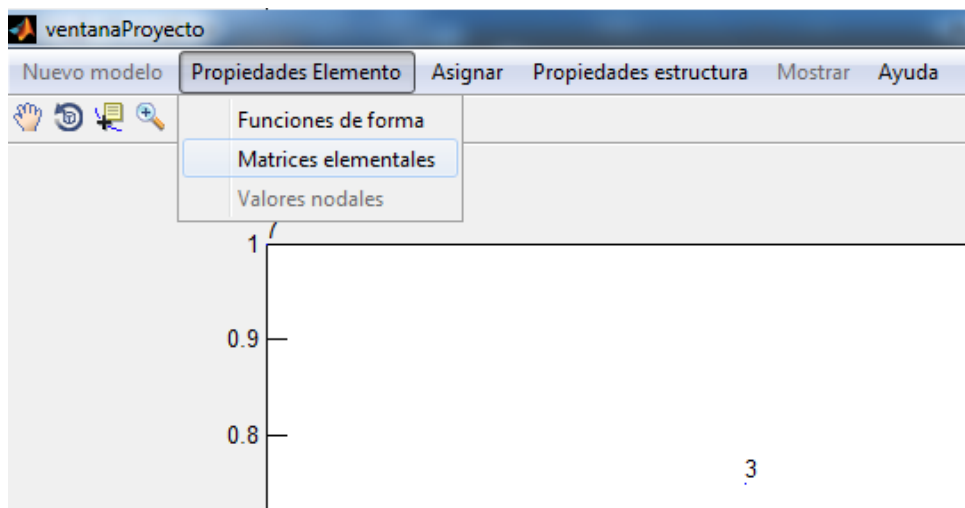


La cual nos permite definir una malla de elementos finitos triangulares a partir de las coordenadas de los vértices del elemento triangular.

3.5.3.- Opción “Propiedades elemento”

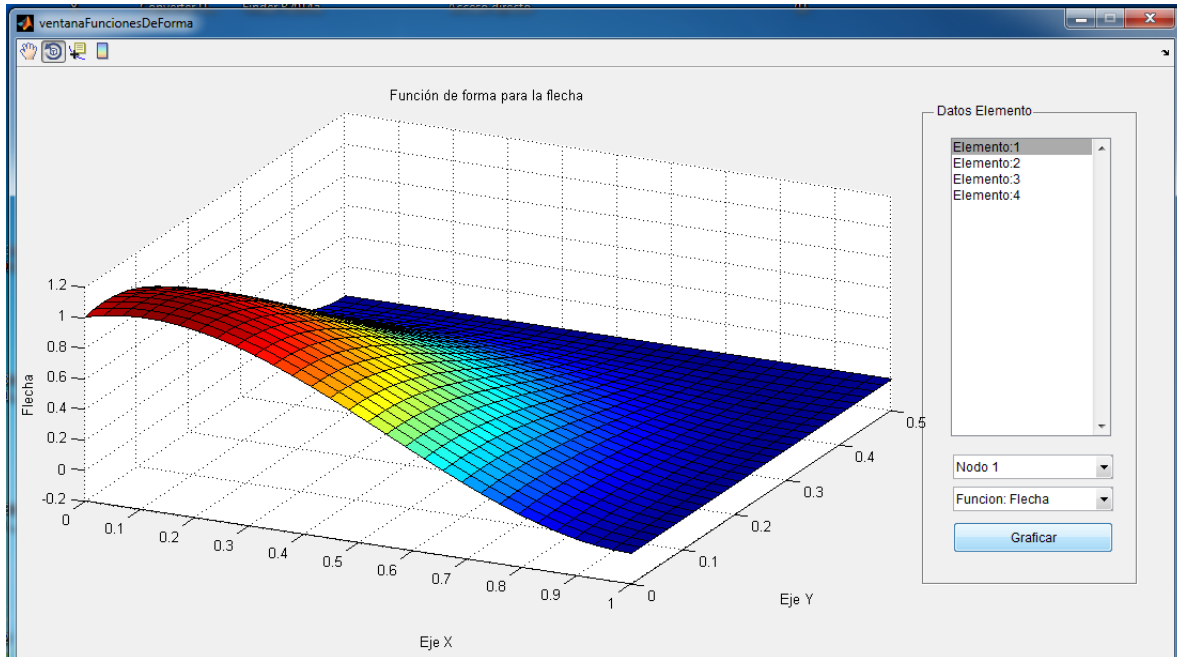
Una vez definido el tipo de elemento finito y la malla de elementos, se activa la opción “Propiedades elemento”, que cuenta con las siguientes propiedades:

Imagen 23: Opción "Propiedades elemento"



Al elegir la opción “Funciones de forma”, el programa muestra una ventana en la cual, el usuario puede ver las funciones de forma del elemento, según el número de nodo y la función de forma escogida para ser visualizada.

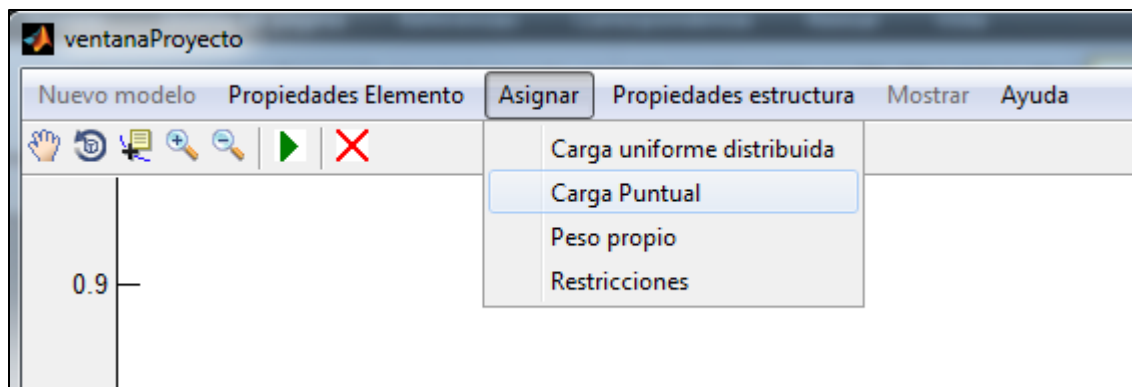
Imagen 24: Ventana "Funciones de forma"



3.5.4.- Opción “Asignar”

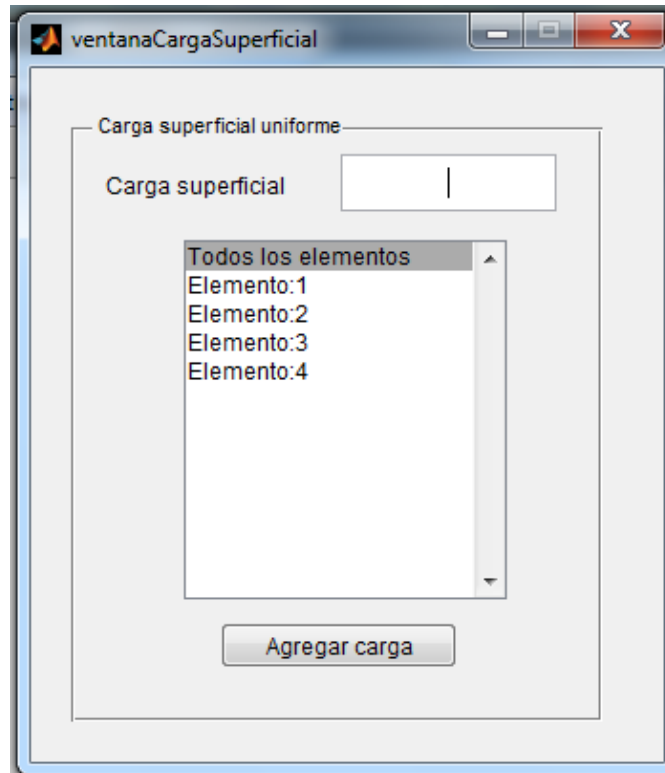
La opción “Asignar” nos permite añadir a la estructura las cargas y las diferentes condiciones de apoyo entre otros.

Imagen 25: Opción "Asignar"



Si escogemos la opción “Carga uniforme distribuida”, nos aparecerá la ventana mostrada a continuación para añadir el dato de una carga uniformemente distribuida.

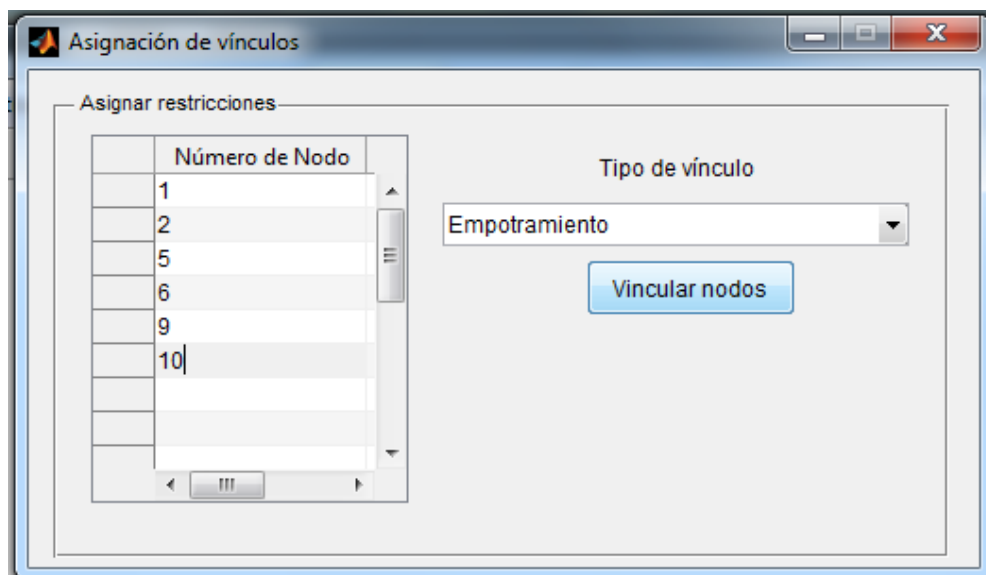
Imagen 26: Ventana para añadir una carga uniformemente distribuida



En la ventana se puede apreciar las opciones para cargar ya sea un determinado elemento o toda la estructura con la misma carga.

La opción “Restricciones” nos permite definir uno o varios nodos, con el tipo de vínculo seleccionado. (Figura 27)

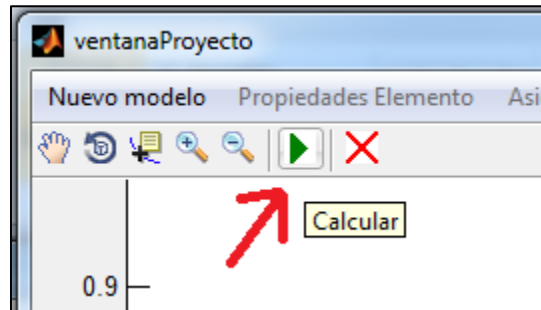
Imagen 27: Ventana para asignación de restricciones



3.5.5.- Opción “Calcular”

La opción “Calcular”, de la cinta de herramientas, se activa una vez hayamos asignado las cargas y los vínculos a los nodos de la estructura, es el botón mostrado a continuación (Figura 28)

Imagen 28: Herramienta de cálculo

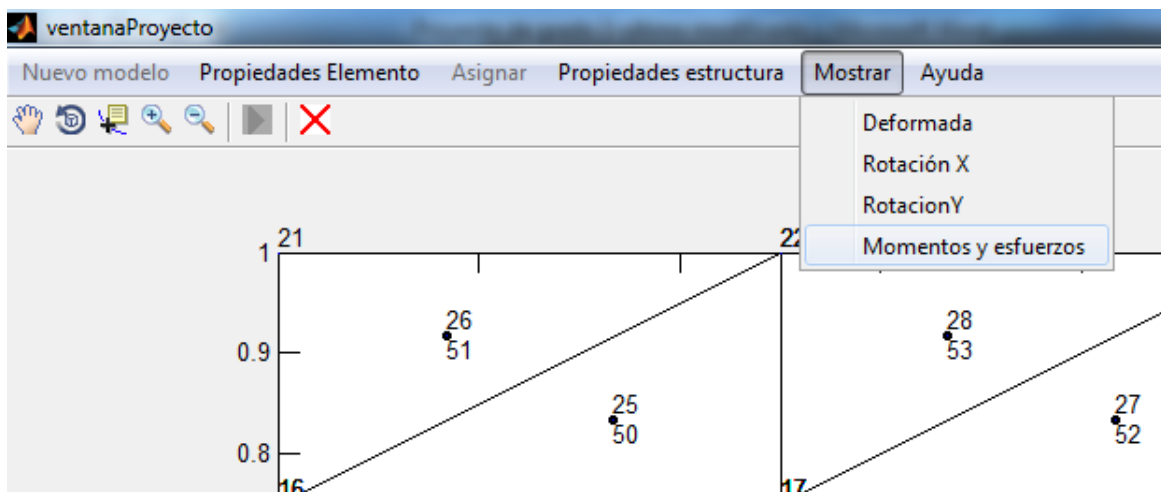


Una vez calculada la estructura, los desplazamientos nodales y vectores de reacción son asignados a los elementos y nodos de la estructura, con estos datos es posible graficar la estructura deformada y la distribución de tensiones y momentos flectores de la estructura.

3.5.6.- Opción “Mostrar”

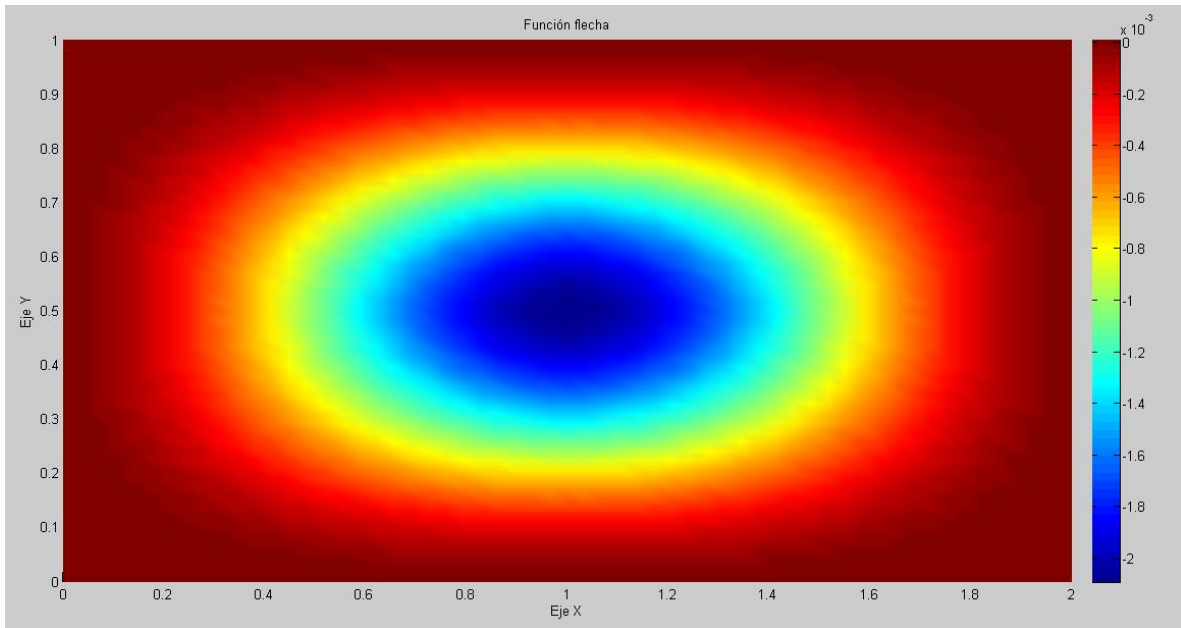
La opción “Mostrar”, nos permite elegir entre una serie de diferentes opciones de las cuales mostraremos las siguientes:

Imagen 29: Opción "Mostrar", con sus diferentes elementos



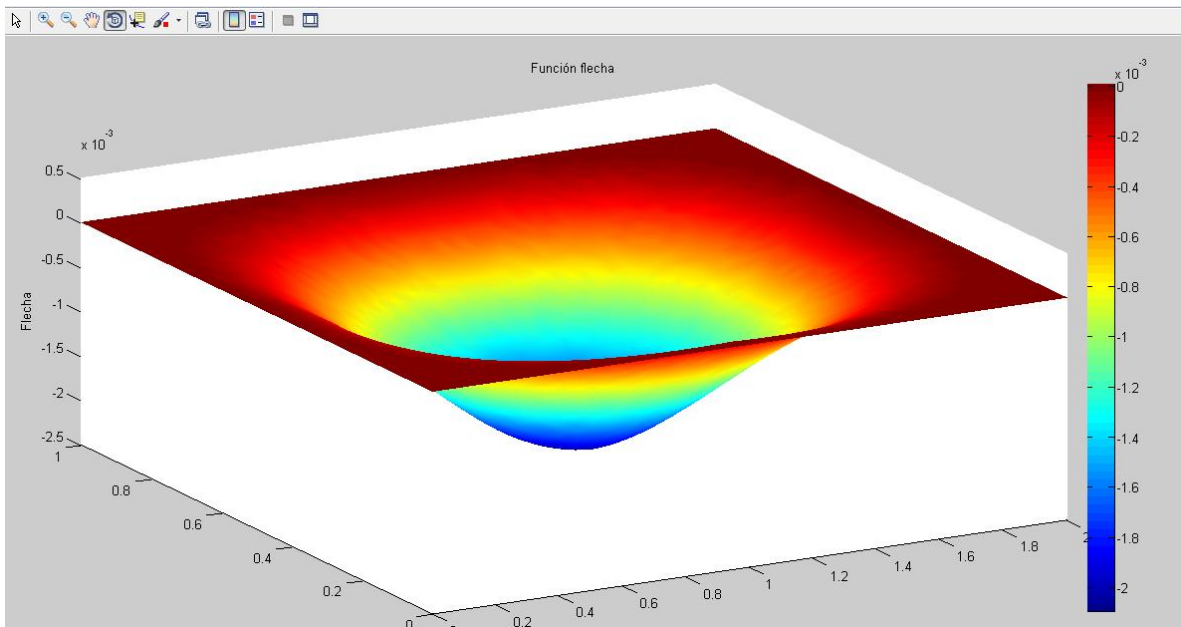
Si escogemos la opción “Deformada” de la paleta de opciones siguiendo el ejemplo hipotético, obtendríamos lo siguiente: (Figura 30)

Imagen 30: Mapa de calor de la flecha



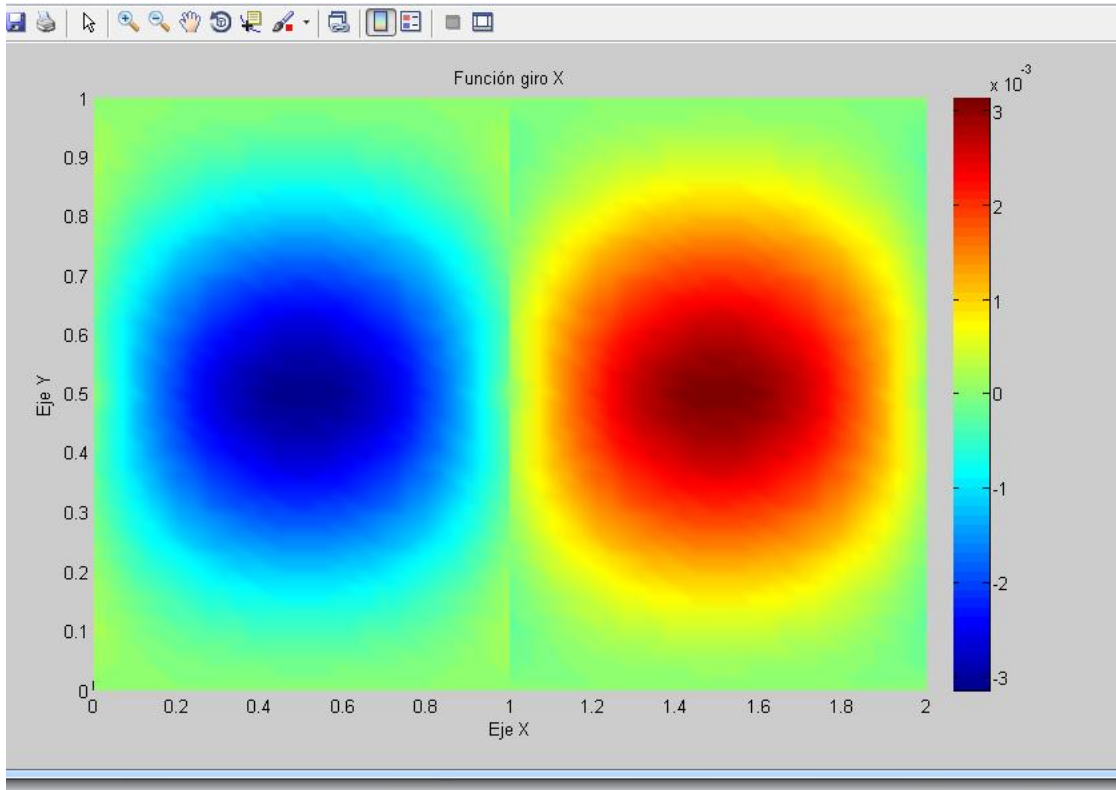
También es posible ver su deformada, haciendo click en la herramienta de rotación, obteniéndose lo siguiente:

Imagen 31: Deformada del ejemplo hipotético



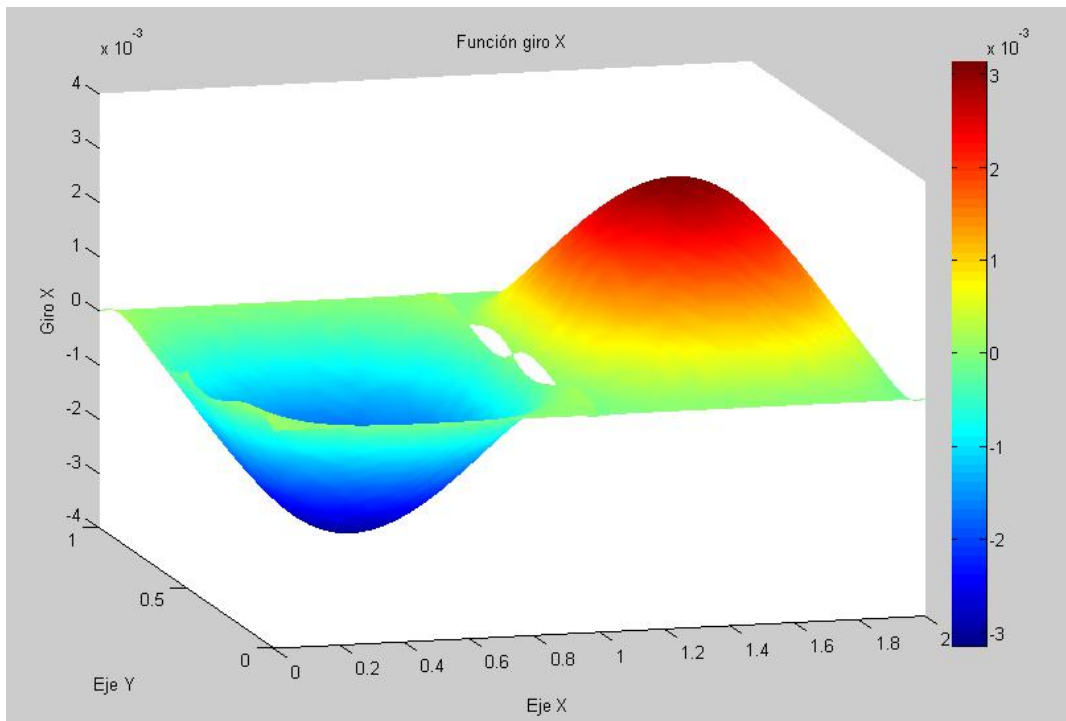
Escogiendo la opción "Rotación X", obtendríamos lo siguiente: (Figura 32)

Imagen 32: Mapa de calor de la Rotación en X (ejemplo hipotético)



Con la rotación 3D, se observa lo siguiente:

Imagen 33: Rotación en X, vista 3D



Al escoger la opción “Esfuerzos de flexión en X”, mostrada abajo se obtiene la imagen 35

Imagen 34: Opción "Mostrar Esfuerzos"

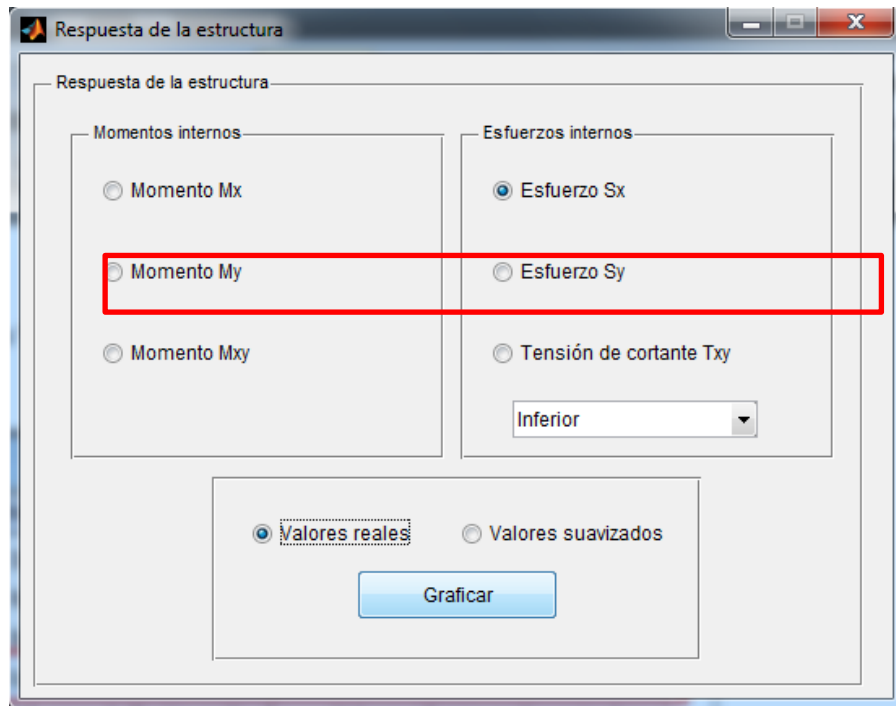


Imagen 35: Distribución de esfuerzos en X como mapa de calor

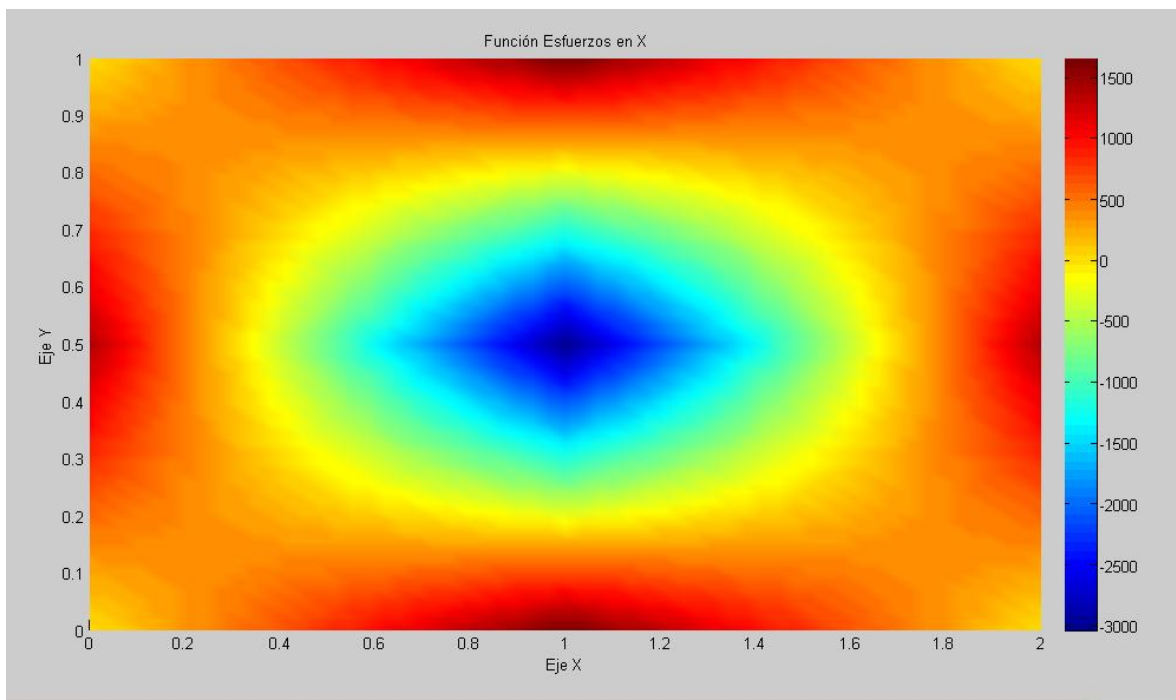
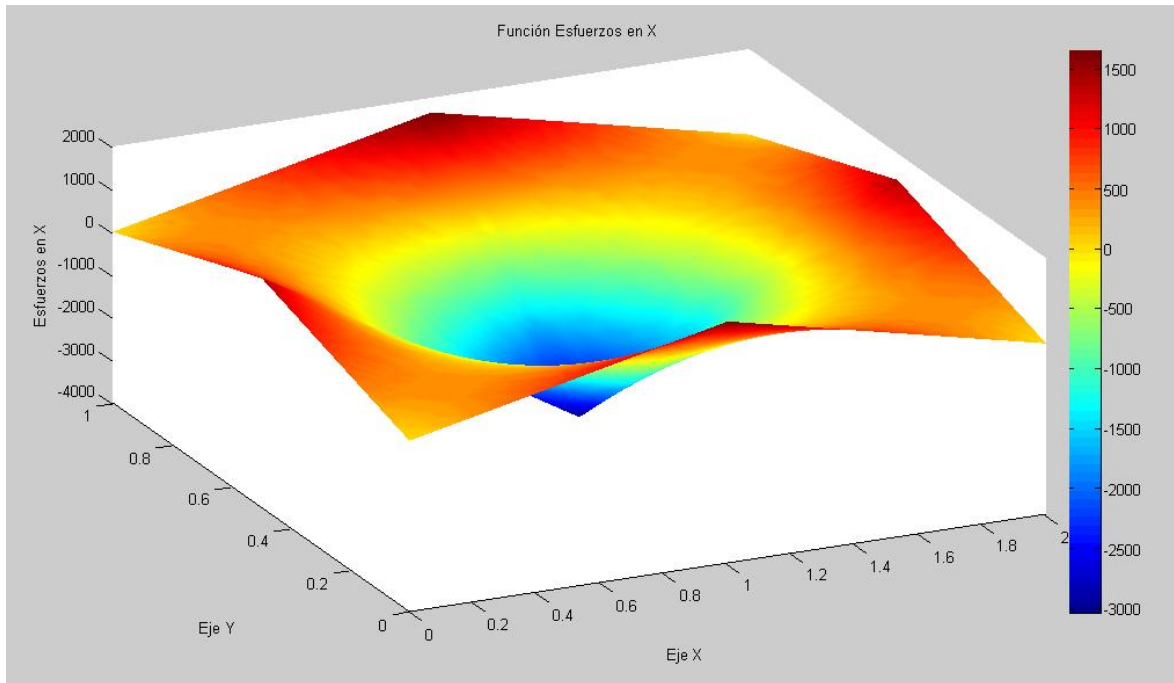


Imagen 36: Distribución de Esfuerzos en X, vista 3D



Para un mejor entendimiento, en el capítulo 4 se realizan ejemplos de demostración en algunos problemas de placas y se comparan resultados con los obtenidos por el programa comercial sap2000.

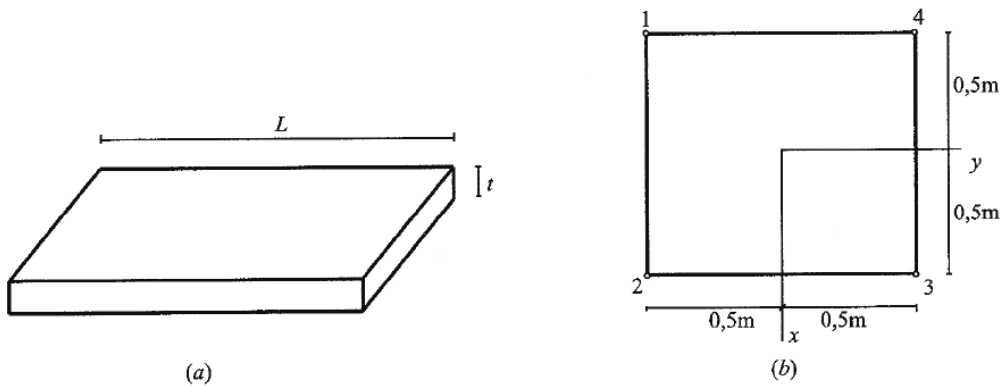
Para la validación del programa desarrollado y tener cierta fiabilidad de los resultados obtenidos, se realizó la simulación de con tres modelos de losa diferentes obtenidos del libro “El método de los elementos finitos aplicado al análisis estructural” citado en la bibliografía, y se comparó con los resultados obtenidos por el autor. También se resolvió los mismos problemas con el programa sap2000 versión 14. Los resultados obtenidos se muestran a continuación.

4.1.- Aplicación del programa desarrollado a modelos de elementos rectangulares

Ejemplo 1.- Obtención de la matriz de rigidez

Determinar la matriz de rigidez de un elemento de placa de hormigón lado L y espesor t .

Imagen 37: Ejemplo 1



Datos:

$L = 1\text{m}$

$t = 0.01\text{m}$

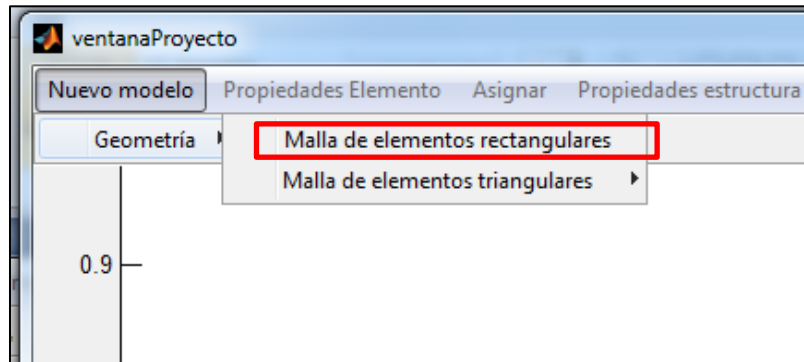
$E = 2\text{e}6 \text{ ton/m}^2$

$\mu = 0.2;$

Para obtener la matriz de rigidez con ayuda del programa, se procede de la siguiente manera:

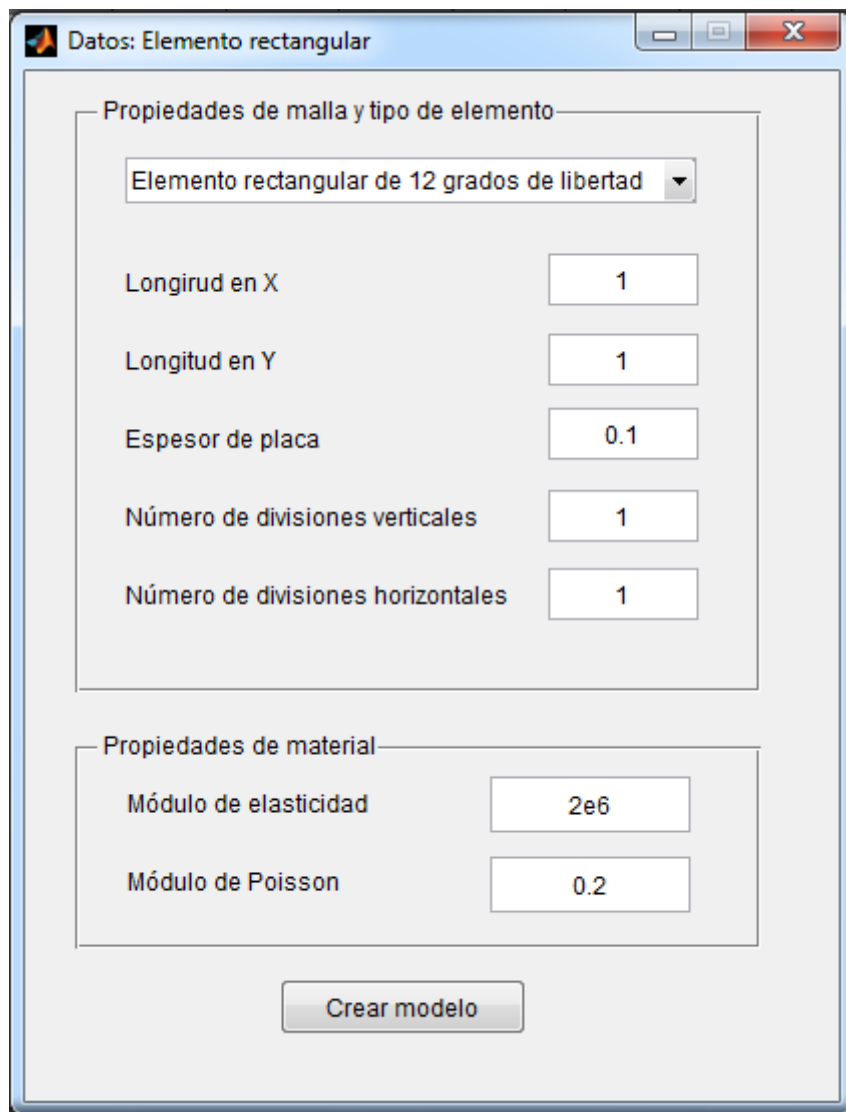
Abrimos el programa y escogemos la opción “Nuevo modelo”, tal como se muestra en la figura, ahí escogemos “Malla de elementos rectangulares”.

Imagen 38: Ejemplo 1, aplicación del programa para hallar la matriz de rigidez de un elemento



Una vez hecho esto se desplegará la siguiente ventana:

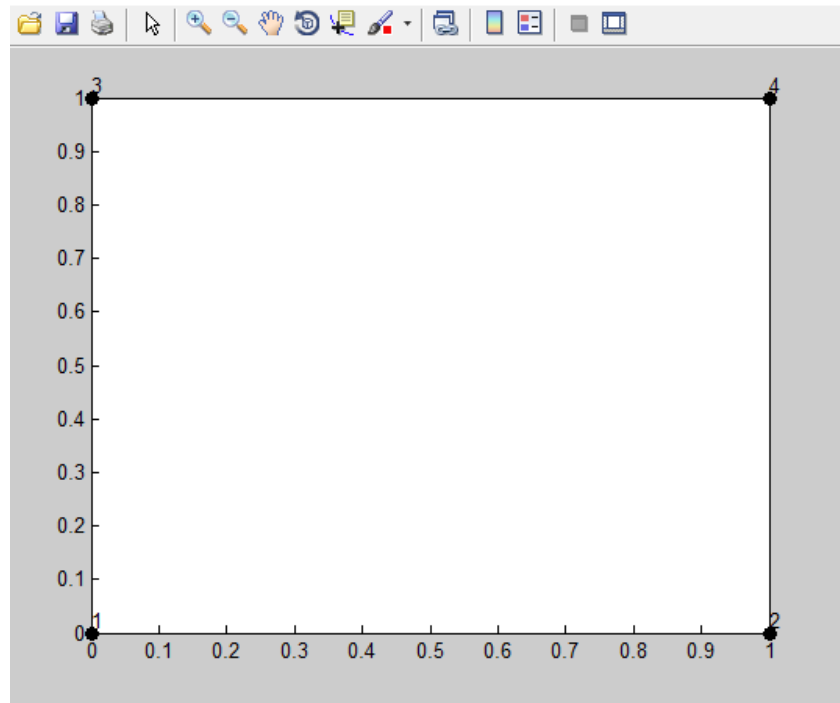
Imagen 39: Datos del ejemplo 1



En la cual colocamos los datos del enunciado y seleccionamos “Crear modelo”, en este caso por tratarse de un solo elemento, el número de divisiones verticales y horizontales es igual a 1.

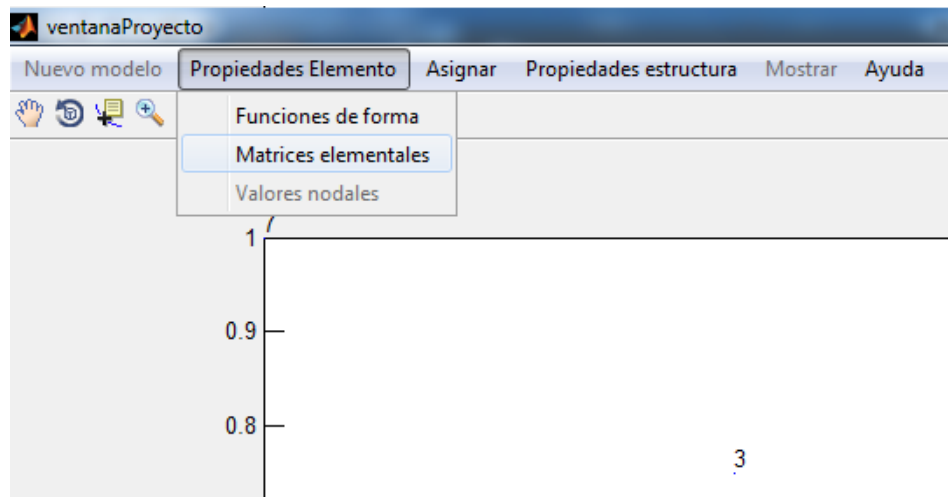
Una vez creado el modelo, se muestra la siguiente gráfica en la pantalla:

Imagen 40: Gráfico de elemento, ejemplo 1



Por solo tratarse de la obtención de la matriz de rigidez, basta con seleccionar la opción “Propiedades Elemento – Matrices elementales” para obtener la solución del problema:

Imagen 41: Propiedades del elemento, ejemplo 1



Para la matriz de rigidez, el autor del libro encuentra la siguiente solución:

$$\begin{bmatrix}
 1847,21 & & & & & & & & & & & & \\
 409,72 & 268,52 & & & & & & & & & & & \\
 409,72 & 34,72 & 268,52 & & & & & & & & & & \\
 -805,55 & -375,00 & 111,11 & 1847,21 & & & & & & & & & \\
 375,00 & 106,48 & 0 & -409,72 & 268,52 & & & & & & & & \\
 111,11 & 0 & 78,70 & 409,72 & -34,72 & 268,52 & & & & & & & \\
 -236,11 & -145,83 & -145,83 & -805,55 & -111,11 & -375,00 & 1847,21 & & & & & & \\
 145,83 & 67,13 & 0 & -111,11 & 78,70 & 0 & -409,72 & 268,52 & & & & & \\
 145,83 & 0 & 67,13 & 375,0 & 0 & 106,48 & -409,72 & 34,72 & 268,52 & & & & \\
 -805,55 & 111,11 & -375,00 & -236,11 & 145,83 & -145,83 & -805,55 & 375,00 & -111,11 & 1847,21 & & & \\
 111,11 & 78,70 & 0 & -145,83 & 67,13 & 67,13 & -375,00 & 106,48 & 0 & 409,72 & 268,52 & & \\
 375,00 & 0 & 106,48 & 145,83 & 0 & 0 & -111,11 & 0 & 78,70 & -409,72 & -34,72 & 268,52 & \\
 \end{bmatrix}$$

simétrica

Por un error tipográfico, en el libro aparece H en lugar de K (página 288), pero este es el resultado de la matriz de rigidez.

Por otra parte, con el programa desarrollado en MatLab se obtiene lo siguiente:

Imagen 42: Solución obtenida por MatLab del ejemplo 1

```

ans =
1.0e+03 *
 1.8472  0.4097  0.4097  -0.8056  0.3750  0.1111  -0.2361  0.1458  0.1458  -0.8056  0.1111  0.3750
 0.4097  0.2685  0.0347  -0.3750  0.1065  -0.0000  -0.1458  0.0671  -0.0000  0.1111  0.0787  -0.0000
 0.4097  0.0347  0.2685  0.1111  -0.0000  0.0787  -0.1458  -0.0000  0.0671  -0.3750  -0.0000  0.1065
 -0.8056  -0.3750  0.1111  1.8472  -0.4097  0.4097  -0.8056  -0.1111  0.3750  -0.2361  -0.1458  0.1458
 0.3750  0.1065  -0.0000  -0.4097  0.2685  -0.0347  -0.1111  0.0787  -0.0000  0.1458  0.0671  -0.0000
 0.1111  0.0000  0.0787  0.4097  -0.0347  0.2685  -0.3750  0  0.1065  -0.1458  0  0.0671
 -0.2361  -0.1458  -0.1458  -0.8056  -0.1111  -0.3750  1.8472  -0.4097  -0.4097  -0.8056  -0.3750  -0.1111
 0.1458  0.0671  0  -0.1111  0.0787  0  -0.4097  0.2685  0.0347  0.3750  0.1065  0
 0.1458  0  0.0671  0.3750  0  0.1065  -0.4097  0.0347  0.2685  -0.1111  0  0.0787
 -0.8056  0.1111  -0.3750  -0.2361  0.1458  -0.1458  -0.8056  0.3750  -0.1111  1.8472  0.4097  -0.4097
 0.1111  0.0787  0.0000  -0.1458  0.0671  0  -0.3750  0.1065  0  0.4097  0.2685  -0.0347
 0.3750  -0.0000  0.1065  0.1458  -0.0000  0.0671  -0.1111  -0.0000  0.0787  -0.4097  -0.0347  0.2685

```

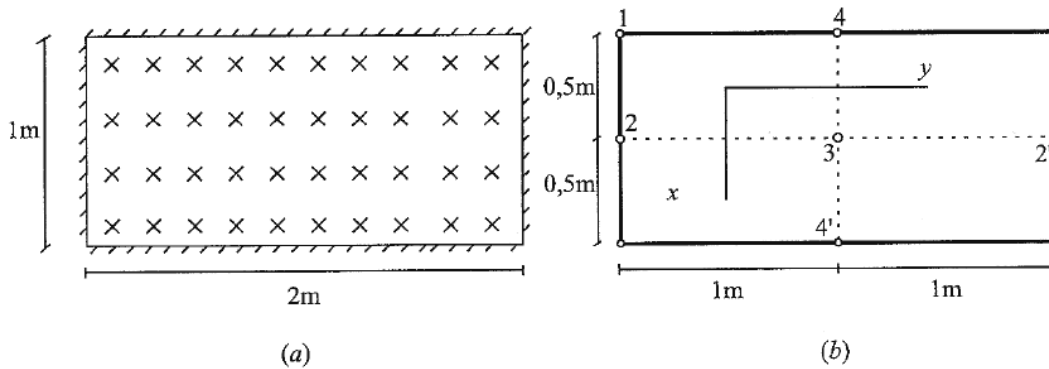
Es fácil ver que los resultados coinciden.

En este caso no se puede comparar los resultados obtenidos con el programa Sap2000, debido a que este no ofrece un acceso simple a la matriz de rigidez del elemento, pero para este ejemplo es suficiente con ver que la solución obtenida por MatLab, coincide con la solución dada por el autor.

Ejemplo 2.- Cálculo de flecha máxima

Una placa rectangular de acero de espesor t está empotrada en sus bordes y sometida a una carga uniformemente distribuida q (Imagen 43). Dividiendo la placa en una malla de 2×2 elementos, determinar su flecha máxima.

Imagen 43: Ejemplo de aplicación 2



Datos:

$$q = 1.2 \text{ ton/m}^2; \quad E = 2e7 \text{ ton/m}^2; \quad t = 0.01 \text{ m}; \quad \mu = 0.3$$

Nuevamente abrimos el programa y escogemos la opción “Nuevo modelo”, escogemos la opción para generar una malla rectangular de elementos e introducimos los datos:

Imagen 44: Datos del ejemplo 2

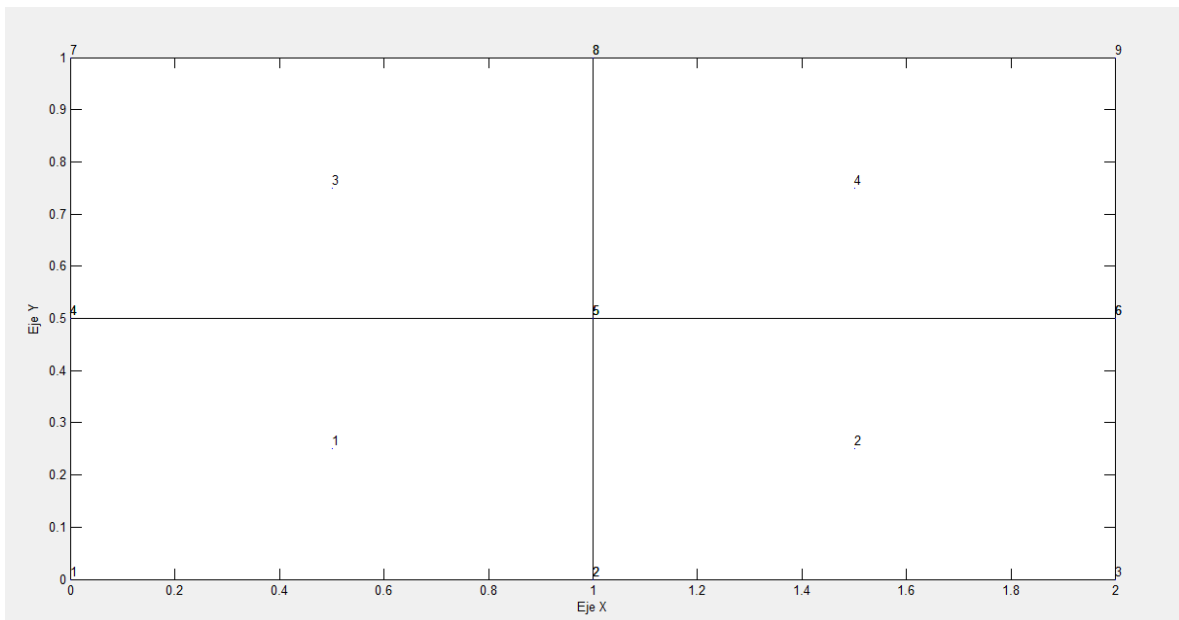
La interfaz de usuario muestra un formulario con los siguientes campos:

- Propiedades de malla y tipo de elemento:
 - Elemento rectangular de 12 grados de libertad (seleccionado)
 - Longitud en X: 2
 - Longitud en Y: 1
 - Espesor de placa: 0.01
 - Número de divisiones verticales: 2
 - Número de divisiones horizontales: 2
- Propiedades de material:
 - Módulo de elasticidad: 2e7
 - Módulo de Poisson: 0.3

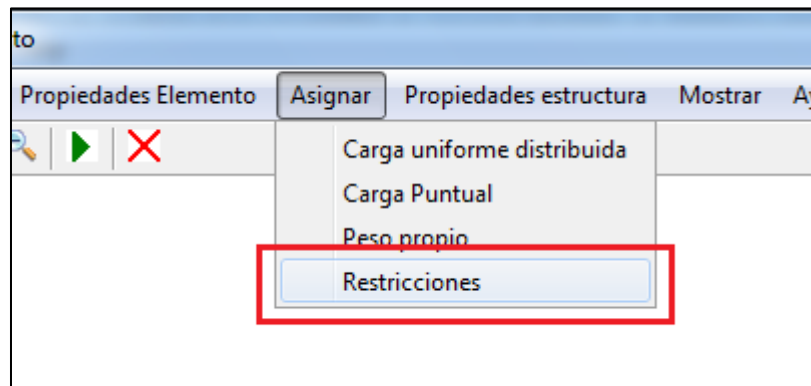
Botón: Crear modelo

Con esto creamos la malla de elementos obteniendo el siguiente gráfico para la malla:

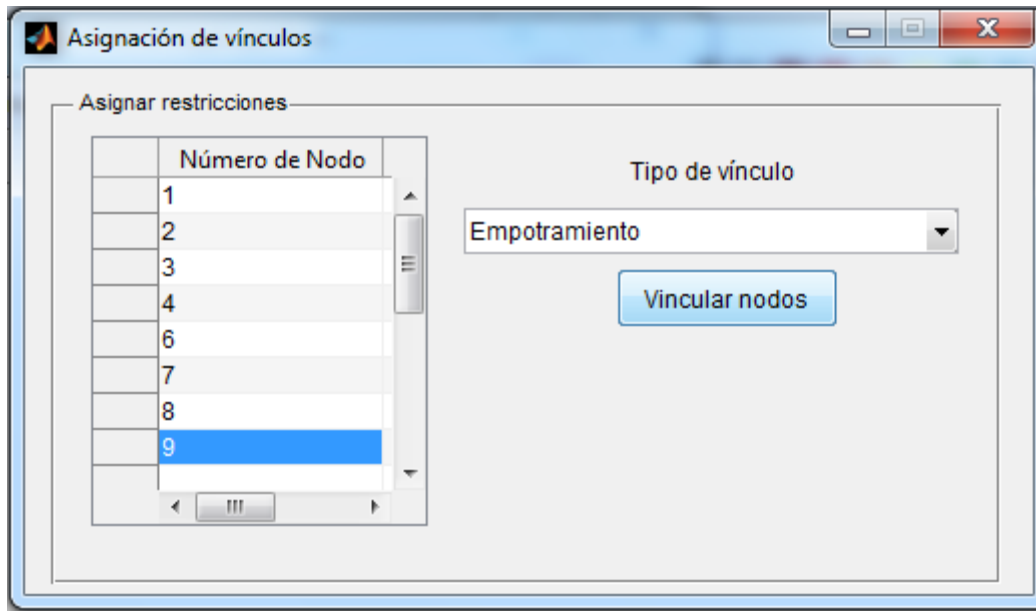
Imagen 45: Malla de elementos del ejercicio 2



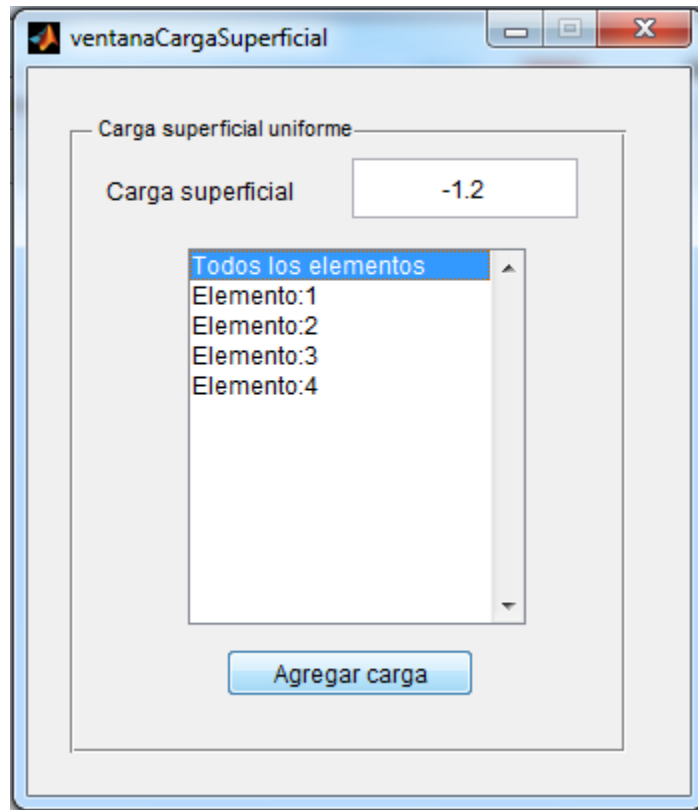
Una vez determinada la malla de elementos se procede a vincular la estructura, para esto vamos a la cinta de menú y escogemos la opción “Asignar”, donde asignamos el tipo de vínculo y los nodos a ser vinculados.



Y escogemos la opción “Empotrado” en los nodos: 1, 2, 3, 4, 6, 7, 8, 9



Una vez vinculados los nodos de la estructura procedemos a añadir la carga uniforme distribuida sobre toda la placa, para esto vamos al menú asignar y seleccionamos la opción “Carga uniforme distribuida” y agregamos el valor -1.2 ton/m^2 a toda la estructura, en nuestro caso el valor de la carga debe ser negativo, ya que el convenio de signos adoptado para el desarrollo del programa es que las fuerzas verticales hacia arriba son positivas y la fuerzas verticales hacia abajo son negativas. Por lo tanto tenemos lo siguiente en nuestra ventana de datos:



Con las cargas ya puestas en la estructura y los vínculos definidos, se procede a calcular los desplazamientos nodales, en este caso el único nodo desplazable es el nodo 5. Al hacer correr el programa se obtiene de manera automática la deformada de la estructura.

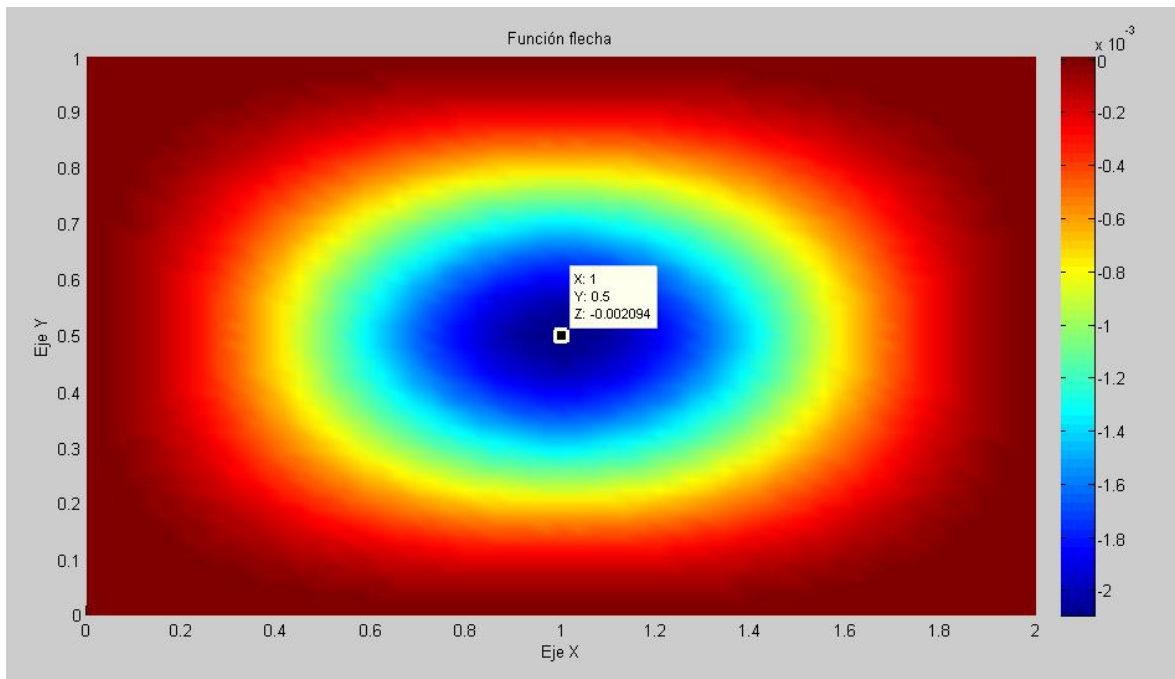
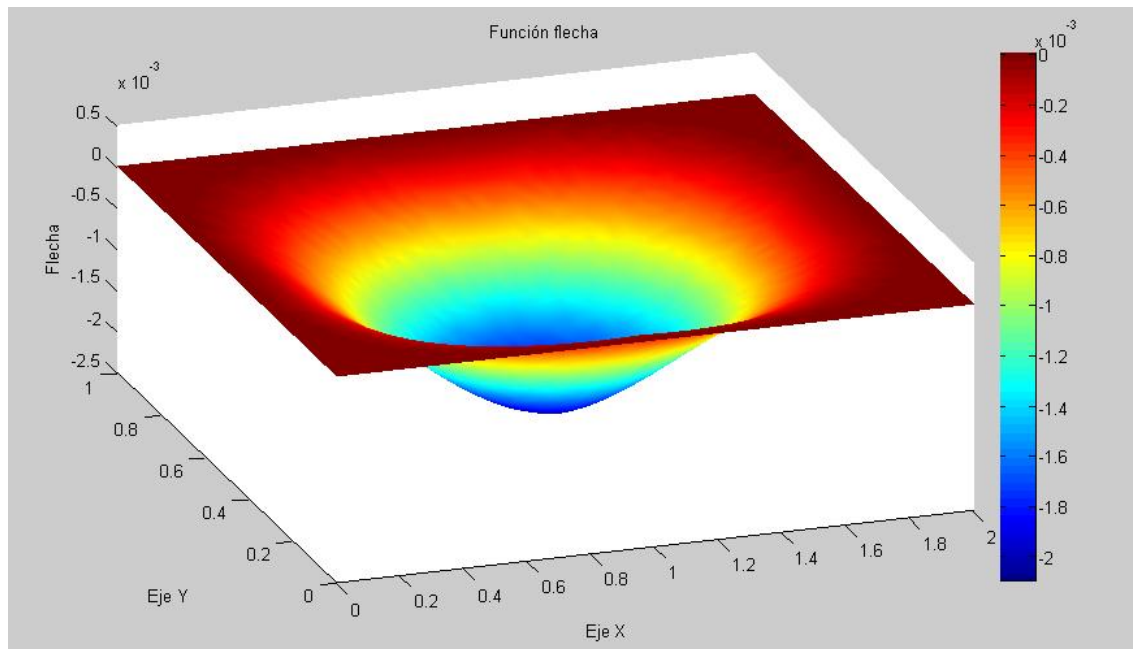


Imagen 46: Vista 3D de la deformada del ejemplo 2



Podemos ver con el “data cursor” el valor de la flecha en el centro de la placa, que para este caso es el valor de la máxima flecha presentada en la estructura. El valor de la máxima flecha hallada con el programa es:

$$w = 0.00209\text{m}$$

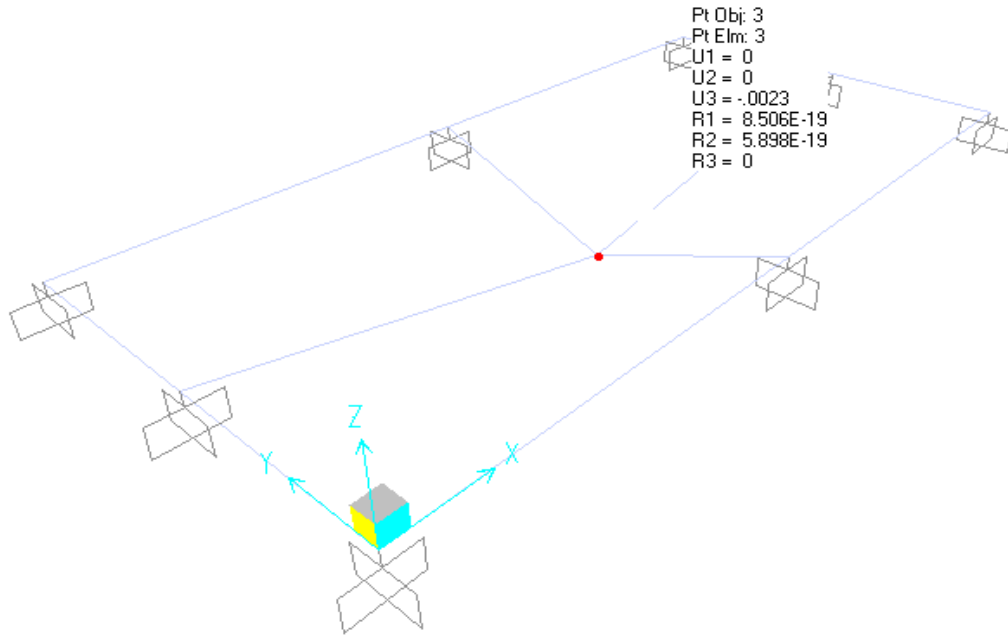
Que comparada con la solución dada por el autor del libro, que es de: $w = 0.209\text{cm}$, que es prácticamente el mismo resultado.

Resolviendo el problema con el programa Sap2000 se obtiene lo siguiente para la deformada (Imagen 47):

Donde podemos ver que el valor de la flecha en el nodo central es de:

$$w = 0.0023\text{m}$$

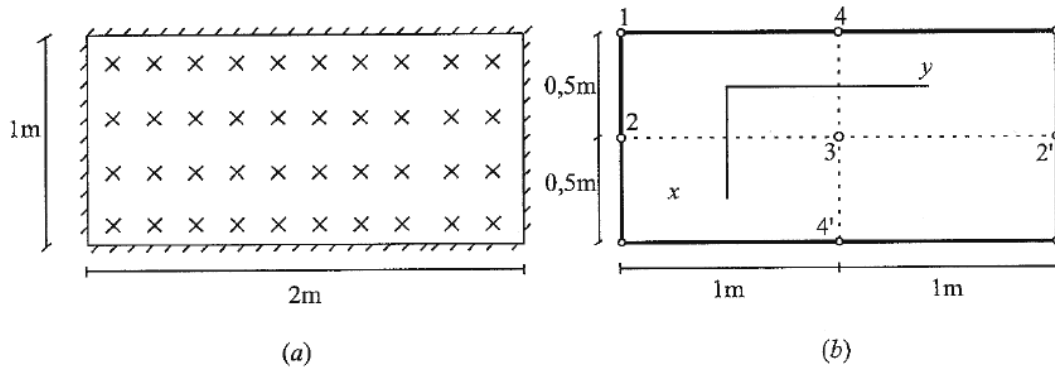
Imagen 47: Deformada obtenida mediante Sap2000



La diferencia entre los resultados hallados por el autor del libro, el programa desarrollado en MatLab y el programa sap2000 puede ser debido a que el programa sap2000 realiza una interpolación lineal de deformaciones, con otras funciones de forma diferentes a las escogidas para el desarrollo de la herramienta en MatLab. Pese a esto la diferencia entre resultados para este caso es de 0.2mm, lo cual constituye una buena aproximación por parte del programa desarrollado.

Ejemplo 3.- Verificación de convergencia

Con los datos del ejemplo anterior se verifica el grado de error cometido y se evalúa el grado de convergencia a la solución mediante el método de los elementos finitos.



Datos:

$$q = 1.2 \text{ ton/m}^2; \quad E = 2e7 \text{ ton/m}^2; \quad t = 0.01 \text{ m}; \quad \mu = 0.3$$

Para evaluar la velocidad de convergencia a la solución considerada exacta, se divide la placa en una malla de elementos de 2x2, 4x4, 6x6 y 8x8, luego se determina la flecha en el centro de la placa y se realiza una comparación entre los resultados obtenidos para las diferentes mallas.

Para la placa en estudio la flecha máxima se determina por la fórmula:

$$w_{max} = -0.00254 \frac{q(2a)^4}{D} \quad (4.1)$$

Dónde:

q = Carga uniformemente distribuida (ton/m²)

a = Semi ancho de placa (m)

$$D = Et^3/12(1-\mu^2);$$

Reemplazando datos se obtiene:

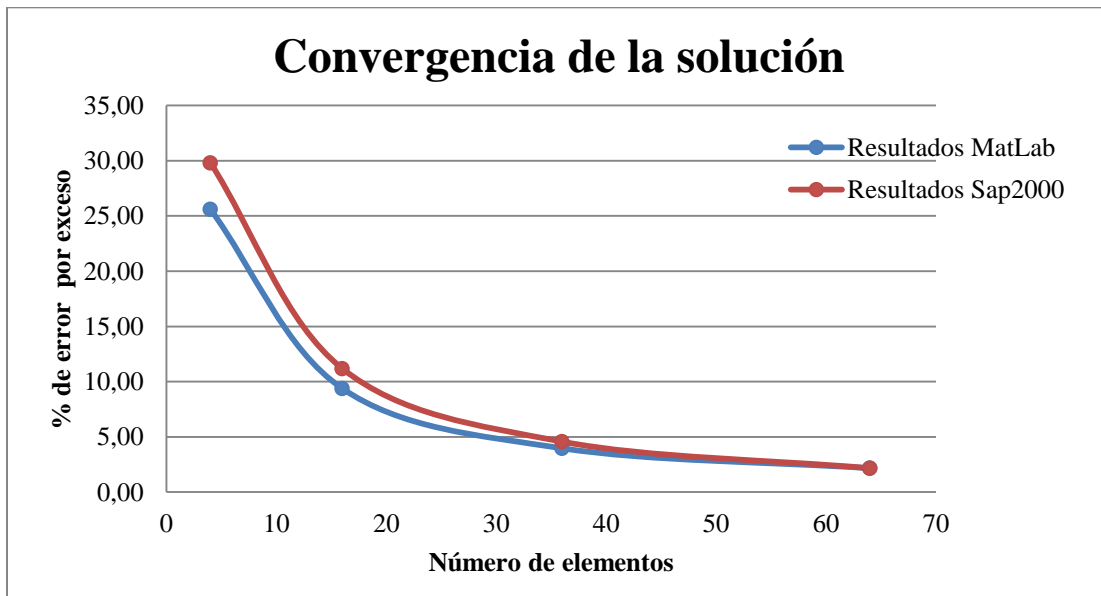
$$w_{max} = -0.00254 \frac{1.2 * (2 * 0.5)^4 * 12 * (1 - 0.3^2)}{20000000 * 0.01^3} = 0.001664 \text{ m}$$

A continuación se presenta una tabla con los valores de flecha hallados en el centro de la placa, para las diferentes divisiones de malla:

Tabla 9: Comparación de convergencia a la solución entre Sap2000 y el programa desarrollado en MatLab

N° elementos	Malla	MatLab			Sap2000		
		Aproximada	Exacta	% error	Aproximada	Exacta	% error
4	2x2	-0.00209	-0.001664	25.60	-0.00216	-0.001664	29.81
16	4x4	-0.00182	-0.001664	9.38	-0.00185	-0.001664	11.18
36	6x6	-0.00173	-0.001664	3.97	-0.00174	-0.001664	4.57
64	8x8	-0.0017	-0.001664	2.16	-0.0017	-0.001664	2.16

Gráfico 1: Comparación de convergencia a la solución de los programas sap2000 y el desarrollado en MatLab



De los anteriores datos se puede apreciar que el programa desarrollado en MatLab converge más rápido a la solución exacta que el programa sap2000, esto puede deberse a que el programa sap2000 realiza una interpolación lineal de deformaciones con funciones de forma de menor grado que las escogidas para el desarrollo del programa en MatLab. Para el desarrollo del presente programa se utilizó polinomios incompletos de cuarto grado, tal como lo sugiere la teoría de placas planas a flexión, esto no quiere decir que el programa sap2000 no brinde resultados precisos, sino que es necesario un mayor número de elementos finitos para acercarse más a la solución considerada como exacta.

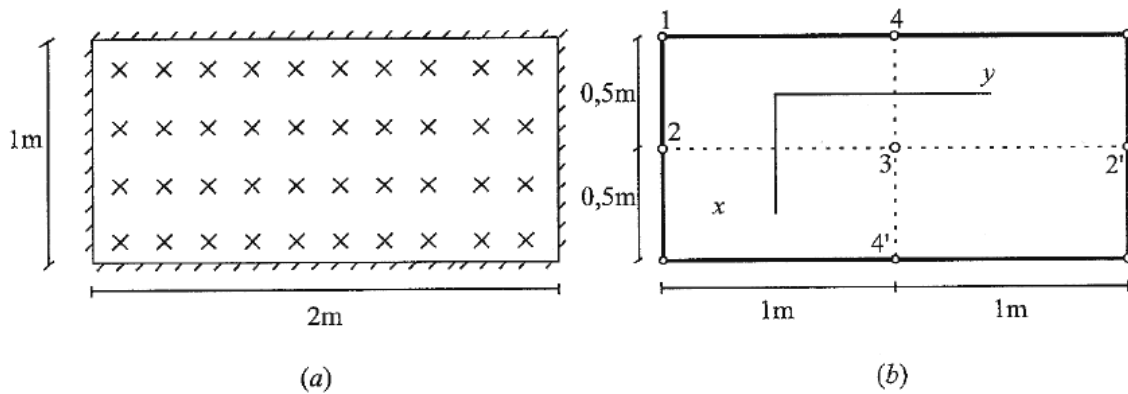
Ejemplo 4.- Comparación de convergencia entre el elemento rectangular de 12 grados de libertad y el elemento rectangular de 16 grados de libertad

En este apartado se procede a comparar la convergencia a la solución exacta entre el elemento rectangular de 12GDL y el elemento rectangular de 16GDL, para esto vamos a continuar con los datos del ejemplo anterior y se analiza el modelo con los diferentes tipo de elemento y para los tamaños de malla: 2x2, 4x4, 6x6 y 8x8.

Datos:

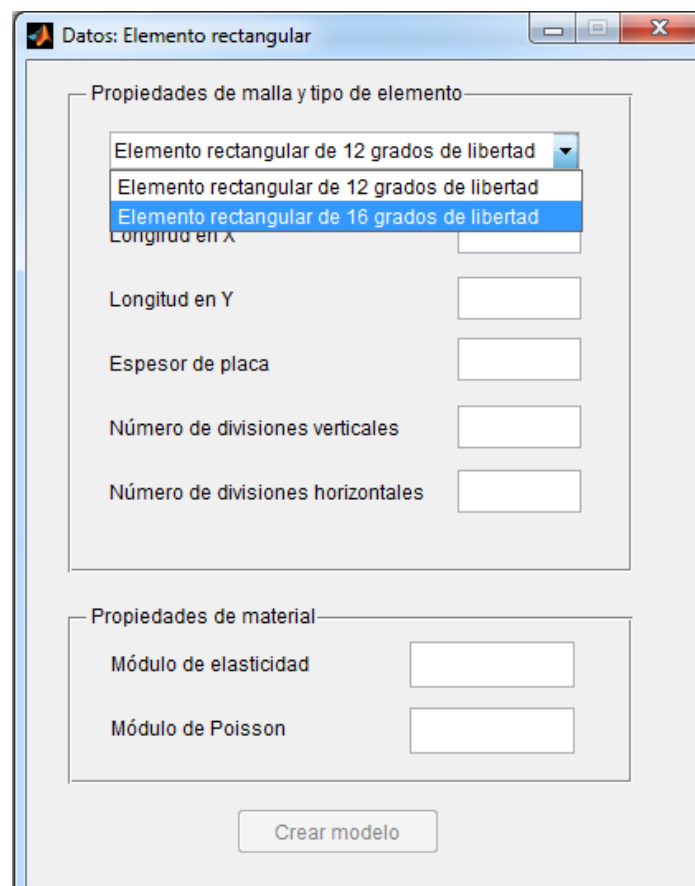
$$q = 1.2 \text{ ton/m}^2; \quad E = 2e7 \text{ ton/m}^2; \quad t = 0.01 \text{ m}; \quad \mu = 0.3$$

El modelo a ser resuelto se muestra a continuación:



Para el cálculo de la placa con elementos de 16 grados de libertad, el procedimiento es el mismo que el seguido para elementos de 12 grados de libertad, con la diferencia de que en la ventana para crear el nuevo modelo, se selecciona la opción de “Elemento rectangular de 16 grados de libertad”

Imagen 48: Creación de modelo de elementos rectangulares de 16 grados de libertad



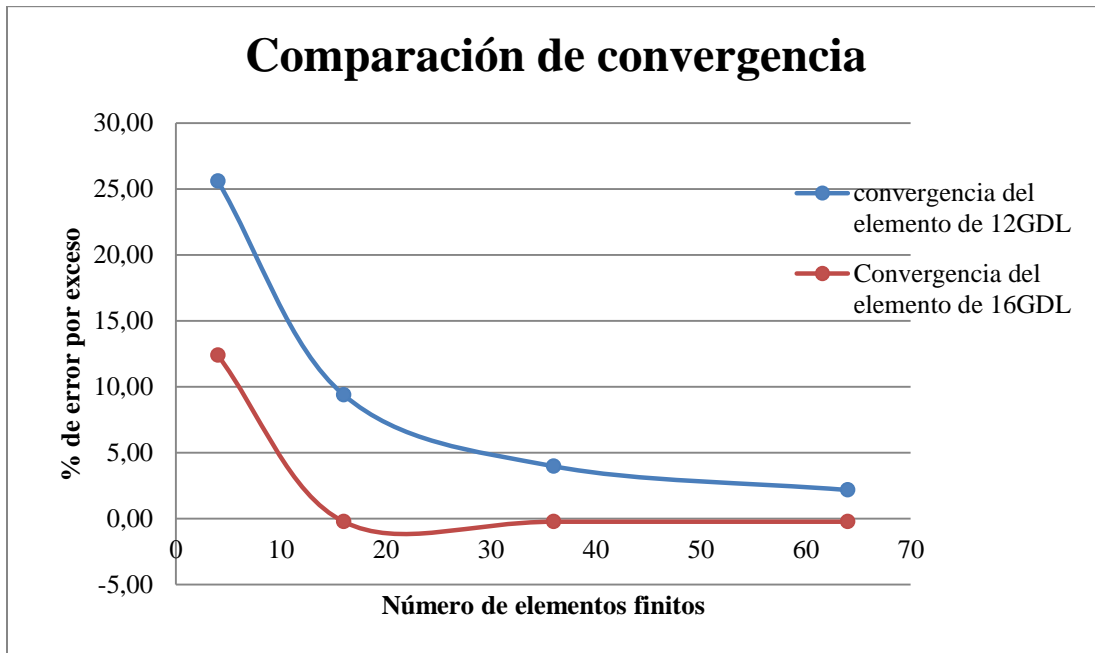
El resto de los datos se introduce de igual manera y se obtienen los resultados que se muestran a continuación:

Tabla 10: Comparación de convergencia entre el elemento de 12GDL y el elemento de 16GDL

N° elementos	Malla	Elemento de 12 grados de libertad			Elemento de 16 grados de libertad		
		Aproximada	Exacta	% error	Aproximada	Exacta	% error
4	2x2	-0.00209	-0.001664	25.60	-0.00187	-0.001664	12.38
16	4x4	-0.00182	-0.001664	9.38	-0.00166	-0.001664	-0.24
36	6x6	-0.00173	-0.001664	3.97	-0.00166	-0.001664	-0.24
64	8x8	-0.0017	-0.001664	2.16	-0.00166	-0.001664	-0.24

Con estos datos se realiza el siguiente cuadro comparativo:

Gráfico 2: Comparación de convergencia del elemento de 12GDL y el elemento de 16GDL



Se puede apreciar que el elemento rectangular de 16 grados de libertad converge a la solución mucho más rápido que el elemento de 12 grados de libertad, esto se debe a que en la formulación del elemento de 16GDL, se utilizan polinomios incompletos de sexto grado, lo cual garantiza una mejor aproximación a la solución exacta.

A continuación se hace una aplicación del elemento triangular de 10 grados de libertad y se validan los resultados obtenidos con ayuda del programa sap2000.

4.2.- Aplicación del programa a problemas con elementos triangulares

En la siguiente sección se hace una demostración del elemento triangular de 10 grados de libertad desarrollado en MatLab y se validan los resultados con ayuda del programa comercial sap2000 versión 14. También se realizará la comparación entre la comparación de convergencia a la solución entre el programa desarrollado en MatLab y el programa comercial sap2000.

Ejemplo 5.- Aplicación del programa para la obtención de la matriz de rigidez de una placa triangular.

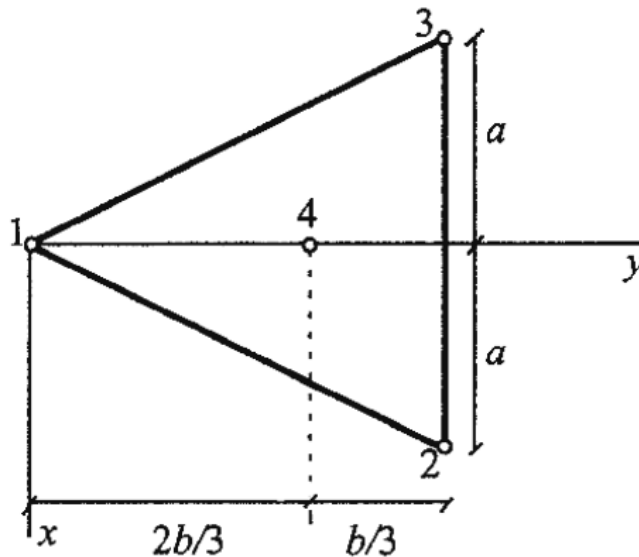
El siguiente problema se tomó del libro citado en la bibliografía “El método de los elementos finitos aplicado al análisis estructural” página 309, cuyo enunciado es el siguiente:

Determinar la matriz de rigidez de un elemento de placa triangular, de acero con espesor t ,

Datos:

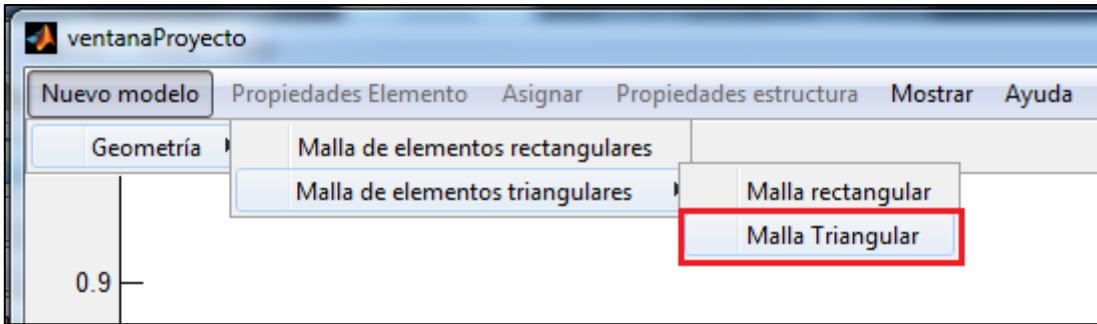
$$a = 0.5\text{m}; \quad b = 1\text{m}; \quad t = 0.01\text{m}; \quad E = 2e7 \text{ ton/m}^2; \quad \mu = 0.3;$$

Imagen 49: Ejemplo de elemento triangular

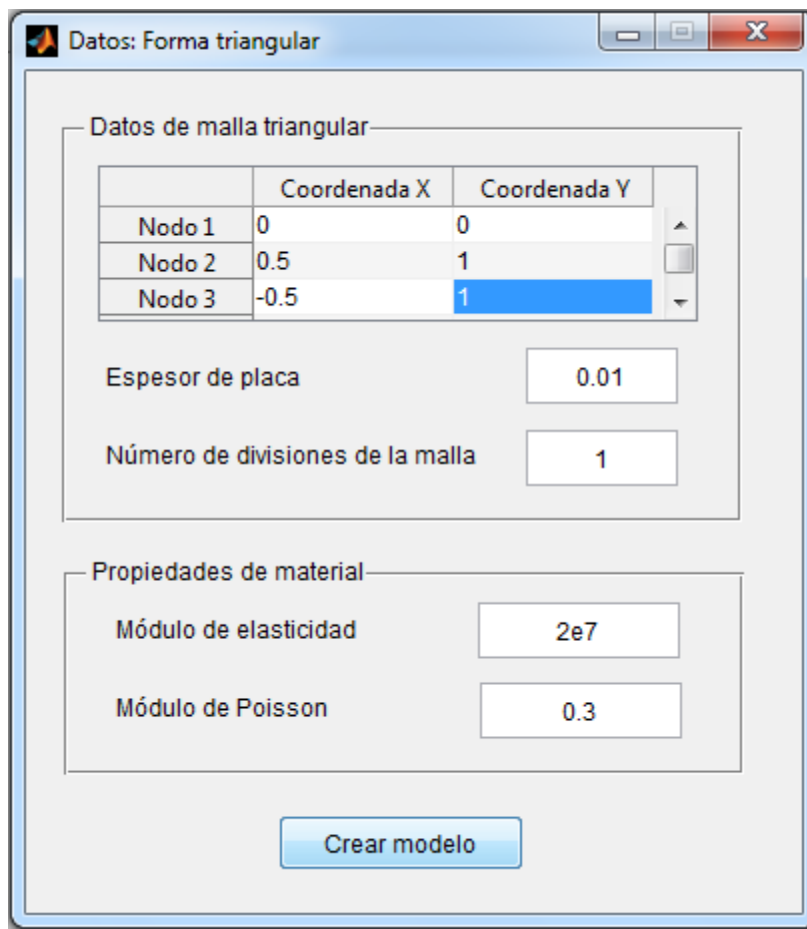


De la imagen podemos obtener las coordenadas de los vértices del elemento triangular las cuales son: $p_1 (0, 0)$, $p_2 (0.5, 1)$, $p_3 (-0.5, 1)$.

Con estos datos abrimos el programa y elegimos en la opción “Nuevo modelo”, “Malla de elementos triangulares”, “Malla Triangular”, como se ve en la figura.

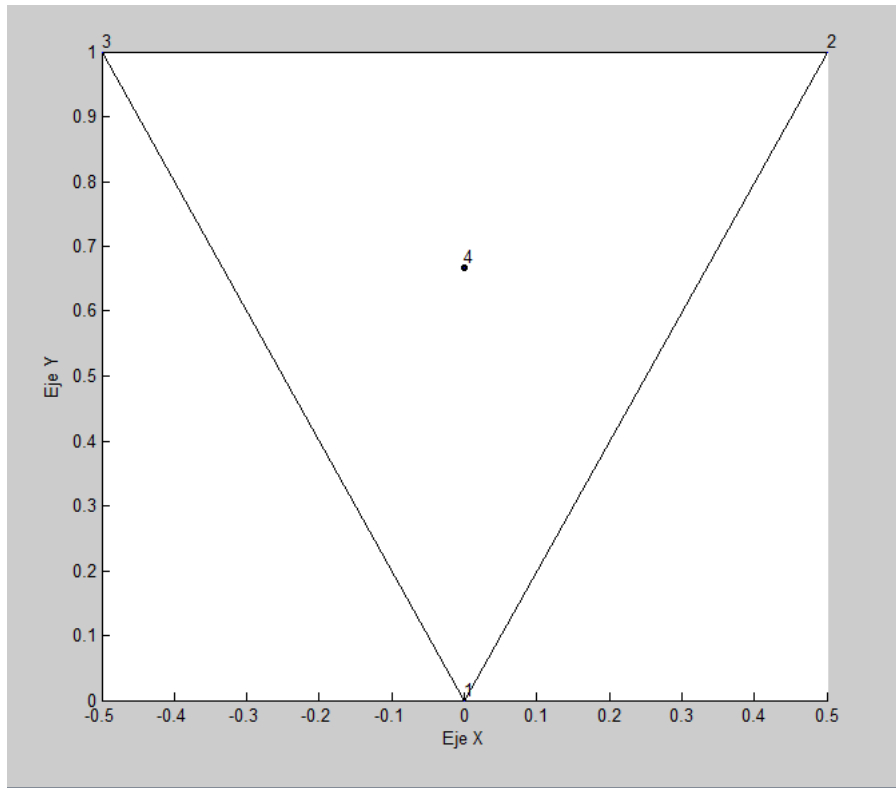


Una vez elegida esta opción, en la ventana para crear la malla de modelo de elementos triangulares introducimos los datos del problema y las coordenadas de los vértices del elemento.



En la casilla “Número de divisiones de la malla”, colocamos 1 para que el programa solo genere un elemento. Luego de introducir los datos, hacemos click en el botón “Crear modelo” y nos aparecerá la siguiente gráfica.

Imagen 50: Ejemplo de elemento triangular



El nodo 4 se genera automáticamente por el programa, por lo que no es necesario introducirlo en la tabla de datos.

Como el problema solo nos pide calcular la matriz de rigidez del elemento, escogemos la opción “Propiedades elemento” y hacemos click en la opción “Matriz de rigidez”, con lo que obtenemos el siguiente resultado para la matriz de rigidez:

Imagen 51: Matriz de rigidez del elemento triangular

```
ans =  
 98.3288    0.0000   26.8773   94.5513  -17.8533  -12.1795   94.5513   17.8533  -12.1795  -287.4313  
  0.0000    1.0073    0.0000   -1.7857    0.3892    0.1526    1.7857    0.3892   -0.1526   -0.0000  
 26.8773    0.0000    7.9670   24.2674   -4.5711   -3.1136   24.2674    4.5711   -3.1136   -75.4121  
 94.5513   -1.7857   24.2674  118.9332  -23.7828  -17.9029  101.7628   16.9834  -16.8956  -315.2473  
-17.8533    0.3892   -4.5711  -23.7828    5.7158    3.3043  -16.9834   -2.7053    2.9533    58.6195  
-12.1795    0.1526   -3.1136  -17.9029    3.3043    3.8767  -16.8956   -2.9533    2.7167    46.9780  
 94.5513    1.7857   24.2674  101.7628  -16.9834  -16.8956  118.9332   23.7828  -17.9029  -315.2473  
 17.8533    0.3892    4.5711   16.9834   -2.7053   -2.9533   23.7828    5.7158   -3.3043   -58.6195  
-12.1795   -0.1526   -3.1136  -16.8956    2.9533    2.7167  -17.9029   -3.3043    3.8767    46.9780  
-287.4313  -0.0000  -75.4121  -315.2473   58.6195   46.9780  -315.2473  -58.6195   46.9780   917.9258
```

Que se puede comparar con el resultado obtenido por el autor del libro dado en la página 310, que es el siguiente:

$$[k_e] = \begin{vmatrix} 98,33 & & & & & & & & & & & \\ & 0 & 1,01 & & & & & & & & & \text{simétrica} \\ 26,88 & 0 & 7,97 & & & & & & & & & \\ 94,55 & -1,79 & 24,27 & 118,93 & & & & & & & & \\ -17,85 & 0,39 & -4,57 & -23,78 & 5,72 & & & & & & & \\ -12,18 & 0,15 & -3,11 & -17,90 & 3,30 & 3,88 & & & & & & \\ 94,55 & 1,79 & 24,27 & 101,76 & -16,98 & -16,90 & 118,93 & & & & & \\ 17,85 & 0,39 & 4,57 & 16,98 & -2,71 & -2,95 & 23,78 & 5,72 & & & & \\ -12,18 & -0,15 & -3,11 & -16,90 & 2,95 & 2,72 & -17,90 & -3,30 & 3,88 & & & \\ -287,43 & 0 & -75,41 & -315,25 & 58,62 & 46,98 & -315,25 & -58,62 & 46,98 & 917,92 & & \end{vmatrix}$$

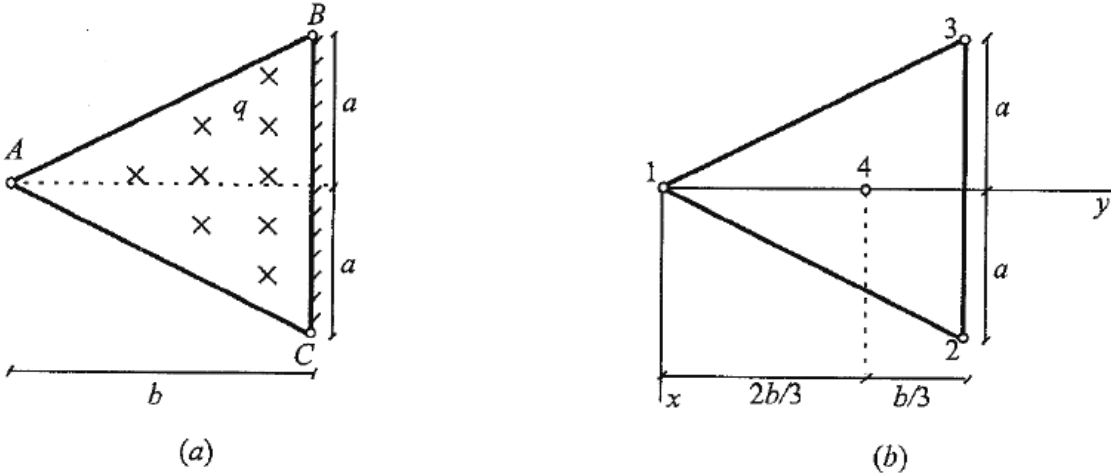
Es fácil notar que los resultados son los mismos.

Ejemplo 6.- Aplicación del programa al cálculo de flechas en elementos triangulares
 Una placa triangular de acero ABC de espesor t, está empotrada en extremo BC y sometida a una carga uniformemente distribuida q. Discretizando la placa en un elemento triangular de 10GDL, determinar los desplazamientos del extremo A de la placa.

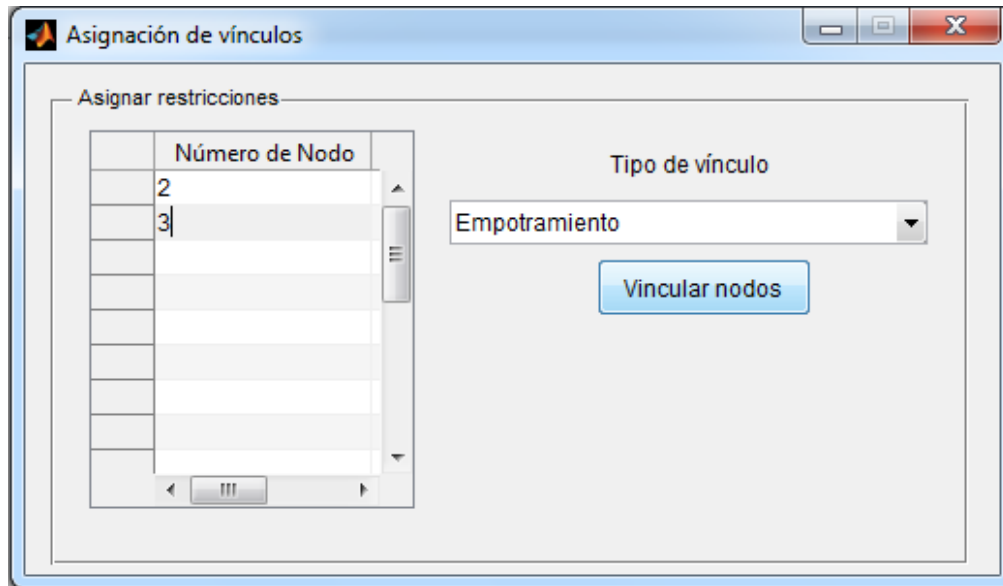
Datos:

$$a = 0.5\text{m}; \quad b = 1\text{m}; \quad E = 2e7 \text{ ton/m}^2; \quad t = 0.01\text{m}; \quad \mu = 0.3; \quad q=0.12\text{ton/m}^2$$

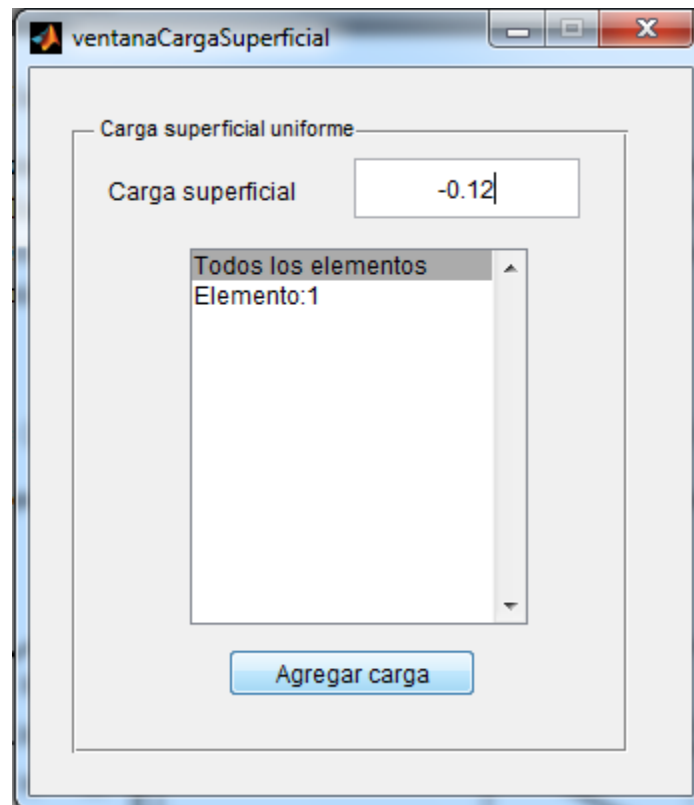
Imagen 52: Ejemplo de aplicación al cálculo de flechas en elementos triangulares



Al igual que en el ejemplo anterior las coordenadas de los vértices del elemento son: p1 (0,0), p2 (0.5, 1), p3 (-0.5, 1). Después de ser creada la malla de un solo elemento elegimos la opción “Asignar”, “Restricciones” y vinculamos los nodos 2 y 3, como se ve en la imagen.

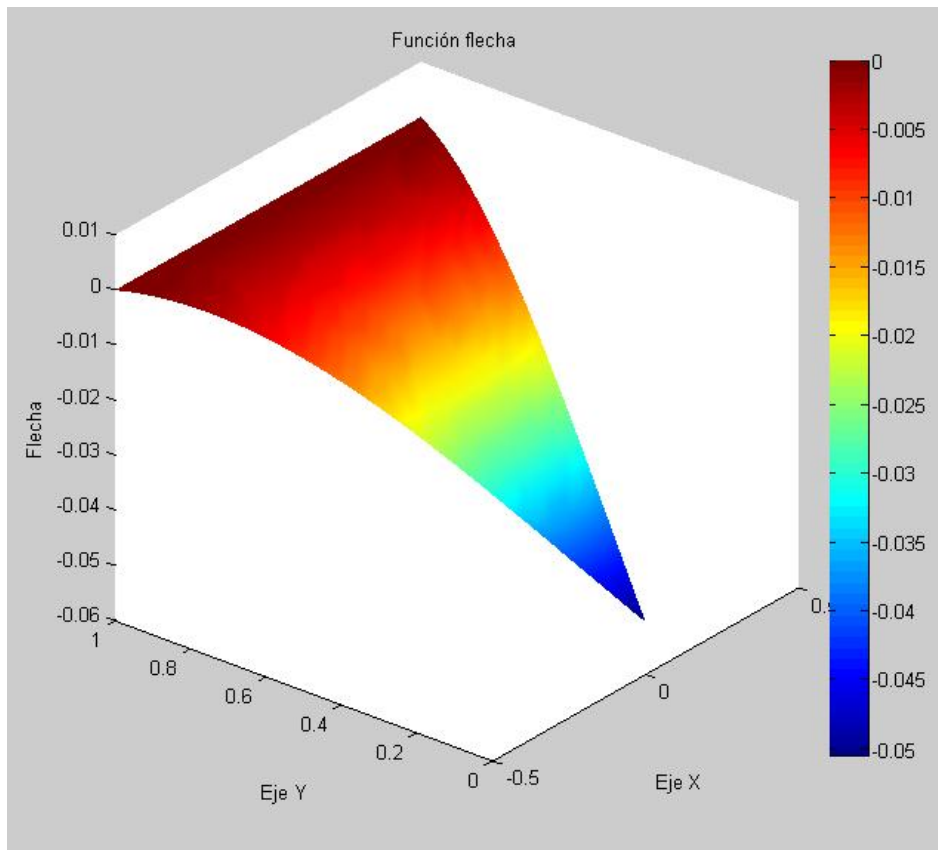


Una vez vinculados los nodos se procede a cargar la estructura para lo cual vamos al menú “Asignar” e introducimos el valor de la carga.



Con la estructura cargada y vinculada hacemos correr el programa, el cual nos muestra por defecto la deformada de la estructura.

Imagen 53: Vista 3D de la deformada del elemento triangular de 10GDL



Para visualizar el valor de la flecha en el gráfico, solo basta con rotarlo al plano XY y con la herramienta “Cursor Data” se puede obtener el valor del punto en cuestión, en este caso se puede ver en la gráfica el valor de la flecha en el extremo libre A, con un valor igual a:

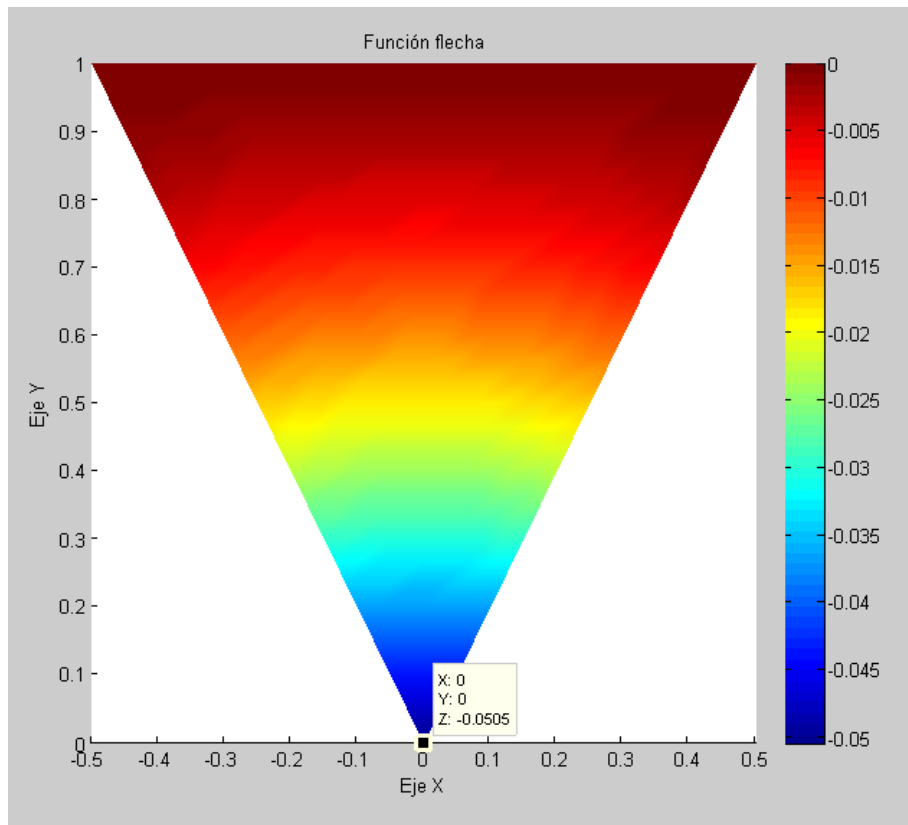
$$w = -0.00505m$$

El autor del libro consultado en la página 312, llega al siguiente resultado para la flecha en el extremo libre:

$$w = -0.005052m$$

Que es prácticamente el mismo resultado.

Imagen 54: Vista en planta de la deformada del elemento triangular de 10 GDL

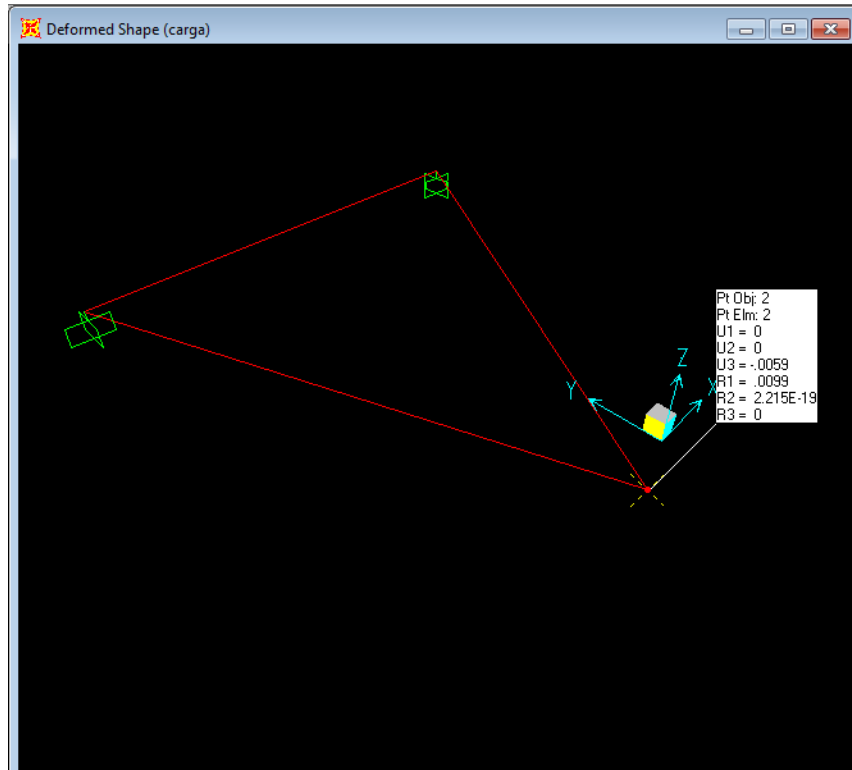


Como prueba adicional se va resolver el mismo problema con diferentes divisiones de malla, para estudiar la convergencia del elemento finito triangular de 10 grados de libertad.

Para validar los cálculos obtenidos, se realiza el mismo ejercicio con el programa sap2000, el cual nos da el siguiente resultado para la flecha en el extremo libre como se puede ver en el gráfico:

$$w = -0.0059\text{m}$$

A continuación se realizará la comparación de la convergencia a la solución entre el programa desarrollado en MatLab y el programa comercial sap2000.



Ejemplo 7.- Comparación de la convergencia a la solución del elemento triangular con el programa desarrollado en MatLab y el programa Sap2000.

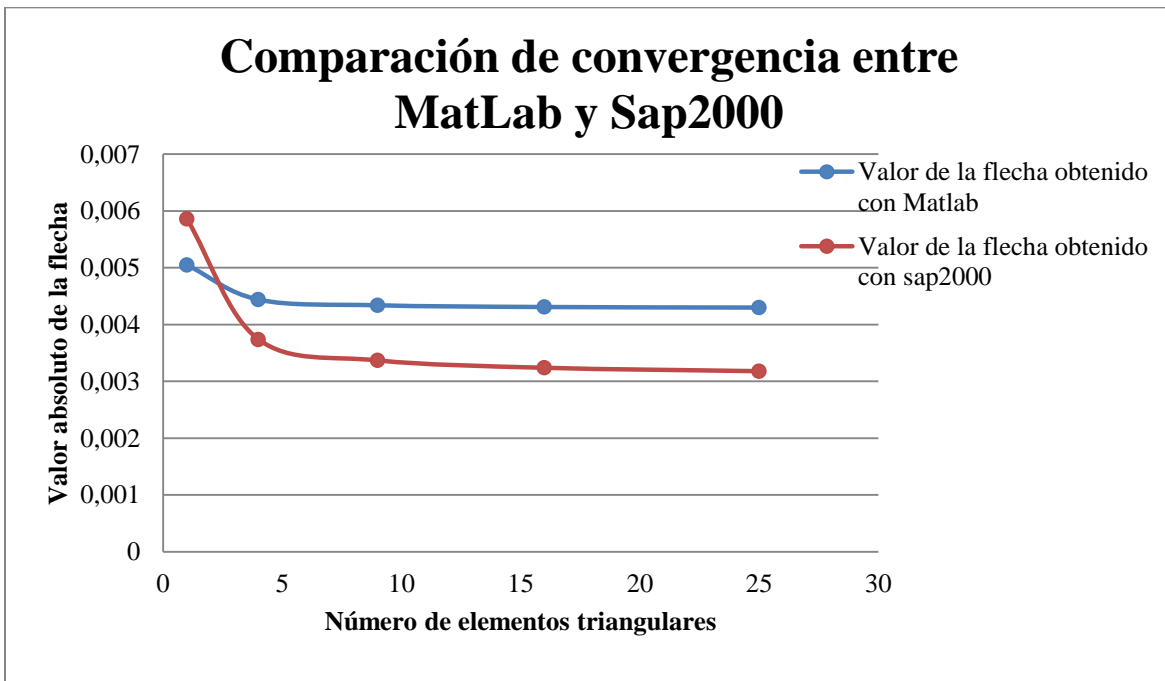
Se realiza la comparación de la convergencia de la solución para el elemento finito triangular desarrollado en MatLab y los resultados obtenidos por sap2000, para esto utilizamos el ejemplo anterior y lo resolvemos con divisiones de malla de: 1, 2, 3, 4, 5 divisiones por lado del elemento.

Los resultados obtenidos se muestran a continuación:

Tabla 11: Cuadro comparativo de la convergencia de MatLab y Sap2000

Nº división	Nº elementos	MatLab	Sap2000
1	1	0.00505	0.00586
2	4	0.00444	0.00374
3	9	0.00434	0.00337
4	16	0.00431	0.00324
5	25	0.0043	0.00318

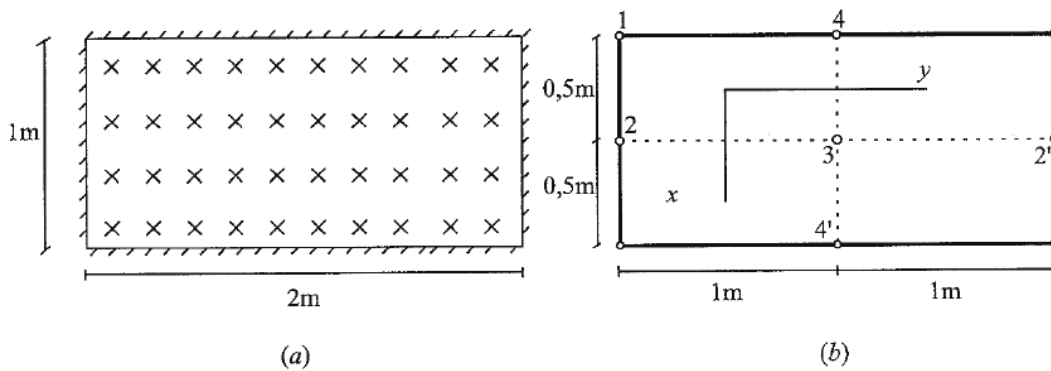
Gráfico 3: Gráfico comparativo de convergencia entre los dos programas



En los datos anteriores, se puede apreciar que el programa desarrollado en MatLab no converge a la misma solución que sap2000, esto puede deberse a que el elemento triangular utilizado en desarrollo del programa en MatLab, no es un elemento conforme.

Ejemplo 8.- Desempeño del elemento triangular de 10 grados de libertad en problemas de placas rectangulares.

En este ejemplo se utilizaron los mismos datos del ejemplo de la placa rectangular utilizada inicialmente y se resuelve el problema utilizando un mallado de elementos triangulares, para luego compararlo con la solución considerada exacta en la placa empotrada en todos sus bordes. Se utilizaron mallas de 2x2, 4x4, 6x6, y 8x8 para el elemento rectangular y el mismo tipo de mallado para el elemento triangular, con la diferencia de que en el mallado de elementos triangulares se obtiene un mayor número de elementos finitos.



Datos:

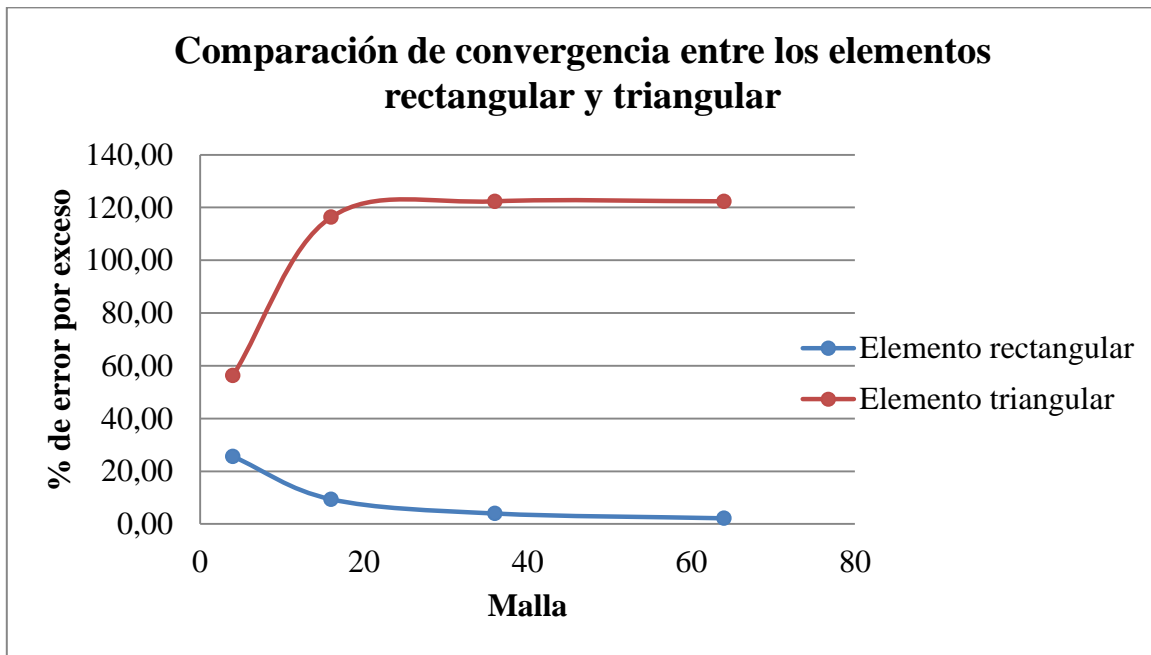
$a = 0.5\text{m}; \quad b = 1\text{m}; \quad E = 2e7 \text{ ton/m}^2; \quad t = 0.01\text{m}; \quad \mu = 0.3; \quad q=0.12\text{ton/m}^2$

Los resultados obtenidos se muestran en la siguiente tabla:

Tabla 12: Desempeño del elemento triangular de 10 GDL frente al elemento rectangular de 12 GDL

N° elementos	Malla	Elemento rectangular de 12 GDL			Elemento triangular de 10 GDL		
		Aproximada	Exacta	% error	Aproximada	Exacta	% error
4	2x2	-0.00209	-0.001664	25.60	-0.0026	-0.001664	56.25
16	4x4	-0.00182	-0.001664	9.38	-0.0036	-0.001664	116.35
36	6x6	-0.00173	-0.001664	3.97	-0.0037	-0.001664	122.36
64	8x8	-0.0017	-0.001664	2.16	-0.0037	-0.001664	122.36

Gráfico 4: Comparación entre la convergencia del elemento triangular de 10 GDL y el elemento rectangular de 12 GDL



Se puede ver que el elemento triangular de 10 grados de libertad tiene un mal desempeño al modelar placas rectangulares, por lo que solo debe ser usado con fines académicos.

Conclusiones y recomendaciones

Conclusiones

Por lo expuesto anteriormente, del desarrollo del presente trabajo de investigación se puede concluir lo siguiente:

1. El programa obtenido como resultado del trabajo de investigación funciona correctamente ya que los resultados fueron verificados con la solución exacta cuando está era conocida.
2. Se puede modelar con éxito placas rectangulares, en el caso de las placas triangulares el programa converge a soluciones aparentemente incorrectas, por lo que el elemento triangular debe ser programado nuevamente.
3. Debido a la formulación de los elementos utilizados en el programa desarrollado en MatLab, se puede lograr soluciones muy satisfactorias con un número reducido de elementos finitos, ya que utiliza funciones de forma de grado elevado para la interpolación de deformaciones.
4. De las pruebas de convergencia realizadas entre el programa desarrollado en MatLab y el programa comercial Sap2000 para los elementos rectangulares, se puede ver que el programa en MatLab converge más rápidamente que el programa Sap2000 debido a que se utilizaron funciones de forma de grado elevado para una mejor aproximación. Esto no quiere decir que el programa Sap2000 no brinde resultados aceptables sino que se requiere un mayor número de elementos finitos para lograr la convergencia a la solución.
5. El programa resultado del presente trabajo de investigación, sirve además de herramienta de cálculo, como apoyo docente ya que muestra todas las propiedades de la formulación del elemento durante el proceso de resolución del problema como también las propiedades del elemento finito en consideración tales como ser: la matriz de rigidez, la matriz de coordenadas particularizadas y la matriz "N" con los coeficientes de las funciones de forma del elemento.
6. Se logró cumplir con el objetivo de brindar un software con una interfaz gráfica que permite el ingreso de datos de forma sencilla y que sea de apoyo en el aprendizaje del método de los elementos finitos aplicado al análisis de placas planas.
7. La velocidad de cálculo obtenida con MatLab en comparación con otros lenguajes más potentes como C, C++ o Java, es relativamente lenta, sin embargo MatLab cuenta con subrutinas ya agregadas que facilitan considerablemente el proceso de cálculo, como es en el caso de operaciones matriciales, contando con algoritmos altamente optimizados, los cuales deben ser formulados en los lenguajes de alto nivel citados anteriormente.

Recomendaciones

Como resultado del presente trabajo de investigación se tiene las siguientes recomendaciones para el uso del programa obtenido y para trabajos futuros:

1. Antes de hacer uso del programa, el usuario debe asegurarse de que las unidades de trabajo sean consistentes debido a que el programa no cuenta con un sistema de unidades definido ni con un algoritmo de conversión de unidades.
2. Dada la formulación utilizada para los elementos rectangulares, sólo se puede crear la matriz de rigidez de elementos cuadriláteros cuyos lados sean paralelos a los ejes x e y .
3. Para modelar elementos de cuatro lados de forma irregular, se debe combinar elementos triangulares.
4. Para modelar placas rectangulares sometidas puramente a flexión, es mejor utilizar el elemento rectangular de 16 grados de libertad con mallas de 4×4 para obtener una solución más aproximada a la solución real.
5. El elemento triangular de 10 grados de libertad debe ser utilizado sólo con fines académicos y no para modelar estructuras reales.
6. Se recomienda incentivar a los estudiantes de nuestra superior casa de estudio al aprendizaje de programación y desarrollo de software para así contar con nuestras propias herramientas de cálculo en las diferentes áreas de aplicación de nuestra carrera.
7. Se recomienda el uso masivo del software MatLab durante el curso de la carrera, ya que es una herramienta muy útil en el campo de la programación científica, con un alto potencial de aplicación en las diferentes ramas de la ingeniería civil como de cualquier otra ingeniería.
8. El estudiante así como el profesional de ingeniería, debe conocer a cabalidad el método de los elementos finitos para así entender cómo funcionan los programas comerciales que se utilizan en la actualidad.
9. Se recomienda añadir un algoritmo de mallado que genere mallas de elementos finitos de cualquier forma geométrica, para extender la aplicación del programa.
10. Se recomienda la adición de código de trabajos previos al presente, desarrollados por los ingenieros Horacio Calizaya (2015) y Jhonatan Nogales (2017), para así contar con un programa de mayor capacidad de resolución de estructuras espaciales o planas, desarrollado en nuestra superior casa de estudio.